

Energy-aware Delay-Constrained 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. Recent years has witnessed a growing interest in the applications of unattended wireless sensor networks. Typically sensors collect data about their surrounding and forward that data to a gateway (sink). Due to the limitation of the energy supply that these miniaturized sensors have, almost all of the proposed routing protocols have aimed at energy efficiency as the ultimate objective and considered relaying data to a stationary gateway. However, many new issues have been posed by the increasing interest in applications that demand certain quality of service (QoS) guarantees. Such issues even get more challenging when the gateway is mobile. In this paper, we propose an energy-aware approach for routing delay-constrained data. The approach pursues multi-hop packet relaying to minimize transmission energy and employs Weighted Fair Queuing (WFQ) packet scheduling methodology along with leaky bucket constrained data sources in order to provide soft real-time guarantees for data delivery. Such employment of WFQ at each node provides a service differentiation between two different classes of traffic, namely real-time and non-real-time traffic and divides the outgoing link capacity among such classes accordingly. In the case of a mobile gateway, uninterrupted data flow for both types of traffic is achieved by dynamic adjustment of the route setup to react to the gateway's departure out of transmission range of relaying nodes. Simulation results demonstrate the effectiveness of the proposed approach for popular performance metrics.

1 Introduction

Recent advances in hardware and wireless networks technologies have led to the development of micro sensors that can be deployed in sheer numbers in order to collaborate in an ad hoc manner on data gathering and processing [1][2][3][4][5][6]. Networking unattended wireless sensors are expected to have significant impact on the efficiency of many military and civil applications such as combat field surveillance, security and disaster management. For instance, in a disaster management setup networking these sensors can assist rescue operations by locating survivors, identifying risky areas and making the rescue crew more aware of the overall situation. On the military side, applications of sensor networks are numerous. For example, the use of networked set of sensors can limit the need for personnel involvement in the usually dangerous reconnaissance missions. In addition, sensor networks can enable a more civic use of landmines by making them remotely controllable and target-specific in order to prevent harming civilians and animals. Security applications of sensor networks include intrusion detection and criminal hunting.

Sensor nodes used in unattended setups are usually miniaturized and constrained in energy supply. Such constraint has necessitated energy-awareness at most layers of networking protocol stack including the network layer. In addition, many applications require the deployment of large number of sensor nodes making it impractical to build a global addressing scheme. Moreover, in contrary to typical communication networks almost all applications of sensor networks require the flow of sensed data from multiple regions (sources) to a particular sink. These unique characteristics of sensor networks have made efficient routing of sensor data one of the technical challenges in wireless sensor networks. To address such challenges, the bulk of research on routing in wireless sensor networks mostly aim at maximizing the lifetime of the network, allowing scalability for large number of sensor nodes and supporting tolerance for sensor's damage and battery exhaustion [7][8][9][10][11][12]. These performance objectives have been deemed sufficient for applications, which do not require on-time response or for which data are not collected at high rate.

However, there has been an increasing interest in sensor networks applications that require certain performance guarantees such as end-to-end delay, bandwidth etc. For instance, routing of imaging data in a battle environment requires careful handling in order to ensure that the end-to-end delay is within acceptable

range and the images are received properly without any distortion. Other typical applications include real-time target tracking, emergent event triggering in monitoring applications and critical information relaying in emergency applications. Since most of the current protocols do not provide performance guarantees for such applications, new routing protocols that can achieve desired Quality of Service (QoS) for the delivery of sensing data are needed. Given the energy constraints of sensor nodes, QoS routing in sensor networks introduces several new challenges at the network layer. The employed routing protocol should not only ensure bounded delay or bandwidth through the duration of a connection but also provide the use of most energy efficient path.

Moreover, most published research on data routing in sensor networks have assumed a stationary sink (gateway) node. However, in numerous applications the mobility of the gateway can be desirable. One of the possible configurations of a mobile gateway can be a computer installed on a moving object. Examples of this configuration includes an emergency vehicle equipped with computing and communication devices in the context of a disaster management application and a walking soldier with a laptop computer in his backpack in a battle environment. Providing performance guarantees when the gateway is mobile brings extra challenges. The gateway in such setup changes its location and thus the network topology will be modified so frequently that the increased control message traffic will dominate the total energy consumption of the network. On the other hand, being irresponsive to changes in the gateway location can lead to high packet losses and negatively affect the application of the sensor network.

In this paper, we present a novel energy-efficient routing mechanism to ensure bounded delay for the data delivery in sensor networks. Our proposed approach sets up energy-aware multi-hop data paths considering sensor's transmission power and sensor energy reserve and imposes end-to-end delay as a constraint. End-to-end delay bound is achieved through the use of a Weighted Fair Queuing (WFQ) based packet scheduling technique in each sensor node [13][14]. WFQ considers a different queue for each incoming flow and has been shown to provide, in statistical term, an upper bound on path delay for a leaky bucket constrained flow [15]. We regulate the incoming traffic from the sources by using the leaky-bucket traffic-regulation mechanism and separate the real-time (delay-constrained) traffic from non-constrained traffic with the usage of two different queues in each node. Once such separation is achieved, the service rate for the real-time data queue is estimated so that most deadlines are met.

We further extend the proposed routing mechanism to provide uninterrupted message delivery for a mobile gateway. The main idea is to watch for gateway's reachability to sensor nodes and dynamically adjust the network topology in order to ensure timely data delivery. The energy efficient multi-hop routes set at the initial gateway location for both real-time and non-real-time traffic are used as long as the gateway is within the transmission range of last hop sensors i.e., sensors that are within one hop distance away from the gateway. When the gateway goes out of range of last hop sensors, it designates new sensor nodes as forwarders and therefore continual delivery of messages is maintained. Such process continuous until establishing new routes is deemed necessary. Rerouting is triggered when the end-to-end delay is not acceptable or current routes become instable due to the exhaustion of the battery of some nodes on the data paths. Our approach balances the energy and timeless goals of the network and prevents the potential excessive topology management overhead.

In the balance of this section we describe the sensor network architecture that we consider and summarize the related work. We discuss our WFQ based approach for achieving on-time delivery of real-time data in section 2. In Section 3, we explain how to extend the approach to handle possible mobility of the gateway with minimal impact on the timeliness of real-time packets. Section 4 discusses performance evaluation and simulation results. Finally we conclude the paper in section 5.

