An Energy-Aware QoS Routing Protocol for 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

Recent advances in wireless sensor networks have led to many new routing protocols specifically designed for sensor networks. Almost all of these routing protocols considered energy efficiency as the ultimate objective in order to maximize the whole network lifetime. However, the introduction of video and imaging sensors has posed additional challenges. Transmission of video and 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, 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, throughput for non-real-time data is maximized by adjusting the service rate for both real-time and non-real-time data at sensor nodes. Simulation results have demonstrated the effectiveness of our approach.

1. Introduction

Recent advances in miniaturization and low-power design have led to active research in large-scale, highly distributed systems of small-size, wireless unattended sensors. Each sensor is capable of detecting ambient conditions such as temperature, sound, or the presence of certain objects. Over the recent few years, the design of sensor networks has gained increasing importance due to their potential for some civil and military applications such as combat field surveillance, security and disaster management. These systems process data gathered from multiple sensors to monitor events in an area of interest. For example, in a disaster management setup a large number of sensors can be dropped by a helicopter. Networking these sensors can assist rescue operations by locating survivors, identifying risky areas and making the rescue crew more aware of the overall situation.

Routing of sensor data has been one of the challenging areas in wireless sensor network research [1,2]. 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. However, the development of video and imaging sensors requires the consideration of quality of service (QoS), which magnifies the difficulties associated with the energy limitation.

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 and/or video sensors. After locating and detecting the target, those sensors can be turned on in order to identify and track that. This requires a real-time data exchange between sensors and controller in order to take the proper actions. However, dealing with real-time multimedia data requires certain bandwidth with minimum possible delay, and jitter. Therefore, a service differentiation mechanism is needed for guaranteeing the reliable delivery QoS traffic.

In this paper, we present an energy-aware QoS routing mechanism for wireless sensor networks. Our proposed protocol extends the routing approach in [13] and considers only end-to-end delay. The protocol looks for a delay-constrained path with the least possible cost based on a cost function 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. To the best of our knowledge, QoS routing has not been addressed in the context of sensor networks.

In the balance of this section we describe our sensor network architecture 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

We consider the sensor network architecture depicted in Figure 1. In the architecture sensor nodes are grouped into clusters controlled by a single command node. Sensors are only capable of radio-based short-haul communication and are responsible for probing the environment to detect a target/event. Every cluster has a gateway node that manages sensors in the cluster. Clusters can be formed based on many criteria such as communication range, number and type of sensors and geographical location. In this paper, we assume that sensor and gateway nodes are stationary and the gateway

Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)
0-7695-1921-0/03 $17.00 © 2003 IEEE

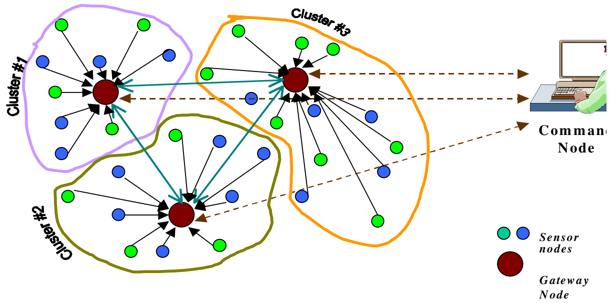

Figure 1. Multi-gateway clustered network sensors

node is located within the communication range of all the sensors of its cluster. Clustering the sensor network is performed by the command node and is beyond the scope of this paper. The command node will inform each gateway node of the ID and location of sensors in its cluster.

Sensors receive commands from and send readings to its gateway node, which processes these readings. Gateways can track events or targets using readings from sensors in any clusters as deemed by the command node. However, sensors that belong to a particular cluster are only accessible via the gateway of that cluster. Therefore, a gateway should be able to route sensor data to other gateways. Gateway nodes interface the command node with the sensor network via long-haul communication links. In this paper only focuses on the QoS routing of data within one particular cluster.

1.2. Related Work

In traditional best-effort routing throughput and average response time are the main concerns. 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 [3,4,5], they cannot be directly applied to wireless sensor networks due to resource limitation. On the other hand, a number of protocols have been proposed for QoS routing in wireless ad-hoc networks taking the dynamic nature of the network into account [6,7,8,9,10]. However, none of these protocols consider energy awareness along with the QoS parameters. Some of the proposed protocols consider the imprecise state information while determining the routes [6,7]. Our model is not subject imprecision since the state of sensor nodes are maintained by the gateway node.

CEDAR is another QoS aware protocol, which uses the idea of core nodes (dominating set) of the network while determining the paths [8]. Using routes found through the network core, a QoS path can be easily found. Since, the data flow in our architecture is many-to-one; there is no need to find a network core. Moreover, if any node in the core is broken, it will cost too much resource to reconstruct the core. Lin [9] and Zhu et al. [10] 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.

