Energy and QoS aware Routing in Wireless Sensor Networks

Kemal Akkaya and Mohamed Younis
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
kemal1 | younis@cs.umbc.edu

Abstract*. Many new routing protocols have been proposed for wireless sensor networks in recent years. Almost all of the routing protocols considered energy efficiency as the ultimate objective since energy is a very scarce resource for sensor nodes. However, the introduction imaging sensors has posed additional challenges. Transmission of imaging data requires both energy and QoS aware routing in order to ensure efficient usage of the sensors and effective access to the gathered measurements. In this paper, we propose an energy-aware QoS routing protocol for sensor networks which can also run efficiently with best-effort traffic. The protocol finds a least-cost, delay-constrained path for real-time data in terms of link cost that captures nodes' energy reserve, transmission energy, error rate and other communication parameters. Moreover, the throughput for non-real-time data is maximized by adjusting the service rate for both real-time and non-real-time data at the sensor nodes. Such adjustment of service rate is done by using two different mechanisms. Simulation results have demonstrated the effectiveness of our approach for different metrics with respect to the baseline approach where same link cost function is used without any service differentiation mechanism.*

Keywords: *Sensor networks, QoS routing, energy-aware routing, real-time traffic*

1. Introduction

Recent advances in micro-electro-mechanical systems (MEMS) and low power and highly integrated digital electronics have led to the development of micro sensors [1][2][3][4][5][5][7]. Such sensors are generally equipped with data processing and communication capabilities. The sensing circuitry measures ambient conditions related to the environment surrounding the sensor and transforms them into an electric signal. Processing such a signal reveals some properties about objects located and/or events happening in the vicinity of the sensor. The sensor sends such sensed data, usually via radio transmitter, to a command center either directly or through a data concentration center (a gateway). The gateway can perform fusion of the sensed data in order to filter out erroneous data and anomalies and to draw conclusions from the reported data over a period of time.

The continuous decrease in the size and cost of sensors has motivated intensive research in the past few years addressing the potential of collaboration among sensors in data gathering and processing via an ad hoc wireless network. Networking unattended sensor nodes is expected to have significant impact on the efficiency of many military and civil applications, such as combat field surveillance, security and disaster management. A network of sensors can be used to gather meteorological variables such as temperature and pressure. These measurements can be used in preparing forecasts or detecting harsh natural phenomena. In disaster management situations such as earthquakes, sensor networks can be used to selectively map the affected regions directing emergency response units to survivors. In military situations, sensor networks can be used in surveillance missions and can be used to detect moving targets, chemical gases, or presence of micro-agents.

However, sensor nodes are constrained in energy supply and bandwidth. Such constraints combined with a typical deployment of large number of sensor nodes have necessitated energy-awareness at the layers of networking protocol stack including network layer. Routing of sensor data has been one of the challenging areas in wireless sensor network research. Current research on routing in wireless sensor networks 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 [2][4][8][9][11][10][12]. Since the data they deal with is not in large amounts and flow in low rates to the sink, the concepts of latency, throughput and delay were not primary concerns in most of the published work on sensor networks. However, the introduction of imaging sensors has posed additional challenges for routing in sensor networks. Transmission of imaging data requires careful handling in order to ensure that end-to-end delay is within acceptable range. Such performance metrics are usually referred to as quality of service (QoS) of the communication network. Therefore, collecting sensed imaging data requires both energy and QoS aware routing in order to ensure efficient usage of the sensors and effective access to the gathered measurements.

QoS protocols in sensor networks have several applications including real time target tracking in battle environments, emergent event triggering in monitoring applications etc. Consider the following scenario: In a battle environment it is crucial to locate, detect and identify a target. In order to identify a target, we should employ imaging sensors. After locating and detecting the target without the need of imaging sensors, we can turn on those sensors to get for instance an image of the target periodically for sending to the base station or gateway. Since, it is a battle environment; this requires a real-time data exchange between sensors and controller in order to take the proper actions. However, we should deal with real-time data, which requires certain bandwidth with minimum possible delay. In that case, a service differentiation mechanism is needed in order to guarantee the reliable delivery of the real-time data.

Energy-aware QoS routing in sensor networks will ensure guaranteed bandwidth (or delay) through the duration of a connection as well as providing the use of the most energy efficient path. To the best of our knowledge, no previous research has addressed QoS routing in sensor networks. In this paper, we present an energy-aware QoS routing mechanism for wireless sensor networks. Our proposed protocol extends the routing approach in [12] and considers only end-to-end delay. The protocol looks for a delay-constrained path with the least possible cost. The cost function which captures remaining and transmission energy and error rate, is defined for each link. Alternative paths with bigger costs are tried until one, which meets the end-to-end delay requirement and maximizes the throughput for best effort traffic is found. Our protocol does not introduce any extra overhead to the sensors.

In the balance of this section we describe the sensor network architecture that we consider and summarize the related work. In section 2, we analyze the complexity of the QoS routing problem in sensor networks and describe our approach. Section 3 includes simulations and evaluations of the protocol. Finally we conclude the paper in section 4 and outline our future research.

1.1. Sensor Network Architecture

A set of sensors is spread throughout an area of interest to detect and possibly track events/targets in this area. The sensors are battery-operated with diverse capabilities and types and are empowered with limited data processing engines. The availability of imaging sensors is of particular interest due to the

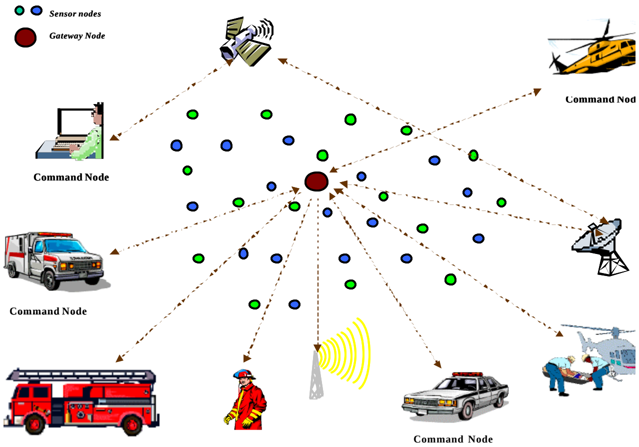

Fig. 1 : Three-tier sensor network architecture

quality of service constraints associated with data generated by such sensors. The mission for these sensors is dynamically changing to serve the need of one or multiple command nodes. Command nodes can be stationary or mobile. In a disaster management environment, coordination centers are typical stationary command nodes, while paramedics, fire trucks, rescue vehicles and evacuation helicopters are examples of mobile command nodes. A gateway node is a less energy-constrained node deployed in the physical proximity of sensors. The gateway is responsible for organizing the activities at sensor nodes to achieve a mission, fusing data collected by sensor nodes, coordinating communication among sensor nodes and interacting with command nodes. We are considering both the gateway and sensor nodes as stationary. All the sensors are assumed to be within the communication range of the gateway node. The architecture is depicted in Fig 1.