1.1 System Model

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 and are empowered with limited data processing engines. The mission for these sensors is dynamically changing to serve the need of a command center. A gateway node, which is significantly less energy-constrained than the sensors, is deployed in the physical proximity of sensors. The gateway is assumed to know the geographical location of deployed sensors. The gateway is responsible for

organizing the activities at sensor nodes contingent to achieving a mission, fusing data collected by sensor nodes, coordinating communication among nodes and interacting with command node. The gateway node sends to the command node reports generated through fusion of sensor readings, e.g. tracks of detected targets. The command node presents these reports to the user and performs system-level fusion of the collected reports for overall situation awareness. The system architecture for the sensor network is depicted in Fig. 1.

While sensor nodes are stationary, we are considering two different models for the gateway: fixed and fully mobile. Sensors are assumed to be within the communication range of the gateway node. 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 worth noting that most of these capabilities are available on some of the advanced sensors, e.g. the Acoustic Ballistic Module from SenTech Inc. [16]. It is also assumed that the sensor can act as a relay to forward data from another sensor. We refer to the gateway's selection of a subset of the sensors for probing the environment as network organization and to the data routing and medium access arbitration as network management.

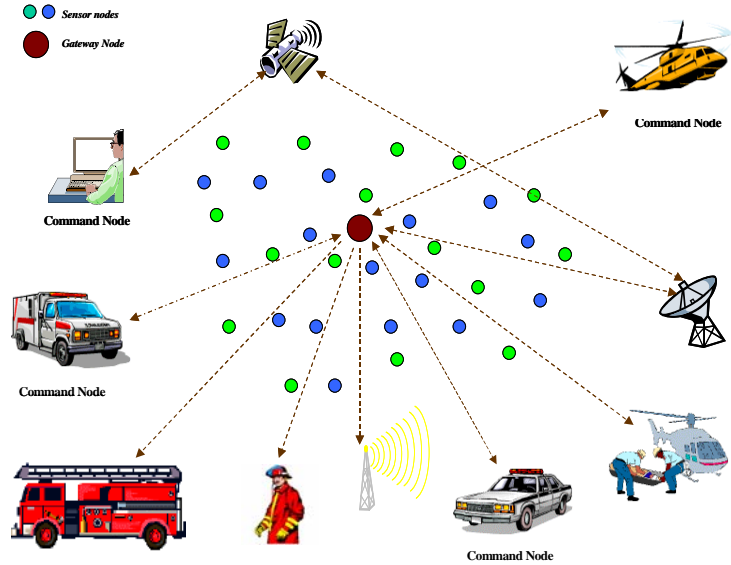


Fig. 1: Three-tier sensor network architecture

1.2 Related Work

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 data in wire-based and wireless networks [17][18][19][20], they cannot be directly applied to wireless sensor networks due to several characteristics that distinguish them from contemporary communication and wireless ad-hoc networks. For instance, it is not possible to build a global addressing scheme for the deployment of sheer number of sensor nodes. Therefore, classical IP-based QoS protocols cannot be applied to sensor networks. Moreover, sensor nodes are tightly constrained in terms of transmission power, on-board energy, processing capacity and storage and thus require careful resource management. Due to such issues, non-conventional QoS routing protocols are needed for wireless sensor networks.

Very little research has been done on QoS routing in wireless sensor networks. The first protocol for wireless sensor networks that includes the 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 the sink to the sensors are formed. One of these paths is selected according to the energy resources and achievable QoS on each path. SAR maintains multiple paths from the nodes to the sink. Although, this ensures fault-tolerance, the protocol suffers from the overhead of maintaining the tables and states at each sensor node especially when the number of nodes is very large. Our approach does not require sensor's involvement in route setup therefore does not introduce an extra overhead to the sensor nodes.

Another QoS routing protocol for sensor networks that provides soft real-time end-to-end guarantees is SPEED [21]. The protocol requires each node to maintain information about its neighbors and uses

geographic forwarding to find the paths. SPEED strives to ensure a certain speed for each packet in the network so that each application can estimate the end-to-end packet delay by dividing the distance to the sink by the speed of the packet before making the admission decision. Moreover, it can provide congestion avoidance when the network is heavily loaded. However, SPEED tries to save energy only through reducing the number of control packets. It does not consider any special energy metric in its routing protocol. Our approach on the other hand, initially finds a list of energy-efficient paths based on the required transmission and residual energy of the nodes. A delay-constrained path is then searched among those alternatives.

None of the described protocols considered the mobility of the sink node and assumed a stationary sink. The only work to consider mobile sinks is reported in [22]. The protocol initially builds a grid structure that divides the network into cells with several dissemination nodes. The dissemination nodes are responsible for relaying the query to the proper sources and the data to the sink. Once a source starts generating data, the path for that data is forwarded to the dissemination nodes. When such data is needed by a sink, the queries are first flooded locally within the cell. If it can not be served within the cell, the dissemination node of that cell handles the query by directing it to the source of data. The dissemination nodes take care of both query and data traffic, eliminating the need for flooding the entire network. In order to support mobility, each sink node selects a primary agent (within the cell) and an immediate agent (within one hop distance) to get data forwarded to it while it is on the move. While the protocol aims at minimizing energy consumption due to the motion of the sink node and maximizing the network throughput, it does not consider any QoS attributes while setting the routes. To the best of our knowledge, no research has been done to provide QoS routing to a mobile sink in wireless sensor networks. Our approach not only provides soft real-time bounds for real-time data, but it also addresses the challenges of handling sink mobility.

2 Energy-aware QoS Routing via WFQ

In order to describe the routing problem, we consider the following scenario: In a battle environment it is crucial to locate, detect and identify a target. Target identification and monitoring can be performed through the employment of imaging sensors. After locating and detecting the target using acoustic sensors, one can turn on imaging sensors to capture images of the target and periodically send them to the gateway. Since, it is a battle environment; this requires a real-time data exchange between the sensors and the controller in order to take the proper actions. In that case, we should deal with real-time data, which requires minimum level of bandwidth and bounded delay. Therefore, a service differentiation mechanism is needed in order to guarantee the reliable delivery of such data.

The goal in such an application will be to find an optimal path to the gateway in terms of energy consumption and error rate while meeting the end-to-end delay requirement. End-to-end delay requirement is associated only with the real-time data. The described QoS routing problem is very similar to typical path constrained path optimization (PCPO) problems, which are proved to be NP-hard [23]. We are trying to find least-cost path, which meets the end-to-end delay path constraint. Our approach is based on associating a cost function for each link and using a K least cost path algorithm to find a set of candidate routes. Such routes are checked against the end-to-end delay constraints and the one that meets the requirements is picked. In order to achieve such delay constraints for real-time traffic, we employ a packet scheduling technique, namely Weighted Fair Queuing (WFQ), at each sensor node to differentiate between delay-constrained and non-constrained packets. In the balance of this section, we will give some background on the utilized packet scheduling technique (WFQ) and explain the details of the proposed algorithm.