The only protocol for sensor networks that includes the notion of QoS in its routing decisions is the Sequential Assignment Routing (SAR) [11]. The SAR protocol creates trees routed from one-hop neighbor of the sink by taking QoS metric, energy resource on each path and 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 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 nodes state at each sensor. Our protocol does not involve sensors in route setup.

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. Delay constraints are associated only with real-time data. Note that, in this case, both real-time and non-real-time traffic coexist 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 constrained path optimization (CPO) problems, which are proved to be NP-complete [14]. There is also 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 model [12]. We use different queues for the two different types of traffic. 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 Figure 2.

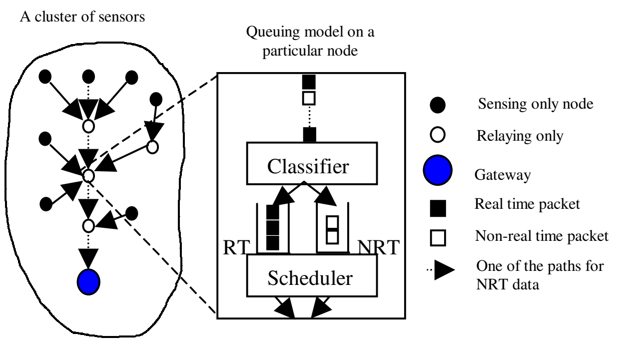

Figure 2. Queuing model in cluster-based sensor network

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 in case of a congestion. Both classes can borrow bandwidth from each other when one of the two types of the traffic is non-existent or under the limits. The service rate of real-time and non-real-time traffic on that sensor node "*i*" are with $r_i \mu$ and $(1 - r_i)\mu$ respectively. 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.

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 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 then we might further check them to see which one meets our end-to-end QoS requirements by trying to find an *r*-value that also maximizes 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. We define the following cost function for a link between nodes *i* and *j*:

$$cost_{ij} = \sum_{k=0}^{3} CF_k = c_0\,(dist_{ij})^1 + c_1\,f(energy_j) + c_2/T_j + c_3\,f(e_{ij})$$

- $dist_{ij}$ is the distance between the nodes *i* and *j*,
- $f(energy_j)$ is the function for finding current residual energy of node *j*,
- T_j is the expected time under the current consumption rate for the node j energy level reaches the minimum acceptable threshold.
- $f(e_{ij})$ is the function for finding the error rate on the link between *i* and *j*.

2.3. Calculation 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, which consists of queuing delay and propagation delay for a particular path *P*, we introduce the notation below:

λ_{RT}	:	Data generation rate for imaging/video sensors
p_i	:	No of sense-only neighbors of node i on path P
q_i	:	No of relay-only neighbors of node i on path P
$\lambda_{RT}^{(i)}$	:	Total real-time data rate on sensor node i
$TQ_{RT}^{(i)}$	:	Queuing delay on a node i for real-time traffic
T_E	:	End-to-end queuing delay for path P
T_p	:	End-to-end propagation delay for path P
$T_{end\text{-}end}$	:	Total end-to-end delay for 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 only nodes using path P to send data to the gateway

Total real-time data rate by p_i nodes will be $p_i \times \lambda_{RT}$ and total real-time data rate by q_i nodes will be added recursively for each relaying only node (in worst-case every relaying-only node produces real-time data by the rate λ_{RT}). Therefore, total real-time data load on a sensor node is:

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

The average waiting time including the service time in the queue is: $W = 1/(\mu-\lambda)$. Hence, total queuing delay (including the service time), on a node "*i*" is: $TQ_{RT}^{(i)} = 1/(r_i \mu - \lambda_{RT}^{(i)})$.

The end-to-end queuing delay for a particular path is:

$$T_E = \sum_{i \in Path} TQ_{RT}^{(i)} = \sum_{i \in Path} 1/[r_i \mu - p_i \lambda_{RT} - \sum_{j=1}^{q_i} p_j \lambda_{RT}^{(j)}]$$

The end-to-end propagation delay for the path is :

$$T_p = \sum_{i,j \in Path} c \times dist_{ij}$$

where ***C*** is a constant obtained by dividing a weighting constant by the speed of wireless transmission. Thus, total end-to-end delay will be:

$$T_{end\text{-}end} = T_E + T_p = \sum_{i \in Path} \frac{1}{r_i \mu - p_i \lambda_{RT} - \sum_{j=1}^{q_i} p_j \lambda_{RT}^{(j)}} + \sum_{i,j \in Path} c \times dist_{ij}$$

2.4. Routing Algorithm

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 introduce 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:

$$T_{end-end} = \sum_{i\in Path} \frac{1}{r\mu - p_i\lambda_{RT} - \sum_{j=1}^{q_i} p_j \lambda_{RT}^{(j)}} + \sum_{i,j\in Path} c\times dist_{ij}$$