The sensor is assumed to be capable of operating in an active mode or a low-power stand-by mode. The sensing and processing circuits can be powered on and off. In addition both the radio transmitter and receiver can be independently turned on and off and the transmission power can be programmed for a required range. It is also assumed that the sensor can act as a relay to forward data from another sensor. It is worth noting that most of these capabilities are available on some of the advanced sensors, e.g. the Acoustic Ballistic Module from SenTech Inc. [13]. The gateway node is assumed to know its location, e.g. via the use of GPS.

The described system's architecture raises many interesting issues such as mission-oriented sensor organization, network management, gateway to command node communication protocol, support of QoS traffic generated by imaging sensors, etc. While many of these issues are studied in the context of wireless networking research, the naturally resource constrained sensor-based environment makes these technical issues untraditional and challenging. For example energy efficiency has to be a core objective of the system design, a factor that has not been considered for typical networks. In this paper, we only focus on the energy-aware and QoS routing of sensor data among the communicating nodes. While the gateway will take charge of sensor organization based on the mission and available energy in each sensor, we assume knowledge of which sensors need to be active in signal processing, e.g. using the approaches presented in [1][14].

1.2. Related Work

In traditional best-effort routing throughput and delay are the main concerns. There is no guarantee that a certain performance in throughput or delay will be ensured throughout the connection. However, in some cases where real-time or multimedia data are involved in communication, some performance guarantees in certain metrics such as delay, bandwidth and delay jitter are needed. Such guarantees can be achieved by employing special mechanisms known as QoS routing protocols.

While contemporary best-effort routing approaches address unconstrained traffic, QoS routing is usually performed through resource reservation in a connection-oriented communication in order to meet the QoS requirements for each individual connection. While many mechanisms have been proposed for routing QoS constrained real-time multimedia data in wire-based networks [15][16][17][18][19], they cannot be directly applied to wireless networks due to inherent characteristics of wireless environments and limited resources, such as bandwidth. Therefore, several new protocols have been proposed for QoS routing in wireless ad-hoc networks taking the dynamic nature (due to mobility of the nodes) of the network into account [20][21][22][23][24]. Some of these proposed protocols consider the imprecise state information while determining the routes [20][21]. CEDAR is another QoS aware protocol, which uses the idea of core nodes (dominating set) of the network while determining the paths [22]. Using routes found through the network core, a QoS path can be easily found. However, if any node in the core is broken, it will cost too much in terms of resource usage to reconstruct the core. Lin [23] and Zhu et al. [24] have proposed QoS routing protocols specifically designed for TDMA-based ad-hoc networks. Both protocols can build a QoS route from a source to destination with reserved bandwidth. The bandwidth calculation is done hop-by-hop in a distributed fashion.

Another protocol for wireless networks that includes some notion of QoS in its routing decisions is the Sequential Assignment Routing (SAR) [4]. The SAR protocol creates trees routed from one-hop neighbor of the sink by taking the QoS metric, the energy resource on each path and the priority level of each packet into consideration. By using created trees, multiple paths from sink to sensors are formed. One of these paths is selected according to the energy resources and achievable QoS on each path. In our approach, we not only select a path from a list of candidate paths that meet the end-to-end delay requirement, but maximize the throughput for best effort traffic as well. In addition, the SAR approach suffers the overhead of maintaining the node states at each sensor node and maintaining the multiple paths from each node to the sink. Our protocol does not require sensor's involvement in route setup.

Most of the QoS routing algorithms discussed in this section are based on the mobility of the nodes and none of them consider energy awareness along with the QoS parameters. Although they are well suited to mobile ad hoc networks, the emerging complexity from mobility in such routing algorithms will be an over-kill for the systems where nodes are not mobile and have limited resources, such as bandwidth and energy. On the other hand, routing protocols proposed specifically for wireless sensor networks are designed according to the needs of sensor networks, none of them considers any QoS or service differentiation mechanism in order to handle challenges posed by imaging sensors and real-time applications of sensor networks. Our proposed approach tackles these challenges into account so that both the system lifetime will be maximized and QoS requirements are met.

2. Energy-aware QoS Routing

Our aim is to find an optimal path to the gateway in terms of energy consumption and error rate while meeting the end-to-end delay requirements. End-to-end delay requirements are associated only with the real-time data. Note that, in this case we have both real-time and non-real-time traffic coexisting in the network, which makes the problem more complex. We not only should find paths that meet the requirements for real-time traffic, but need to maximize the throughput for non-real time traffic as well. This is because most of the critical applications such as battlefield surveillance have to receive for instance acoustic data regularly in order not to miss targets. Therefore it is important to prevent the real-time traffic from consuming the bulk of network bandwidth and leave non-real-time data starving and thus incurring large amount of delay.

The described QoS routing problem is very similar to typical path constrained path optimization (PCPO) problems, which are proved to be NP-complete [25]. We are trying to find least-cost path, which meets the end-to-end delay path constraint. However, in our case there is an extra goal, which is basically to maximize the throughput of non-real-time traffic. Our approach is based on associating a cost function for each link and used a K least cost path algorithm to find a set of candidate routes. Such routes are checked against the end-to-end constraints and the one that provides maximum throughput is picked. Before explaining the details of proposed algorithm, we introduce the queuing model.