2.1 Background on WFQ

In our system model, each node employs a packet scheduling discipline that approximates the Generalized Processor Sharing (GPS). GPS achieves exact weighted max-min fairness by dedicating a separate FIFO queue for each session (flow) and serving an infinitely small amount of data from each queue in a weighted round robin fashion. Before explaining how GPS works in details, we introduce the following notation:

- σ_i : Maximum burst size for leaky bucket on flow i
- ρ_i : Average data rate of the flow i
- $D(i)$: End-to-end delay for flow i
- C : Link bandwidth
- $P_{\max}(i)$: Maximum packet size for flow i
- P_{\max} : Maximum packet size allowed in the network
- g_i^m : Service rate on node m for flow i
- $g(i)$: Minimum of all service rates for flow i
- M : The number of nodes on path of flow i

A GPS server m handles n sessions on a link by giving each session a share of the link based on n positive real numbers, $\Phi_1^m, \Phi_2^m, \dots, \Phi_n^m$. These numbers denote the relative amount of service to each flow on the server m . Note that this sharing is only for backlogged connections since non-backlogged connections already receive what they ask for. The GPS server ensures that backlogged connections share the remaining bandwidth in proportion to the assigned weights. As analyzed in [13], each backlogged connection “ i ” receives a service rate of:

$$g_i^m = \frac{\Phi_i^m}{\sum_{j=1}^n \Phi_j^m} C \quad (1)$$

However, GPS is not implementable in practice due to its ideal fluid model. Therefore, packet approximation algorithms of GPS were proposed. The Weighted Fair Queuing (WFQ) [14] and the Packetized Generalized Processor Sharing (PGPS) [15] are two identical disciplines developed independently and do not require GPS’s infinitely small service assumption. They serve the incoming packets according to their service times under GPS. Therefore, for each flow the packet with the earlier service time is served first. Throughout this paper, we use the term WFQ for the packet-based version of GPS.

WFQ has two important features: First, it can provide fair allocation of bandwidth among all backlogged sessions as long as the total service rate of all sessions is less than the link bandwidth. Second, when combined with traffic regulation, it has been shown to provide an upper bound for the end-to-end delay [15]. Such regulation is done through the use of a leaky bucket mechanism at the sources in order to ensure a constant data rate and to restrict the burst size for the traffic reaching the next relay nodes. As shown in [13], if a session “ i ” is leaky bucket constrained, the amount of session i traffic entering the network during interval $(\tau, t]$ will be:

$$A_i(\tau, t) \leq \sigma_i + \rho_i(t - \tau), \forall t \geq \tau \geq 0$$

Assuming flow i is constrained by a leaky bucket with parameters (σ_i, ρ_i) , the maximum end-to-end delay, introduced by packet queuing and handling till full transmission, for a packet of flow i under WFQ, given in [15], is:

$$D(i) \leq \frac{\sigma_i}{g(i)} + \sum_{m=1}^{M-1} \frac{P_{\max}(i)}{g_i^m} + \sum_{m=1}^M \frac{P_{\max}}{C} \quad (2)$$

2.2 Proposed 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 [24]. We use different queues for the two types of traffic, real-time and non-real-time (normal), 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 a value set by the gateway and is used in allocating the amount of bandwidth to be dedicated to the real-time and non-real-time traffic on a particular outgoing link. As indicated in Fig. 2, this r -value is also used to calculate the service rate for each type of traffic on that particular node, with $r_m\mu$ and $(1-r_m)\mu$ being respectively the service rate for real-time and non-real-time data on sensor node m .

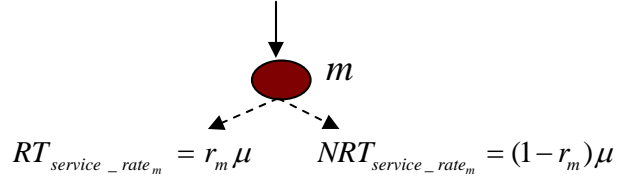


Fig. 2: Link sharing for a sensor node

Since in WFQ each flow has its own queue, we consider each imaging sensor node as a source of different real-time flow, however only one real-time queue is used in the node to serve the data coming from these multiple flows. This mechanism is an approximation to flow-based WFQ approach and is used due to two reasons: First, having a different queue for each real-time flow will be inefficient in terms of the storage capacity of a sensor node. Second, the real-time flows are generated dynamically depending on the number of active imaging sensors. Since the number of such flows can change during the sensing activity, having one queue will reduce the maintenance overhead. In our model, we dedicate another separate queue to serve non-real-time data coming from different sources. The model is depicted in Fig. 3.

In this model, the service ratio r_m for the real-time queue on a node will be the summation of link shares of all real-time flows passing through that node.

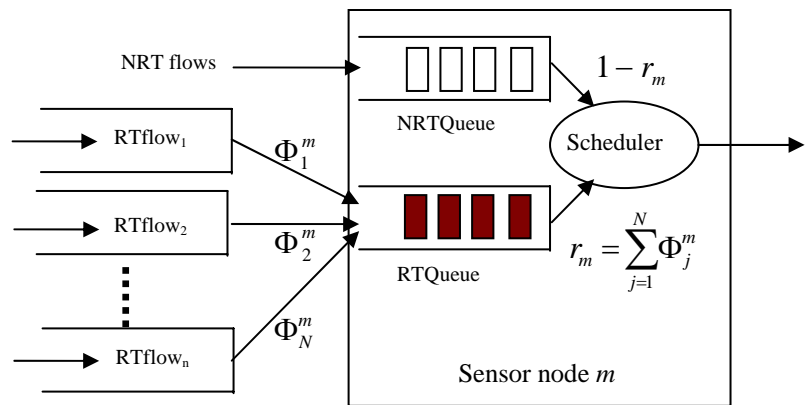


Fig. 3: Queuing model on a sensor node

While delay-constrained delivery of real-time packets is achieved through WFQ, our approach also addresses energy awareness through the usage of a cost function that considers relevant energy factors at sensor nodes. In the next subsection, we describe the gateway role and the handling of energy concerns.

2.3 Achieving Energy Efficiency

The responsibilities of the gateway node include sensor organization and management of the network topology. Sensor organization refers to the appropriate selection of the required sensors that need to be operational in order to achieve the application objective. In addition, the gateway sets up paths for collecting the sensor data. The gateway informs each sensor with its role, namely probing the environment, relaying data or performing both sensing and relaying. Changes in the set of active sensors and/or depletion of the battery of some of these sensors require dynamic adjustment of the network topology. The gateway broadcast the routing table to all sensors prior to starting or resuming data transmission. The gateway will use model-based energy consumption for the data processor, radio transmitter and receiver onboard the sensor in order to track the life of the sensor battery. This model is used in the link cost function as explained below. The gateway updates the sensor energy model with each packet received by changing the remaining battery capacity for the nodes along the path from the source sensor node to the gateway.