Thus, the optimizations problem is stated as as follows:

$$Max\left(\sum_{i\in Path} ((1-r)\mu) \right) \text{ with } T_{end\text{-}end} \le T_{required} \text{ and } 0 \le r < 1$$

In order to find *r* from the inequality $T_{end\text{-}end} \le T_{required}$, we divide $T_{required}$ into *m* equal time slots, where *m* is the number of nodes on a particular path. Since the last node will be getting the actual longest queuing delay, we consider finding an *r*-value that satisfies the last node's delay. As a consequence, the delay other nodes on the path will be less than $T_{required}/m$. The value of *r* can be calculated using:

$$\frac{T_{required}}{m} = TQ_{RT}^{n} = \frac{1}{r\mu - \lambda_{RT}(p_m + \sum_{k\in Nodes} p_k)} + \sum_{i,j\in Path} c\times dist_{ij}$$

By considering the optimization problem above, we propose the algorithm shown in Figure 3 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 based on the cost function defined in section 2.2. 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/video sensors to the gateway. For each sensor node that has imaging/video capability, an *r*-value is calculated on the current path (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 feasible *r*-value, the connection of that node to the gateway is simply 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 satisfies the end-to-end delay requirement for all paths established to the gateway.

In order to find the *K*-least cost paths, we modified an extended version of Dijkstra's algorithm given in [15]. Since, the algorithm be trapped in cycles during execution; we modified the algorithm in order to avoid cycles and ensure simplicity and efficiency. Each time a new path is searched for a particular node, only node-disjoint paths are considered during the process. This might also help finding a proper *r*-value easily since that node-disjoint path will not inherit the congestion in the former path.

1 Calculate $\cos t_{ij}, \forall i, j \in V$
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 $T_{end-end}(p_i) \le T_{required}$ (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 $\left(P_i^K\right)$ to the gateway
10   **for each** $k \in K$ **do**
11   Recompute *r* from $T_{end-end}(p_i^k) \le T_{required}$
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

Figure 3. Pseudo code for the proposed algorithm

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

- *Time to network partition*: When the first node runs out of energy, the network is said to be partitioned.
- *Average lifetime of a node*: This gives a good measure of the network lifetime.
- *Average delay per packet*: Defined as the average time a packet takes from a sensor node to the gateway.
- *Network Throughput*: Defined as number of data packets received at the gateway divided by the simulation time.

3.2. Environment Setup

In the experiments the cluster consists of 100 randomly placed nodes in a 1000×1000 meter square area. The gateway position is determined randomly within the cluster boundaries. A free space propagation channel model is assumed [17] 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 15 packets [18]. 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 [16].

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. [19]. The real-time packet generation rate (λ_{RT}) unless changed in the experiment, is set to 3 packets/sec. A service rate (μ) of 5 packets/sec is assumed. Each 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. A packet drop probability is taken to be 0.01.

We assume that the cluster 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 cluster. The set of sensing nodes changes as the target moves. Since targets are assumed to come from outside the cluster, 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 can turn on its imaging capability on demand. During simulation, a small subset of the closest nodes to the target uses the imaging capability. Such subset changes as the target moves. The r-value is initially assumed to be 0 but it is recalculated when imaging sensors get activated. The default end-to-end delay requirement for a QoS path is assumed to be 10 seconds, which is a reasonable amount of time to get image data periodically in a real-time target tracking application. 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. It is assumed that only one target is active at a time. This target remains active until it leaves the deployment region area.

3.3. Performance Results

Different parameters are considered for end-to-end delay and real-time data generation to capture the effects on the performance metrics defined earlier in this section.

In order to see how the algorithm behaves under stringent conditions, we varied the end-to-end delay and monitored how this change affects the network *r*-value. The results are depicted in Figure 4. The network *r*-value goes down while the end-to-end delay requirement gets looser. Since the delay is not too strict, most nodes will be able to find a QoS path. 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 5).