2.1 Queuing Model

The queuing model is specifically designed for the case of coexistence of real-time and non-real-time traffic in each sensor node. The model we employ is inspired from class-based queuing [26]. We use different queues for the two different types of traffic. Basically, we have real-time traffic and non-real-time (normal) traffic whose packets are labeled accordingly. On each node there is 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 according to the bandwidth ratio ‘r” of each type of traffic on that link. The model is depicted in Fig. 2.

The bandwidth ratio r, is actually an initial value set by the gateway and represents the amount of bandwidth to be dedicated both to the real-time and non-real-time traffic on a particular outgoing link. Moreover, both classes can borrow bandwidth from each other when one of the two types of traffic is non-existent or under the limit. As indicated in Figure 3, this r-value is also used to calculate the service rate of real-time and non-real-time traffic on that particular node, with $r_i m$ and $(1-r_i)m$ being respectively the service rate for real-time and non-real-time data on sensor node i.

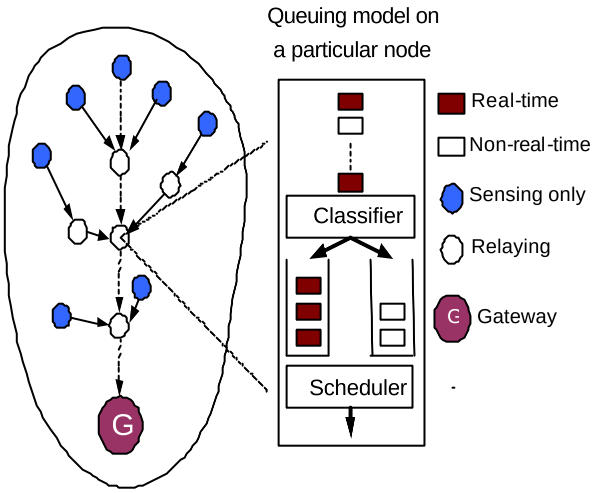

Fig. 2. Queuing model in a particular sensor node

Since the queuing delay depends on this r-value, we cannot calculate the end-to-end delay for a particular path without knowing the r-value. Therefore we should first find a list of candidate least-cost paths and then select one that meets the end-to-end delay requirement. Our approach is based on a two-step strategy incorporating both link-based costs and end-to-end constraints. First we calculate the candidate paths without considering the end-to-end delay. What we do is simply calculate costs for

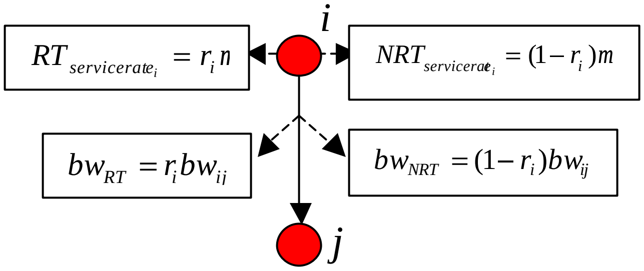

Fig. 3. Bandwidth sharing and service rates for a sensor node

each particular link and then use an extended version of Dijkstra's algorithm to find an ascending set of least cost paths. Once we obtain these candidate paths, we further check them to identify those that meet our end-to-end QoS requirements by trying to find an optimal r-value that will also maximize the throughput for non-real-time traffic.

2.2 Calculation of link costs

We consider the factors for the cost function on each particular link separately except the end-to-end delay requirement, which should be for the whole path (i.e. all the links on that path). We define the following cost function for a link between nodes i and j:

$$\cos t_{ij} = \sum_{k=0}^{2} CF_k = c_0 \times (dist_{ij})^l + c_1 \times f(energy_j) + c_2 \times f(e_{ij})$$ where,

- $dist_{ij}$ is the distance between the nodes i and j,
- $f(energy_j)$ is the function for finding current residual energy of node j,
- $f(e_{ij})$ is the function for finding the error rate on the link between i and j.

Hence, it's not part of the cost function. Cost factors are defined as follows:

- CF_0 (Communication Cost)= $c_0 \times (dist_{ij})^l$, where c_0 is a weighting constant and the parameter l depends on the environment, and typically equals to 2. This factor reflects the cost of the wireless transmission power, which is directly proportional to the distance raised to some power l. The closer a node to the destination, the less its cost factor CF_0 and more attractive it is for routing.
- CF_1 (Energy Stock)= $c_1 \times f(energy_j)$. This factor reflects the remaining battery lifetime (i.e. energy usage rate), which favors nodes with more energy. The more energy the node contains, the better it is for routing.
- CF_2 (Error rate)= $c_2 \times f(e_{ij})$ where f is a function of distance between nodes i and j and buffer size on node j (i.e. $dist_{ij} / buffer\_size_j$). The links with high error rate will increase the cost function, thus will be avoided.

2.3 Estimation of end-to-end delay for a path

In order to find a QoS path for sending real-time data to the gateway, end-to-end delay requirement should be met. Before explaining the computation of the delay for a particular path P, we introduce the notation below:

l_{RT} : Real-time data generation rate for imaging sensors

$r_i m$: Service rate for real-time data on sensor node *i*

$(1-r_i)m$: Service rate for non-real-time data on sensor node *i*

p_i : The number of sensing neighbors (data generators) of node *i* on path *P*

q_i : The number of relaying neighbors (data forwarders) of node *i* on path *P*

$l_{RT}^{(i)}$: Real-time data rate on sensor node *i*

$Q_{RT}^{(i)}$: Queuing delay on a node *i* for real-time traffic

T_E : End-to-end queuing delay for a particular path *P (ignoring propagation delay)*