Our routing approach is based on a two-step strategy incorporating both link-based costs and end-to-end constraints. First we designate some 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. Upon obtaining these candidate paths we further check them to pick the least cost path that meets the end-to-end delay requirements and for which a feasible r -value for each node can be found. To account for the limitation of sensor's energy supply, we formulate a link cost function that captures the effect of required transmission power and available energy reserve at the nodes. In addition to

energy consideration, we also account for the transmission error due noise and packet drop. The end-to-end delay requirement, as we explain in the next subsection, is imposed on the whole path as a constraint. We define the following cost function for a link between nodes i and j :

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

where,

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

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 the more attractive it is for routing.
- CF_1 (Energy Stock) = $c_1 \times f(energy_j)$. This factor reflects the remaining battery lifetime (energy usage rate) favoring nodes with more energy. The more energy the node maintains, the better the stability of the formed routes.
- 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. $dist_{ij} / buffer_size_j$). Links with high error rate will increase the cost function, thus will be avoided.

2.4 Setting QoS Routes

In order to find a QoS path for sending real-time data to the gateway, first we obtain a set of energy-efficient paths. Then, we further check them to identify the one that can meet the end-to-end delay requirement by trying to find an r -value for each node on that path. Therefore, for each flow the necessary service rate at each node should be estimated. Let $T_{required}$ be the required end-to-end delay for the application. Thus, we should find a set of r -values for the nodes on the path so that $D(i) \leq T_{required}$. Using (2),

$$\frac{\sigma_i}{g(i)} + \sum_{m=1}^{M-1} \frac{P_{max}(i)}{g_i^m} + \sum_{m=1}^M \frac{P_{max}}{C} \leq T_{required}$$

In order to find g_i^m from the above equation, we assume that the service rate is same for all the nodes on the path of a particular flow, i.e. $g(i) = g_i^m$. Once g_i^m is calculated, then the link share can be calculated directly from (1),

$$\Phi_i^m = \frac{g_i^m * \sum_{j=1}^N \Phi_j^m}{C}$$

As mentioned in section 2.2, the service ratio for real-time data on a sensor node will be the summation of the link shares of all flows passing through that node, i.e. $r_m = \sum_{j=1}^N \Phi_j^m$, where N is the number of real-time data flows passing through node m and Φ_j^m is the link share for a real-time flow j on node m . The gateway calculates this r -value and sends it to the corresponding node.

By considering the above calculations, we propose the algorithm shown in Fig. 4, to find a least-cost path that meets the constraints for real-time data. The algorithm calculates the cost for each link, line 0 of Fig. 4, based on the cost function defined in subsection 2.3. Then, for each flow from imaging sensors, the least cost path to the gateway is found by running a K-shortest path algorithm in line 3. Here, the least cost path is taken among K alternatives by setting k to 1 in line 1. In lines 4-5, appropriate r -values are calculated for each real-

time data flow. In lines 6 and 7, the overall bandwidth spilt ratio of each node on the path is checked while considering all flows through the node. If that value is not between 0 and 1, alternative paths with bigger costs are tried by increasing the k value (line 2 and 3). As soon as a proper r -value is found, the loop exits (line 11). If there is no such r -value, the connection request of that node to the gateway is rejected (line 12-13).

In order to find the K least-cost paths, we modified an extended version of Dijkstra's algorithm given in [25]. Since the algorithm finds a set of paths with similar nodes and links, we modified the algorithm so that each time a new path is searched for a particular node; only node-disjoint paths are considered during the process. This ensures simplicity and helps in finding a proper r -value more easily since that node-disjoint path will not inherit the congestion of paths whose costs are less and for which the end-to-end delay requirements could not be met.

The performance of our routing approach is qualified through simulation. Section 4 describes the validation environment and analyzes the performance results. The next section presents an extension of our approach for handling delivery of delay-constrained sensor's data to mobile gateways.

3 Supporting Mobility of the Gateway

The routing mechanism, described in the previous section, provides timeliness for real-time data in a network of stationary sensor and gateway nodes. However, some applications might require the gateway node to change its location dynamically. For instance in a disaster management application the gateway can be a mobile robot or a moving emergency vehicle equipped with computing and communication devices. In setups where the gateway is mobile, ensuring uninterrupted data traffic and timely delivery of QoS data to the gateway becomes a challenge.

The main issue with a mobile gateway is the need for dynamic maintenance of the multi-hop routes in order to provide the desired service for the data as if the gateway is stationary. When the gateway goes out of the transmission range of a sensor node that is within one hop distance to the gateway, a mechanism to relay the data packets of that node to the gateway is needed. This process of maintaining routes will require frequent acknowledgment of the gateway's location to the network and setting up totally new routes based on the gateway's new locations when needed. This will cause significant overhead due to the increase in the volume of control packets and to the excessive number of state transitions of the inactive sensor nodes in turning their radio on and off. Such overhead will eventually increase the energy consumption of the sensor nodes and can significantly worsen the utilization of network resources.

Moreover, providing on-time delivery for real-time data in the case of a mobile gateway is much harder. Since real-time data is associated with end-to-end delay constraints, maintaining enough resources for real-time traffic while the gateway is on the move is crucial. Based on the new locations of the gateway, either new relay nodes with enough resources are found to keep the end-to-end delay within acceptable range or the routes are set up from scratch. This will again introduce extra traffic and hence increase the energy consumption of the sensor nodes.

Based on the above observations we define the problem as follows: Given the location and speed of the gateway and end-to-end delay constraints for real-time data, we are interested in extending our WFQ-based routing approach for efficient handling of gateway mobility such that the number of lost packet drops is

```

0 Calculate  $\text{cost}_{ij}, \forall i, j \in V$ 
1  $k \leftarrow 1$ 
2 Repeat
3   Find  $k^{\text{th}}$  least cost path for each node
4   for each flow from imaging sensor  $i$  do
5     begin
6       Compute  $\Phi_i$  for all the nodes from  $D(i) \leq T_{\text{required}}$ 
7       for each node  $m$  (on the path of flow  $i$ ) do
8          $r^m \leftarrow r^m + \Phi_i$ 
9         if ( $r^m$  is not in range  $[0,1]$ ) then
10          break;
11      end
12    $k \leftarrow k + 1$ 
13 until  $k > K$  or a proper  $r^m$  for each node is found
14 if no proper  $r^m$  is found then
15   Reject the connection

```

Fig. 4: Pseudo code for setting r -values

minimized and the hit ratio for real-time data is not diminished while the gateway is on the move. In this section we will first look at the design issues and tradeoffs in handling mobility and then describe our algorithm to extend the WFQ-based routing approach to handle gateway mobility.