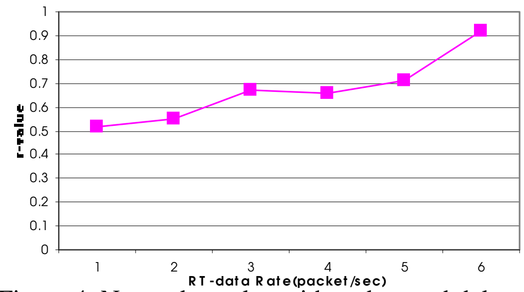

Figure 4. Network r-value with end-to-end delay values

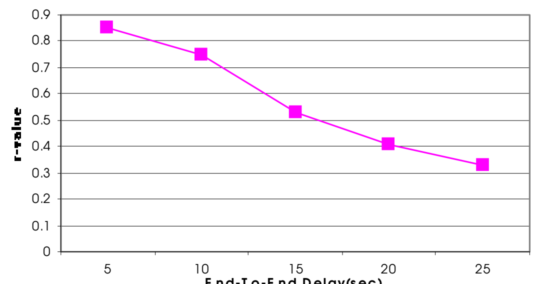

Figure 5. Network r-value with real-time data rates

The performance of the algorithm for different values of λ_{RT}, are depicted in Figure 6, 7 and 8. 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 real-time data causing throughput for such data to decrease. However, the throughput for non-real-time data does not change much since there is already a constant dedicated bandwidth for such data, ensured by the *r*-value. We restricted *r*-value to be strictly less than 1, causing the throughput for non-real-time data (1–r)μ always greater than 0 (See Figure 6). Figure 7 shows the effect of λ_{RT} on average delay per packet. The delay increases with the rate since packets incur more queuing delay and share the same amount of bandwidth. We also looked for the lifetime of a node in order to see the effect of λ_{RT} on the energy metric. Figure 8 shows that the average lifetime of a node and time for first node to die increase with the λ_{RT}. The reason for this increase is that the throughput decreases, causing the number of packets arriving to the gateway to decrease. Therefore, fewer packets will be relayed by the sensor nodes, which will save energy from transmission and reception energy costs.

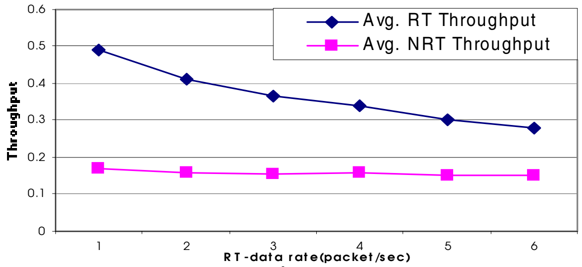

Figure 6. Effect of λ_{RT} on packet throughput

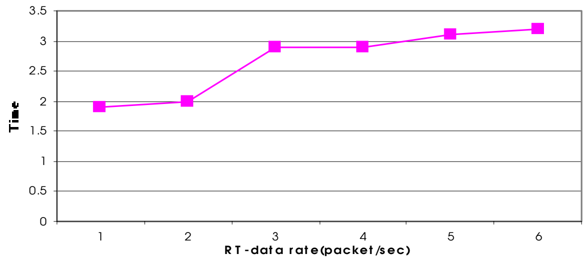

Figure 7. Effect of λ_{RT} on average delay per packet

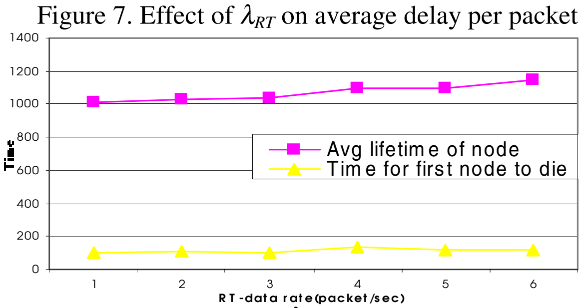

Figure 8. Effect of λ_{RT} on lifetime of nodes

4. Conclusions and Future Work

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. Moreover, the selected queuing model for the protocol allows the throughput for normal data not to diminish by employing a network wide *r*-value, which guarantees certain service rate for real-time and non-real-time data on each link. The effectiveness of the protocol is validated by simulation. Simulation results show that our protocol consistently performs well with respect to QoS metrics, e.g. throughput and average delay as well as energy-based metric such as average lifetime of a node. We are currently extending the model to allow different *r*-values can be assigned to sensor nodes and plan to examine the performance of such extended model.

References

[1] C. Intanagonwiwat et al., "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in the *Proceedings of MobiCOM '00*, MA, 2000.

[2] W. Heinzelman et al., "Energy-efficient communication protocol for wireless sensor networks," in the *Proceedings of Hawaii Intl. Conf. System Sciences, Hawaii*, 2000.

[3] W. Lee et al., "Routing Subject to Quality of Service Constraints Integrated Communication Networks," *IEEE Network*, 1995.

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

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

[6] R. Querin and A. Orda, "QoS-based routing in networks with inaccurate information: Theory and algorithms," in the *Proceedings of IEEE INFOCOM'97*, Japan, 1997.

[7] 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.

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

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

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

[11] K. Sohrabi et al., "Protocols for self-organization of a wireless sensor network," *IEEE Personal Comm.*, pp.16-27, Oct. 2000.

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

[13] M. Younis et al., "Energy-Aware Routing in Cluster-Based Sensor Networks", *Proc. of MASCOTS'2002,* TX, 2002.

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

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

[16] S. Singh et al., "Power-Aware Routing in Mobile Ad-Hoc Networks," in the *Proceedings of MOBICOM'98*, TX, 1998.

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

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

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