$T_{end-end}$: End-to-end delay for a particular path *P*

$T_{required}$: End-to-end delay requirement for all paths

m : The number of nodes on path P

$Nodes$: The set of all the sensing nodes that are part of path P

We assume that the propagation delay is negligible. We also assume that all the imaging sensors have the same real-time data generation rate l_{RT}. Total real-time data rate by p_i nodes will be $p_i \, l_{RT}$ and total real-time data rate by q_i nodes will be added recursively for each relaying only node. Then total real-time data load on a sensor node is:

$$l_{RT}^{(i)} = p_i \, l_{RT} + \sum_{j=1}^{q_i} p_j \, l_{RT}^{(j)}$$

The average waiting time including the service time in the queue in M/M/1 model is stated as $W = \frac{1}{m - l}$ where μ is the link transmission rate or service rate and ? is the packet arrival rate [27]. Hence, total queuing delay (including the service time), $TQ_{RT}^{(i)}$ on a node *i* is:

$$Q_{RT}^{(i)} = \frac{1}{r_i m - l_{RT}^{(i)}} \qquad [1]$$

We make an approximation to simplify the end-to-end queuing delay by assuming the incoming traffic to real-time and non-real-time queues are stochastically independent. Thus, the end-to-end queuing delay for a particular path is:

$$T_E = \sum_{\forall i \in Path} Q_{RT}^{(i)} =$$

$$\sum_{\forall i \in Path} \frac{1}{r_i m - l_{RT}^{(i)}} = \sum_{\forall i \in Path} \frac{1}{r_i m - p_i \, l_{RT} - \sum_{j=1}^{q_i} p_j \, l_{RT}^{(j)}}$$

Since we ignore the propagation delay, total end-to-end delay will be:

$$T_{end-end} = \sum_{\forall i \in Path} \frac{1}{r_i m - p_i \, l_{RT} - \sum_{j=1}^{q_i} p_j \, l_{RT}^{(j)}} \qquad [2]$$

2.4 Single-r Mechanism

While we generate a formula for calculating the end-to-end delay for a particular path, finding the optimal *r*-values for each link as far as the queuing delay is concerned, will be very difficult optimization problem to solve. Moreover, the distribution of these *r*-values to each node is not an easy task because each value should be unicasted to the proper sensor node rather than broadcasting it to all the sensors, which might bring a lot of overhead. Therefore, we follow an approach, which will eliminate the overhead and complexity of the problem. Basically, we define each *r*-value to be same on each link so that the optimization problem will be simple and this unique *r*-value can be easily broadcasted to all the sensors by the gateway.

If we let all *r*-values be same for every link then the formula will be stated as:

$$T_{end-end} = \sum_{\forall i \in Path} \frac{1}{rm - p_i l_{RT} - \sum_{j=1}^{q_i} p_j l_{RT}^{(j)}}$$

Then the problem is stated as an optimization problem as follows:

$$Max\left(\sum_{\forall i \in Path} ((1-r)m) \right)$$

subject to : $T_{end-end} \leq T_{required}$ and $0 \leq r < 1$

In order to find *r*-value from the above inequality of $T_{end-end} \leq T_{required}$, for simplification we consider finding an *r*-value which will satisfy the last hop node's delay since the last node will be getting the actual longest queuing delay. As a consequence, the other nodes before the last node will already be satisfied with that *r*-value and will use the same value. We divide $T_{required}$ into m equal time slots, where m is the number of nodes on a particular path. The calculation of r for the last hop node m is as follows: $\frac{T_{required}}{m} = Q_{RT}^{m} = \frac{1}{rm - l_{RT}(p_m + \sum_{\forall k \in Nodes} p_k)}$

$$r = \frac{m}{mT_{required}} + \frac{l_{RT}}{m}(p_m + \sum_{\forall k \in Nodes} p_k) \qquad [3]$$

By considering the optimization problem above, we propose the algorithm shown in Fig. 4, to find a least-cost path, which meets the constraints and maximizes the throughput for non-real-time data. The algorithm calculates the cost for each link, line 1 of Fig. 4, based on the cost function defined in section 3.2. Then, for each node the least cost path to the gateway is found by running Dijkstra's shortest path algorithm in line 2. Between lines 5-15, appropriate *r*-values are calculated for paths from imaging sensors to the gateway. For each sensor node that has imaging capability, an *r*-value is calculated on the current path

```
1 Calculate $cost_{ij}, \forall i, j \in V$
2 Find least cost path for each node by using Dijkstra
3 for each imaging sensor node i do
4 begin
5   Compute r from $T_{end-end}(p_i) = T_{required}$ (as above)
6   if (r is in range [0,1)) then
7       Add r to a list corresponding to node i
8   else
9       Find K least cost paths $(P_i^K)$ to the gateway
10      for each $k \in K$ do
11          Recompute r from $T_{end-end}(p_i^k) = T_{required}$
12          if (r is in range [0,1)) then
13              break;
14          if no appropriate r is found
15              Reject the connection
16 end
17 Find max r from the list
```

Fig. 4. Pseudo code for the proposed algorithm

(line 5). If that value is not between 0 and 1, extended Dijsktra algorithm for K-shortest path is run in order to find alternative paths with bigger costs (line 9). K different least-cost paths are tried in order to find a proper *r*-value between 0 and 1 (lines 10-13). If there is no such *r*-value, the connection request of that node to the gateway is rejected.

The algorithm might generate different *r*-values for different paths. Since, the *r*-values are stored in a list; the maximum of them is selected to be used for the whole network (line 17). That *r*-value will satisfy the end-to-end delay requirement for all the paths established from imaging sensors to the gateway.

In order to find the K least cost paths (i.e. K shortest paths), we modified an extended version of Dijkstra's algorithm given in [28]. Since, the algorithm can suffer loops during execution; we modified the algorithm in such a way that each time a new path is searched for a particular node; only node-disjoint paths are considered during the process. This eliminates loops and ensures simplicity and efficiency. This might also help finding a proper *r*-value easily since that node-disjoint path will not inherit the congestion in the former path. Interested reader is referred to [28] for further information.

2.5 Multi-r Mechanism

Since the single-*r* mechanism is just an approximation to the optimal solution of allocating r-values for each node by assuming a unique r-value for each node, we extended the model so that it will allow different *r*-values to be assigned to sensor nodes for better resource allocation. In order to find different r-values, each node's *r*-value is calculated by setting maximum allowable queuing delay for every node on the path proportional to arrival rate of real-time traffic to that node. The least-cost path is picked. The gateway calculates a delay factor "*Dd*" by dividing the value of the end-to-end delay "*d*" by the accumulative arrival rates of real-time traffic at all nodes on the path. The gateway then broadcasts the value of "*Dd*" to all nodes on the path so that they can use it to derive their *r*-value.