3.1 Design issues and tradeoffs

A number of design and operational factors will affect the effectiveness and energy efficiency of the handling of the gateway mobility. Moreover, gateway mobility introduces some trade-offs that can impact the performance of the network in terms of various metrics. In this subsection, we highlight these issues and trade-offs and set the stage for presenting our algorithm for routing QoS traffic to a mobile gateway in the next subsection.

Rerouting overhead. Establishing a new network topology introduces two sets of overhead. The first set is concerned with control traffic. Once the gateway sets up new routes, it has to broadcast these routes in order to inform the sensor nodes about their role. Since a number of management strategies for sensor networks, e.g.[26][27], turn idle sensor to low-power sleep mode to save their energy, such broadcast requires either all sensor nodes to be continuously active or nodes in the sleep mode to switch to an active state on-time for hearing such route update. Keeping all sensors active all the time can deny the network an opportunity for energy conservation. On the other hand excessive state transition from active to sleep mode can also be a significant energy burden [28]. The network will suffer even more burden in case the routing protocol requires sensors to acknowledge the receipt of the routing messages and to engage with other neighboring nodes in establishing connections.

The second set of overhead is temporal in nature. Changes in the topology require the gateway and the sensors to put aside the data collection duties and get involved in exchanging control traffic. Such distraction can be damaging for QoS traffic since many packets are expected to either get dropped or arrive tardier than tolerable.

Energy-delay tradeoff. An intuitive approach for handling the gateway motion is to continually readjust the topology based on the new locations of the gateway. This rerouting needs to be done very frequently and new route information should be broadcasted to the network so that the loss of data packets is avoided and enough resources are reserved for real-time data. An alternative approach is for the sensor nodes to adjust their transmission range based on the future location of gateway as long as it is feasible to reach such location. Once the gateway becomes out of range, the current routes can be extended via designating new nodes as forwarders to the gateway.

A trade-off does exist between these two approaches. While ensuring gateway reachability to last hops through increasing the transmission power and designating ad hoc forwarders can quickly make the network operation inefficient, frequent changes to the route setup, as explained earlier, can raise the overhead of topology maintenance to an unacceptable level. Our algorithm, as described later, exploits this trade-off and further empirical analysis is conducted through simulation in section 4.

MAC-layer issues. One of the important factors that affect the network performance is the medium access control. In a stationary gateway setup, the underlying MAC protocol is typically designed to limit the interference among the transmission of sensors and minimize the energy consumed by the radio circuit. Medium arbitration techniques that reduce the collision among sensors have been proven effective in extending the life of the sensor networks [29]. Many of the proposed MAC protocols in the sensor networks literature aim at increasing the predictability of medium access pattern and decreasing the transmission power in order to limit the potential of signal interference. Therefore, the handling of gateway mobility has to consider the impact on medium access arbitration and avoid magnifying the level of signal interference.

While the gateway is on the move, frequent rerouting can limit the impact on the medium access or at least prevent subtle increase in signal interference. For a setup with a stationary gateway, an efficient network management methodology typically ensures that the routing and the MAC protocols are coherent and consistent with the performance goals. Thus, adjusting the topology of the sensor network in response to a

gateway repositioning will strive to maintain the efficiency of the medium access for the new set of routes. In addition, increasing the transmission power of the sensor nodes during the gateway motion can increase the probability of dropping the packets and collisions.

Although rerouting can maintain the efficiency at the MAC layer, as concluded in the above discussion, the frequency of rerouting should be minimized in order to limit the overhead to an acceptable level. Clever handling of the gateway motion at the MAC-layer is a part of our future research plan.

3.2 Routing to a Mobile Gateway

Given the energy and delay tradeoff associated with the frequency of rerouting, we pursue a mechanism that balances the interest in maintaining efficient network operation and avoiding excessive rerouting overhead. To limit the frequency of rerouting, our gateway motion handling mechanism tries to maintain continual packet delivery to the gateway through the adjustment of the transmission power of some sensors and via designating some forwarders to extend the current routes. A new network topology is formed when the end-to-end delay cannot be kept under an acceptable bound or the current routes become instable due to the failure of a node, e.g. the depletion of its energy supply.

Our approach relies on the gateway's full or partial knowledge of the navigation map and the travel schedule in order to set the network topology appropriately. The gateway is assumed to move in strides to reach intermediate positions. The strides are to be in straight line. This is similar to piece-wise linear approximation of curves in analytical geometry. The size of the strides can be determined based on the gateway travel speed. It should be noted that the gateway speed is assumed to be reasonable enough to handle. We will capture the effect of gateway speed on performance experimentally in section 4. Energy efficient routes are set at the initial gateway location so that the gateway keeps receiving the messages uninterruptedly till the next intermediate position, at which the network topology is reassessed. The reassessment concludes in one of three options: (1) The gateway can still be reachable by the last hops while traveling on the next stride, (2) forwarder nodes should be identified (3) rerouting is necessary. Such process is repeated till the gateway stops at a terminal location.

If the gateway still lays within the reachable range of the last hop nodes on the current routes, even at the end of the stride, the gateway will simply instruct these nodes to adjust the transmission power of their radio to cover the gateway's next move. While increasing the transmission power defies the energy efficiency achieved through the pursuit of multi-hop routing, the impact of such process on energy consumption will not be significant since the number of affected sensor nodes is limited. In addition, the gateway can consider the available energy of these nodes when making the decision of pursuing this option. For example, if the sensor node would lose much of its remaining battery life, the gateway exercises the second option of designating a forwarder. Of particular interest is the scenario when the gateway moves away from the last hop of a path towards the data sources. In such a scenario, the gateway can listen to the source sensors or intermediate relays other than the last hop.

If the gateway detects that it would go out of the transmission range of last hop nodes and cannot receive the data from other relay nodes, it explores the second option of finding sensor nodes to forward the packets. Ideal forwarder nodes should not be currently involved in relaying real-time traffic. If such free nodes can be augmented to a data path, there will be very limited additional queuing delay introduced to the path of real-time data. When an ideal forwarder could not be found, the gateway looks for a relay node that would have enough resources to handle the additional load. The delay, which such a potential forwarder would introduce, will be considered to re-evaluate the timeliness of real-time packets flowing over the path that this relay node is to join. If the delay increment that the real-time packets will incur is still acceptable, the node will be designated as a potential forwarder. Such delay increment is assessed by looking at the summation of current relay node's r -value and the r -value coming from the last hop node on the path. The summation should be within acceptable range, i.e. between 0 and 1.