Then Δd will be calculated as follows:

$$\Delta d = \frac{T_{required}}{\sum_{\forall i \in Path}\left(p_i + \sum_{j=1}^{q_i} p_j\right) * l_{RT}} \qquad [4]$$

Each sensor node i will calculate its r-value r_i by using Δd as follows:

From [1], $\Delta d = \dfrac{1}{r_i m - l_{RT} * \left(p_i + \sum_{j=1}^{q_i} p_j\right)}$

$$r_i = \frac{1}{\Delta d * m} + \frac{l_{RT}}{m} * \left(p_i + \sum_{j=1}^{q_i} p_j\right) \qquad [5]$$

Then the problem will be to maximize the total throughput on each particular path:

$Max\left(\sum_{\forall i \in Path}(1 - r_i)\right)$ where $0 \le r_i < 1$.

3. Experimental Results

The effectiveness of the energy-aware QoS routing approach is validated through simulation. This section describes the performance metrics, simulation environment, and experimental results.

3.1. Performance Metrics

We have used the following metrics to capture the performance of our QoS routing approach:

- *Time to first node to die*: When the first node runs out of energy, the network within the cluster is said to be partitioned. The name network partitioning reflects the fact that some routes become invalid and cluster-wide rerouting may be immanent.
- *Average lifetime of a node*: This gives a good measure of the network lifetime. A routing algorithm, which maximizes the lifetime of the network, is desirable. This metric also shows how efficient is the algorithm in energy consumption.
- *Average delay per packet*: Defined as the average time a packet takes from a sensor node to the gateway. Most energy aware routing algorithms try to minimize the consumed energy. However, the applications that deal with real-time data is delay sensitive, so this metric is important in our case.
- *Network Throughput*: Defined as the total number of data packets received at the gateway divided by the simulation time. The throughput for both real-time and non-real-time traffic will be considered independently.

3.2 Environment Setup

In the experiments we have considered a network of 100 randomly placed nodes in a 1000×1000 meter square area. The gateway position is determined randomly within the boundaries of deployment area. A free space propagation channel model is assumed [29] with the capacity set to 2Mbps. Packet lengths are 10 Kbit for data packets and 2 Kbit for routing and refresh packets. Each node is assumed to have an initial energy of 5 joules. The buffers for real-rime data and normal data have default size of 20 packets [30]. A node is considered non-functional if its energy level reaches 0. For the term CF_1 in the cost function, we used the linear discharge curve of the alkaline battery [31].

For a node in the sensing state, packets are generated at a constant rate of 1 packet/sec. This value is consistent with the specifications of the Acoustic Ballistic Module from SenTech Inc. [13]. The real-time packet generation rate (l_{RT}) for the nodes, which have imaging capability is greater than the normal rate. The default value is 6 packets/sec. A service rate (m) of 20 packets/sec is assumed. Each data packet is time-stamped when it is generated to allow the calculation of average delay per packet. In addition, each packet has an energy field that is updated during the packet transmission to calculate the average energy per packet since our cost function defined for each link is using remaining energy as part of the cost. A packet drop probability is taken to be 0.01. This is used to make the simulator more realistic and to simulate the deviation of the gateway energy model from the actual energy model of nodes. We assume that the network is tasked with a target-tracking mission in the experiment. The initial set of sensing nodes is chosen to be the nodes on the convex hull of sensors in the deployment area. The set of sensing nodes changes as the target moves. Since targets are assumed to come from outside the area, the sensing circuitry of all boundary nodes is always turned on. The sensing circuitry of other nodes are usually turned off but can be turned on according to the target movement. We also assume that each sensor node is capable of taking the image of a target to identify it clearly and can turn on its imaging capability on demand. During simulation, a small subset of current active nodes, which are the closest nodes to the target, are selected to turn on their imaging capability. Therefore, the imaging sensor set may change with the movement of the target.

The packet-generation rate for imaging sensors is bigger than the normal sensors; hence more packets are generated when imaging sensors are employed. These packets are labeled as real-time packets and treated differently in sensor nodes. The *r*-value is initially assumed to be 0 but it is recalculated as imaging sensors get activated. The default end-to-end delay requirement for a QoS path is taken to be 0.8 sec [32]. Targets are assumed to start at a random position outside the convex hull. Targets are characterized by having a constant speed chosen uniformly from the range 4 meters/s to 6 meters/s and a constant direction chosen uniformly depending on the initial target position in order for the target to cross the convex hull region. It is assumed that only one target is active at a time. This target remains active until it leaves the deployment region. In this case, a new target is generated.

3.3 Performance Results

In this section, we present some performance results obtained by the simulation. Different parameters are such as buffer size, packet drop probability and real-time data generation rates are considered in order to capture the effects on the performance metrics defined earlier in this section.