In addition to the consideration of the real-time traffic that passes through a potential forwarder, the gateway always ensures that the picked forwarder can reach the gateway along its travel path until the next intermediate location and the forwarder has energy reserve enough for maintaining route stability. Among multiple potential forwarders, the gateway can employ multiple criteria in selecting the most appropriate

node. One obvious criterion is picking the node with the least load in order to minimize the introduced queuing delay. Other alternatives include the maximum remaining energy and the closest node to the next gateway intermediate location. Using the node with maximum energy reserve will extend the life of the node and enhance route stability, especially if the node serves on other data paths. The node closeness to the next intermediate gateway's position would increase the probability that the node will not need an additional forwarder on the travel stride. It should be noted that some combinations of these criteria can also be employed. When an appropriate forwarder cannot be identified, the gateway finally pursues the third option of setting up new routes.

A pictorial illustration of our approach is shown in Fig. 5. In Fig. 5a, when the gateway goes out of the transmission range of last hop nodes B and C, it searches for forwarders between its current location and the last hop nodes in order to relay B's and C's data. Nodes B_1 and C_1 are selected as forwarders and such information are sent to B and C for route update as seen in Fig. 5b. B_1 and C_1 act as relay nodes as long as the gateway is in their transmission range. While C_1 is an example for the unused forwarder, B_1 has to accommodate the additional traffic. New forwarders are found when gateway goes out of transmission range of B_1 and C_1 and such process continuous until the next reroute.

When the gateway moves closer to a particular intermediate node other than the last hop node on a path, the gateway can listen to the transmission of that node to the last hop node to receive the packets. This helps in decreasing the end-to-end delay for that packet by reducing the number of hops it travels. For example, in Fig. 5c, while the gateway moves closer to A, it can receive packets from A by listening to its transmission to D, instead of receiving them from D.

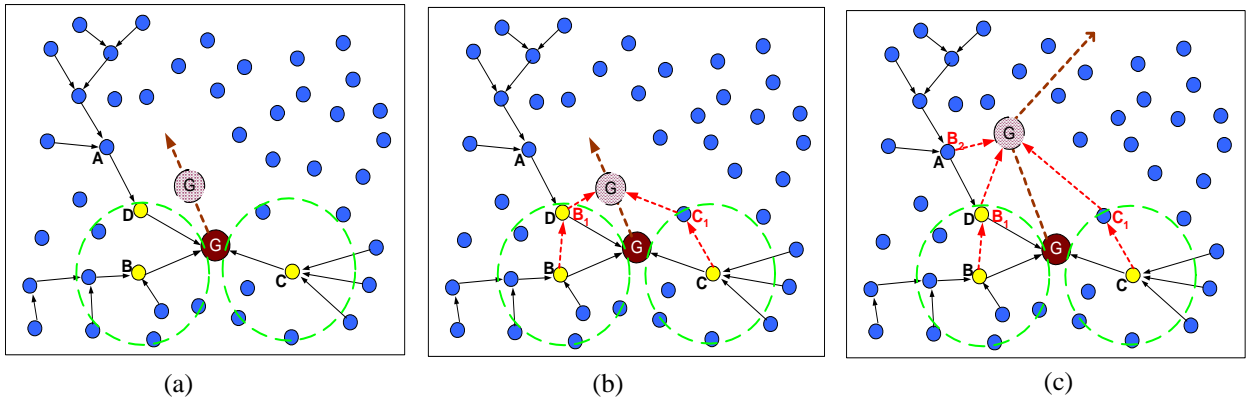


Fig. 5: (a) Gateway moves out of range, (b) Last hop nodes B and C finds new forwarders, i.e. B_1 and C_1 to relay their data when the gateway is out of their transmission range, (c) When gateway comes closer to a node on a path, it is used as the forwarder.

The proposed algorithm is shown in Fig. 6. At each stride, the algorithm checks each sensor in the list of nodes that are one hop away from the gateway to check whether the gateway is to get out of the range of that sensor or not (line 3-5). If the gateway is not out of range, it estimates the next intermediate position and sets the transmission range for the last hop node so that the gateway receives the messages uninterruptedly until reaching that intermediate position (line 10). If the last hop of a path is not reachable while a preceding node on that path becomes closer to the gateway while on the move, the gateway can listen to the node's transmission (line 7-8). If the gateway will be out of range, a new forwarder is found for that particular node in line 9. Line 14 reflects the criteria for selecting the most appropriate node among multiple potential forwarders. While the real-time traffic volume is the criterion used in the algorithm, as discussed earlier other criteria such energy-reserve and closeness to the new gateway location can be easily incorporated. If a forwarder cannot be found or even it is found but does not have enough resources to relay the real-time packets, a new reroute event is requested in line 23.

```

1 for (each movement (stride) of Gateway (G))
2   begin
   /*The list of nodes within one hop neighbor of G */
3   for ( $\forall j \in LastHopNodes$ )
4     begin
   /*TRange is the transmission range of a sensor, G(next) is the next location of G */
5     if ( $Dist(G(next), j) > TRange[j]$ )
6       GatewayOutOfRange  $\leftarrow 1$ ;
   /* When the gateway comes closer to a node on the path of j, it listens packets from that node */
7       if ( $Dist(G(next), j) > Dist(G(next), k \in Path(j))$ ) /* Path(j) is the path which has j as last hop node*/
8         UpdateRouteTable(k); /*Now listen k's transmission instead of j*/
9         FindForwarder(j); /*Find another node to relay the packets */
10      else TransmissionRange[j]  $\leftarrow Dist(G(next), j)$ ;
11    end
12  end

12 FindForwarder(j)
   /*Find a relay node between the new position of G and node j which is either free or has enough resources i.e.,
   an r-value between 0 and 1 can be found*/
13 Find Forwarders s.t. {  $\forall i \in Forwarders | Dist(j, i) < TRange[j] \ \&\& \ Dist(i, G_{new}) < TRange[j] \ \&\&$ 
   ( $r^i + r^j$ ) in range [0,1] && Remain_energy(j)  $< \delta$  }
14  $f \leftarrow \min_{(i \in Forwarders)} (r^i)$  /*Pick the least loaded node (other criteria are also possible)*/
15 Update RouteTable[j];
16 if (f not in LastHopNodes) /*Update the list of LastHopNodes accordingly*/
17   begin
18     Add f to LastHopNodes
19     UpdateRouteTable[f];
20   end
21 Remove j from LastHopNodes
22 if (Forwarders =  $\phi$ ) /*if no forwarder can be found, then trigger a reroute*/
23   RerouteReq  $\leftarrow 1$ ;

```

Fig. 6: Pseudo code for handling gateway motion under QoS traffic

4 Experimental Validation

The effectiveness of our energy-aware delay-constrained routing approach under stationary and mobile gateway architectures is validated through simulation. This section describes the underlying network operation, simulation parameters, performance metrics and experimental results.