<u>Performance comparison of three different protocols</u>

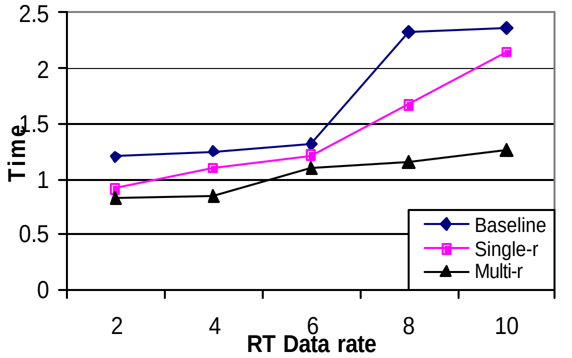

Fig. 5. Average delay per packet with different real-time data rates

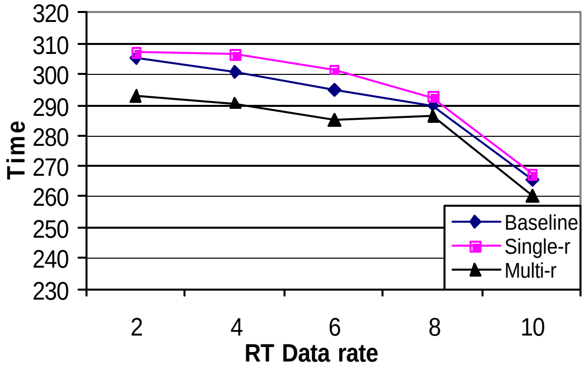

Fig. 6. Average lifetime of a node with different real-time data rates

As a baseline approach, we have used the same cost function with same routing algorithm (i.e. Dijkstra) without doing any service differentiation. That is, we have not differentiated between packets and have used only one queue in each sensor node, which accommodates all kinds of packets. Therefore, no bandwidth sharing on any path is performed. We have compared this approach with our single-*r* and multi-*r* mechanisms by looking at the average delay per packet, average lifetime of a node and time to first node to die. When we compare the average delay per real-time packets generated in our model with the average delay per packet generated in single queue model, we observed that both multi-*r* and single-*r* mechanisms have less average delay (See Fig. 5). This is due to the priority given to real-time packets when transmitting to the gateway. On the other hand, multi-*r* mechanism performs better than single-*r* mechanism as expected. Because, every particular node adjusts its *r*-value based on the resources it has. This is more efficient than the single-*r* case in which a unique *r*-value is imposed by the gateway for all the nodes. Furthermore, in all cases the average delay per packet increases for higher rates and real-time data causes more queuing delay at each sensor node.

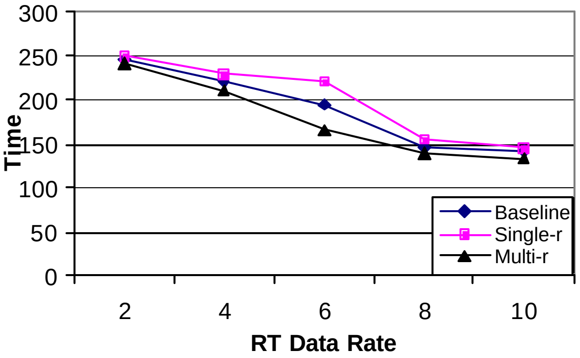

Fig. 7. Time for first node to die with different real-time data rates

In figures 6 and 7, we have looked at the energy usage of the protocols. The average lifetime of a node and the time for first node to die decreases when real-time data increases, causing the nodes to sense and transmit more packets. Since the same cost function is used for all protocols, the lifetime of the nodes and the time for first node to die are very close to each other as confirmed by figures 6 and 7. However, the energy usage of the single-*r* mechanism is slightly less than the others. This can be explained by looking at the throughput. For the single-*r* mechanism, sometimes an *r*-value for the whole network cannot be found; causing the rejection of some connections. This decreases the throughput hence fewer packets are relayed. This is not the case for the baseline protocol. On the other hand, for the multi-*r* mechanism, it is easier to find an *r*-value for a particular node. Furthermore, the efficiency in the usage of resources for multi-*r* mechanism causes an increase in the throughput especially for non-real-time data as seen in figure 8. Such increase incurs a little more energy consumption in the sensor nodes.

<u>Effect of real-time data rate on throughput and delay</u>

In order to study the performance of the algorithm for different real-time data rates, we ran simulation for different values of real-time packet data rates. The results are depicted in figures 8 and 9. First, we have looked at the non-real-time data throughput. While the number of real-time packets increase, it gets more difficult to satisfy increasing number of QoS paths. Hence, this can cause rejection of paths or packet drops for non-real-time data causing throughput for such data to decrease. However, such decrease is very less, becoming constant after a certain point (See figure 8).

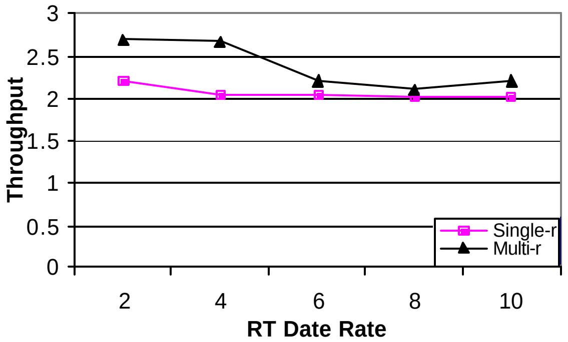

Fig. 8. Non-real-time data throughput for different real-time data rates

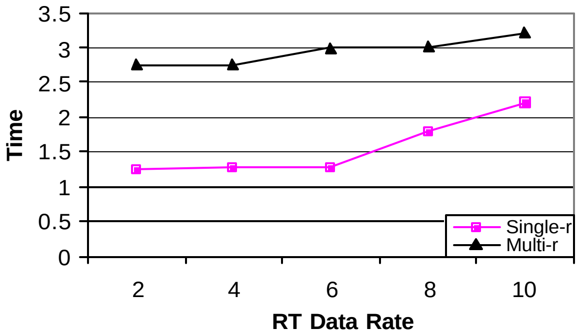

Fig. 9. Non-real-time packet delay for different real-time data rates

We restricted *r*-value to be strictly less than 1, causing the throughput for non-real-time data $((1-r)m)$ to stay greater than 0. Hence, the algorithm does not sacrifice the throughput for non-real-time data for the sake of real-time data. Multi-*r* mechanism has greater throughput than the single-*r* since the resources are handled more efficiently.

Fig. 9 shows the effect of real-time data rate on average delay per non-real-time packet. The delay increases with the rate since packets incur more queuing delay and share the same amount of bandwidth. It is interesting to note that the average packet delay for non-real-time packets in the case of multi-*r* mechanism is bigger than the single-*r* mechanism. In multi-*r* mechanism, the increase in the throughput of non-real-time packets cause extra queuing delay on the nodes; leading non-real-time packets to have more end-to-end delay.