4.1 Network Operation

We have adapted the network operational model of [26] for validating our approach. The gateway assumes responsibility for sensor organization based on missions that are assigned to the network. Mission-oriented organization of the sensor network enables the appropriate selection of only a subset of the sensors to be turned on and thus avoids wasting the energy of sensors that do not have to be involved. Thus the gateway will control the configuration of the data processing circuitry of each sensor.

The sensor nodes can be in one of four main states: sensing only, relaying only, sensing-relaying, and inactive. In the *sensing state*, the node sensing circuitry is on and it sends data to the gateway at a constant rate. In the *relaying state*, the node does not probe the environment but its communications circuitry is on to relay the data from other active nodes. When a node is both sensing the target and relaying messages from

other nodes, it is considered in the *sensing-relaying state*. Otherwise, the node is considered *inactive* and can switch to a low power sleep mode. The decision for determining the node's state is done at the gateway based on the current sensor organization, node battery levels, and desired network performance measures.

The typical operation of the network consists of two alternating cycles: data cycle and routing cycle. During the data cycle, the nodes, which are sensing the environment sends their data to the gateway. In the routing cycle, the state of each node in the network is determined by the gateway and the nodes are then informed about their newly assigned states and how to route the data. Rerouting is performed after a specified number of data cycles, which we refer as reroute period in this section. Rerouting is triggered by an application-related event requiring different set of sensors to probe the environment including the activation of new sensors based on the gateway movement or the depletion of the battery of an active node. A time division multiple access (TDMA) based MAC protocol is used. The gateway manages slot assignment based on the network topology. The gateway informs each node about slots in which it should listen to other nodes' transmission and slots, which the node can use for its own transmission.

4.2 Environment Setup

In the experiments, the network consists of 100 randomly placed sensor nodes deployed in a 500×500 meter square area. The gateway initial position is determined randomly within the region boundaries. A free space propagation channel model is assumed [30] with the capacity set to 2Mbps. Packet lengths are 10 Kbit for data packets and 2 Kbit for routing packets. Each node is assumed to have an initial energy of 5 joules. A node is considered non-functional if its energy level reaches zero. The maximum transmission range for a sensor node is assumed to be 50 meters [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. [16]. The sources generating data are assumed to be leaky bucket constrained with the maximum burst parameter σ of 10 packets. 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.

We assume that the network is tasked with a target-tracking mission in the experiment and the gateway is moving based on the Random Waypoint Model [32] with a maximum speed of 5m/s unless otherwise specified. 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 the other nodes is usually turned off but can be turned on according to the target or gateway movement. We also assume that each sensor node is capable of taking the image of 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 capabilities. 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. Packets, generated by imaging sensors, are labeled as of real-time type and treated differently at the relaying 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 real-time data is taken to be 0.08 sec [33]. 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 m/s to 6 m/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 area triggering the generation of a new target.

4.3 Performance Metrics

Given our interest in providing on-time delivery of real-time data and minimizing the impact of gateway mobility on the timeliness of real-time packets, we used the following metrics to capture the performance of our approach:

- *Average delay per packet*: Defined as the average time a packet takes from a sensor node to the gateway. The applications that deal with real-time data is delay sensitive, so this metric is important in our case.
- *Deadline Miss Ratio*: This is one of the most important metrics in real-time applications, which indicates the number of packets that could not meet the specified delivery deadline.
- *Average energy per packet*: This metric represents the average energy consumed in the network for transmitting and relaying a data packet until the gateway successfully receives it.

4.4 Performance Results

In this section we present some performance results obtained through simulation. As a baseline approach, we have used the same cost function with same routing algorithm without doing any service differentiation. That is, we have only one queue in each sensor node for all kinds of packets. Therefore, no bandwidth allocation is done through adjusting of r -values.

Delay and Timeliness: When comparing the average delay per real-time packet achieved through our WFQ based model with the average delay per packet obtained in the single queue model, we have observed that the service differentiation employed using our WFQ-based approach achieves substantially better average delay for both mobile and stationary gateway models (See Fig. 7). For all approaches, there is an increase in the average delay per real-time packet when the real-time data rate is boosted. This is due to the increase in the queuing delay that the real-time packets incur. However, such increment in the average delay per real-time packet is very small when using the WFQ-based approach. Although the average delay per real-time packet from distant nodes increases, nodes close to the gateway, which have less average delay, also generate more real-time packets, causing the overall average delay for real-time packets to stabilize. In the case of a mobile gateway, the results in Fig. 7 have shown that our approach for handling mobility slightly increases the average delay per real-time packet since the routes are adjusted by adding new hops introducing extra delay for each packet.

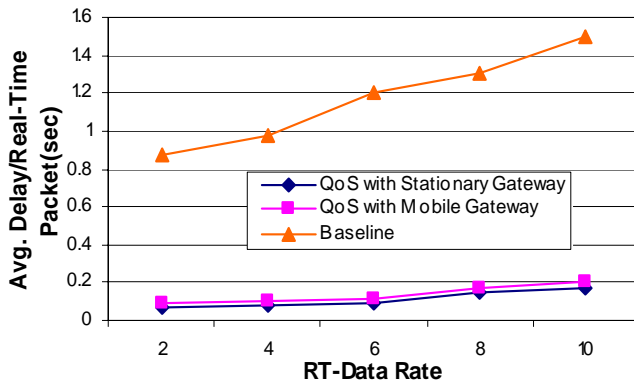


Fig. 7: Effect of the data generation rate on average delay per real-time packet for three approaches.

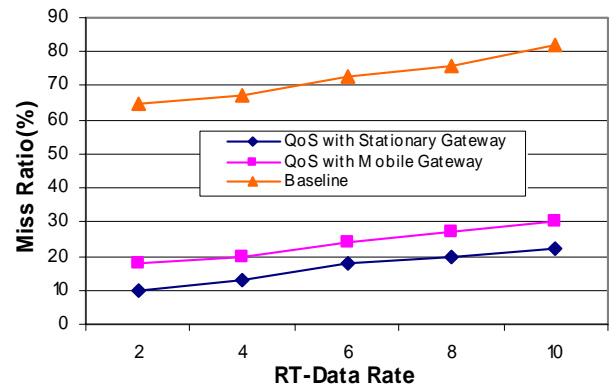


Fig. 8: Miss ratios of real-time packets versus data generation rate.