Effect of end-to-end delay requirement and real-time date generation rate on *r*-values

In order to see how the algorithm behaves under stringent conditions, we varied the end-to-end delay requirement and monitored how this change affects the network *r*-value. The results are depicted in figure 10. The network *r*-value goes down while the end-to-end delay requirement gets looser. Since the delay is not too strict, the nodes will be able to meet the end-to-end delay requirement with a smaller *r*-value as

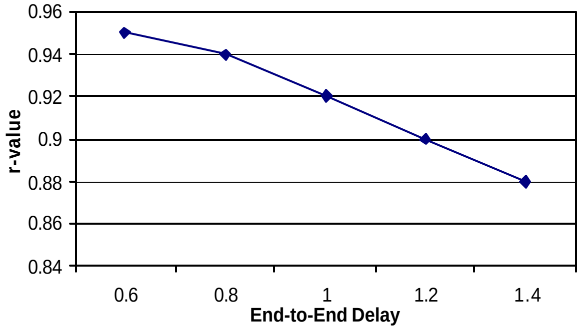

Fig. 10. Network r-value with different end-to-end delay values

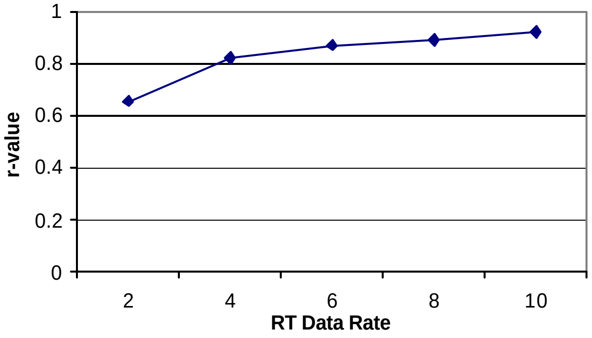

Fig. 11. Network r-value with different real-time data rates

expected from equation 3. On the other hand, while we congest the network with more real-time data packets by increasing the real-time data generation rate, more bandwidth will be required for real-time packets. This will cause the *r*-value to increase so that each node can serve more real-time packets (See figure 11).

Effect of packet drop probability on delay and average lifetime of a node

To study the effect of packet drop probability on performance, we varied the probability of packet drop from 0.01 to 0.05. The results are depicted in figures 12 and 13. The average delay per packet decreases with the increasing probability. This can be explained by noting that as the number of hops the packet traverse increases, the probability that it will be dropped increases. This means that the packets that arrive to the gateway are most probable to take a small number of hops and thus incurring less delay. As expected, the throughput decreases due to lost packets. The average node lifetime increases since not all packets reach their destination and thus the node energy is conserved.

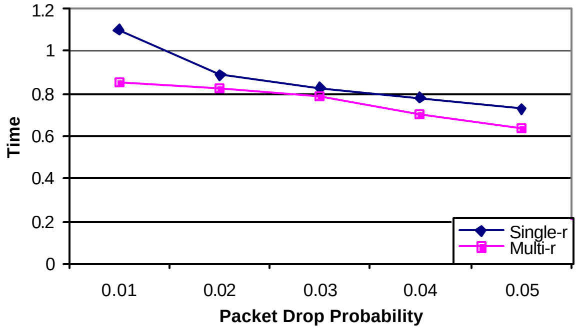

Fig. 12. Average delay per RT packets for different packet drop probabilities

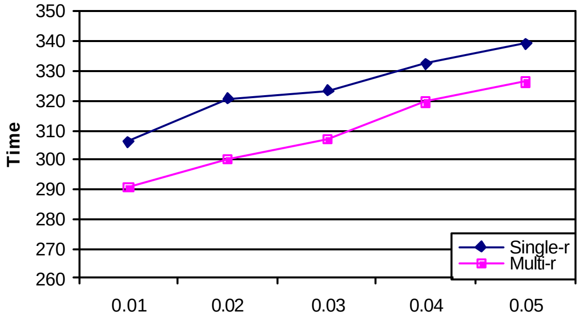

Fig. 13. Average lifetime of a node for different packet drop probabilities

Effect of buffer size on delay and average lifetime of a node

Since, the queuing model we employed uses buffers in each node and there is a limit on the size of those buffers, we varied the buffer size to see if this has any effect on the performance of the algorithm. The results are shown in figures 14 and 15. The average delay per packet increases with the buffer size since the throughput increases. Packets are not dropped when there is enough space in the buffers. This will increase the number of packets arriving to the gateway. The packets from far nodes will be also able

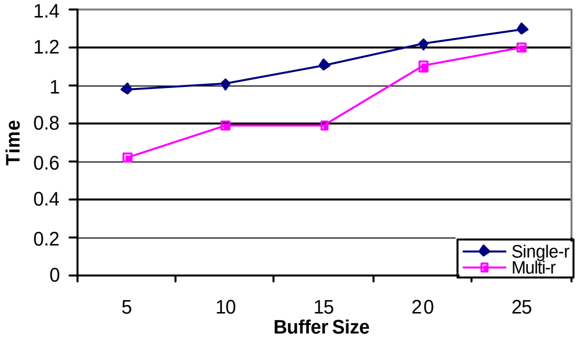

Fig. 14. Average delay per RT packets for different buffer size

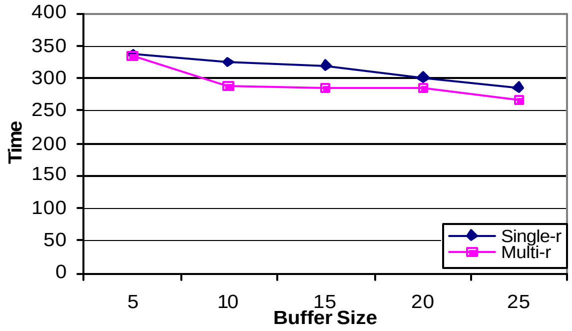

Fig. 15. Average lifetime of a node for different buffer size

to reach the gateway. More packets from far nodes mean more delay, which eventually increases the average delay per packet. The increasing number of packets arriving to the gateway will also increase the energy consumption by increasing the number of transmission and reception costs, therefore decreasing the average lifetime of a node.

It is worth noting that, for both delay and average lifetime metrics, multi-*r* mechanism performs better because of the more efficient adjustment of the packet service rates on sensor nodes as depicted in figures 12, 13, 14 and 15.

4. Conclusion

In this paper, we presented a new energy-aware QoS routing protocol for sensor networks. The protocol finds QoS paths for real-time data with certain end-to-end delay requirements. In order to support both best effort and real-time traffic at the same time, a class-based queuing model is employed. The queuing model allows service sharing for real-time and non-real-time traffic. A ratio *r* is defined as an initial value set by the gateway and is used to calculate the amount of bandwidth to be dedicated to the real-time and non-real-time traffic on a particular outgoing link. The selected queuing model for the protocol allows the throughput for normal data not to diminish by utilizing that service rate on each node.

Two different mechanisms, namely single-*r* and multi-*r*, for setting that service rate on each node are presented. Single-*r* mechanism sets a network wide *r*-value for every sensor node. In the multi-*r* mechanism, the gateway broadcasts the necessary information to the sensor nodes in order for them to calculate their own *r*-value. The effectiveness of the protocol for both mechanisms is validated by simulation. Simulation results have shown that our protocol consistently performs well with respect to QoS metrics, e.g. throughput and average delay, in comparison to a baseline non-QoS aware protocol that use the same link cost. The multi-*r* mechanism has provided better end-to-end delay for real-time packets with a slight increase in energy usage. It has also increased the throughput for non-real-time data packets, which has extended the queuing delay on the nodes causing an increase in the average delay per non-real-time packets.

While our proposed protocol fits a fixed gateway model, we plan on addressing issues related to the relocation and mobility of the gateway under QoS traffic as a future work. In such cases, the frequent update of the position of the gateway and the propagation of that information through the network may excessively drain the energy of nodes. We plan to extend to model in order to handle the overhead of mobility and topology changes.

References

[1] I. F. Akyildiz et al., “Wireless sensor networks: a survey”, *Computer Networks*, Vol. 38, pp. 393-422, March 2002.

[2] D. Estrin et al., “Next Century Challenges: Scalable Coordination in Sensor Networks,” in the *Proc. of MobiCom’99*, Seattle, Washington, August 1999.

[3] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Communications of the ACM,* Vol. 43, No 5, pp. 51 – 58, May 2000.

[4] K. Sohrabi et al., "Protocols for self-organization of a wireless sensor network,” *IEEE Personal Communications*, Vol. 7, No. 5, pp. 16-27, October 2000.

[5] R. Min et al., "Low Power Wireless Sensor Networks", in the *Proc. of Int. Conference on VLSI Design*, Bangalore, India, January 2001.

[6] J.M. Rabaey et al., "PicoRadio supports ad hoc ultra low power wireless networking," *IEEE Computer*, Vol. 33, pp. 42-48, July 2000.

[7] R. H. Katz et al., “Mobile Networking for Smart Dust,” in the *Proc. of MobiCom’99*, Seattle, WA, August 1999.

[8] W. Heinzelman et al., “Adaptive protocols for information dissemination in wireless sensor networks,” in the *Proc. of MobiCom’99*, Seattle, WA, August 1999.

[9] C. Intanagonwiwat et al., "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the *Proc. of MobiCom'00*, Boston, MA, August 2000.

[10] R. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", in the *Proc. of IEEE Wireless Communications and Networking Conference*, Orlando, FL, March 2002.

[11] W. Heinzelman et al., “Energy-Efficient Communication Protocols for Wireless Microsensor Networks,” in *Hawaii International Conference on System Sciences*, January 2000.

[12] M. Younis et al., “Energy-Aware Routing in Cluster-Based Sensor Networks”, in the *Proc. of MASCOTS2002*, Fort Worth, TX, October 2002.

[13] "Data sheet for the Acoustic Ballistic Module", SenTech Inc., http://www.sentech-acoustic.com/

[14] A. Buczak and V. Jamalabad, "Self-organization of a Heterogeneous Sensor Network by Genetic Algorithms," *Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et al. (eds.), Vol. 8, pp. 259-264, ASME Press, New York, 1998.

[15] W. C. Lee et al., "Routing Subject to Quality of Service Constraints Integrated Communication Networks," *IEEE Network*, July/Aug. 1995.

[16] Z. Wang and J. Crowcraft, "QoS-based Routing for Supporting Resource Reservation," *IEEE Journal on Selected Area of Communications*, Sept 1996.

[17] Q. Ma and P. Steenkiste, "Quality-of-Service routing with Performance Guarantees," in the *Proc. of the 4th Int.Workshop on Quality of Service*, May 1997.

[18] L. Zhang et al., "RSVP: A New Resource ReServation Protocol," *IEEE Network*, Sept 1993.

[19] E. Crowley et al., "A framework for QoS based routing in the Internet," Internet-draft, draft-ietf-qosr-framework-06.txt, Aug. 1998.

[20] R. Querin and A. Orda, "QoS-based routing in networks with inaccurate information: Theory and algorithms," in *Proc. IEEE INFOCOM'97*, Japan, pp. 75-83, 1997.

[21] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in ad-hoc Networks," *IEEE Journal on Selected areas in Communications*, Vol. 17, No. 8, August 1999.

[22] R.Sivakumar et al., "Core extraction distributed ad hoc routing specification," IETF Internet draft draft-ietf-manet-cedar-spec-00.txt, 1998.

[23] C.R. Lin, "On Demand QoS routing in Multihop Mobile Networks," *IEICE Transactions on Communications*, July 2000.

[24] C. Zhu and M. S. Corson, "QoS routing for mobile ad hoc networks," *in the Proc. of IEEE INFOCOM*, 2002.

[25] G. Feng et al., "Performance Evaluation of Delay-Constrained Least-Cost Routing Algorithms Based on Linear and Nonlinear Lagrange Relaxation," in the *Proc. of ICC'2002*, New York, April 2002.

[26] S. Floyd and V. Jacobson, "Link Sharing and Resource Management Models for Packet Networks," *IEEE/ACM Transactions on Networking*, Vol. 3 No. 4 pp.365-386, August 1995.

[27] D. Julian et al., "QoS and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," in the *Proc. of IEEE INFOCOM*, NY, June 2002.

[28] Q.V.M. Ernesto et al., "The K shortest paths problem," *Research Report, CISUC*, June 1998.

[29] J. Andresen et al., "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, Vol. 33, No. 1, January 1995

[30] M. Gerla et al., "Wireless, Mobile Ad-Hoc Network Routing," in the *Proc. of IEEE/ACM FOCUS'99*, New Brunswick, NJ, May 1999.

[31] M. Woo et al., "Power aware routing in mobile ad hoc networks", in the *Proc. of MobiCom'98*, Dallas, TX, 1998

[32] K. Danilidis et al., "Real-time tracking of moving objects with an active camera," *Real-Time Imaging Journal*, Vol. 4, No.1 pp.3-20. February 1998.