Fig 8 depicts the results for the number of real-time packets that missed the deadline for all approaches. While our approach achieves almost 90% on-time delivery with stationary gateway and 80% with mobile gateway when there is no congestion in the network, applying the baseline approach causes most of the packets to miss their deadlines.

It is interesting to note that the amount of growth both in average delay and miss ratio due to gateway mobility is stable under different loads in comparison to a stationary gateway, hinting to the predictability of our approach. In addition, the effect of gateway mobility becomes marginal under heavy load. For instance in Fig. 8, when the RT-data rate is 2, the miss ratio for the mobile gateway case almost doubles that for a stationary gateway. However, the significance of such increase in miss ratio is much less when the RT-data rate is 10, going down from 100% to about 30% of the miss ratio of a stationary gateway setup.

Energy Consumption: Fig. 9 depicts the impact of the generation rate of real-time data on energy usage, measured in terms of average energy per packet. It can be observed that for moderate real-time data rates, routes set using our algorithm lead to the consumption of almost the same amount of energy compared to the baseline approach. As the real-time data grows, the gap between the QoS routing and baseline curves widens. This is due to not being able to consistently use the least cost path while meeting en-to-end delay constraints. On the other hand, the average energy consumed per packet in the case of a mobile gateway is higher than that for a stationary gateway. This is because the transmission power of last hop sensors gets increased during the gateway motion in order to sustain the reachability to the gateway at further distance. Moreover, any extra hop added when the gateway is out of range will introduce new transmission and reception energy for a packet. Again the increment in energy usage to handle gateway mobility is very stable and becomes relatively marginal at high data rates.

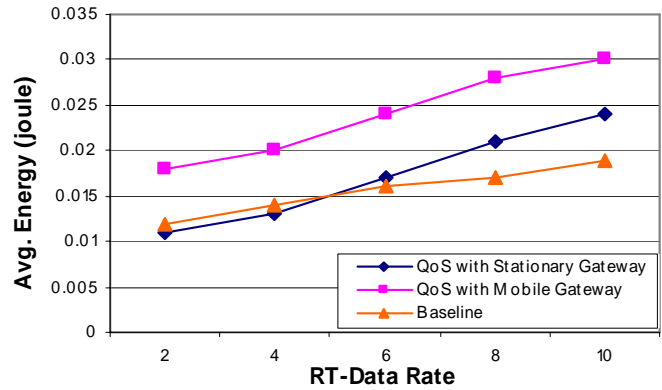


Fig. 9: Average energy per packet under different real-time data rates.

Mobility and rerouting effects: When the gateway is mobile, frequent adjustment of the network topology will introduce significant overhead in terms of energy consumption as explained in section 3.1. In our simulated network setup such overhead is due to increased number of state changes from active to sleep mode employed by the MAC layer protocol in order to save energy. Recent research has shown that such excessive state transitions between active and sleep modes can introduce significant amount of energy consumption [28]. Therefore, we conducted experiments in order to observe the effect of rerouting frequency on both energy consumption and packet delay.

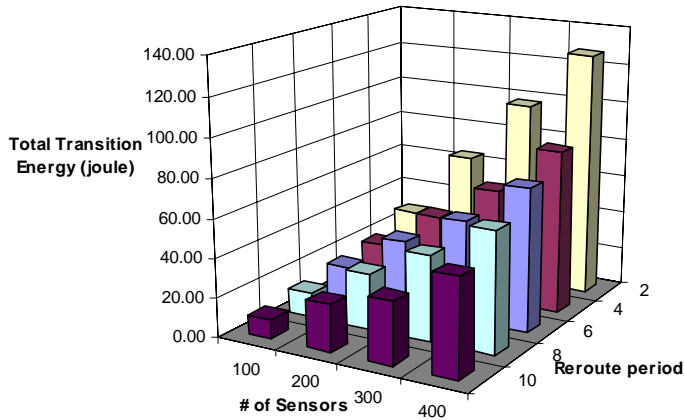


Fig. 10: Total overhead energy with respect to different number of sensors and varying reroute periods.

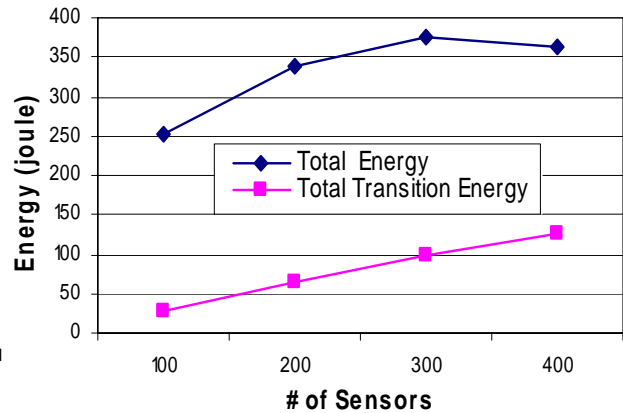


Fig. 11: Total energy and transition energy for increased number of sensors.

In order to investigate the effect of reroute frequency, we have varied the reroute period i.e. the number of data phases to pass before the next reroute. We have also studied how such effect relates to the number of sensors. The results, depicted in Fig. 10, have shown that the total transition energy due to state changes increases significantly with a smaller reroute period. Such effect is very much expected. Moreover, we observe from Fig. 11 that when the number of sensors increases significantly, the overhead of rerouting becomes dominant in total energy consumption.

The effect of reroute frequency on the energy consumption for a mobile gateway based setup is captured in Fig. 12. The observed results indicate that the average energy per packet grows, in comparison to the case of a stationary gateway, when the duration between successive rerouting is increased. Such effect is due to the use of the longer paths as explained when discussing Fig. 9.

We have also considered the effect of the different reroute periods on the delay and timeliness metrics. As indicated by Fig. 13, the average delay per real-time packet grows when the frequency of rerouting get lower. Such observation is expected since a longer reroute period means more forwarders are engaged increasing the number of hops for the routes. Such increase in hops adds extra delay for the packets, especially when the forwarder is not dedicated to one path. In case of a stationary gateway, the average delay per real-time packet remains stable. Looking at the miss ratio (Fig 14), it can be observed that the number of real-time packets missing their deadline increases in the case of a mobile gateway when the reroute period increases. A stationary gateway almost keeps the same performance. This is due to same reason explained above.

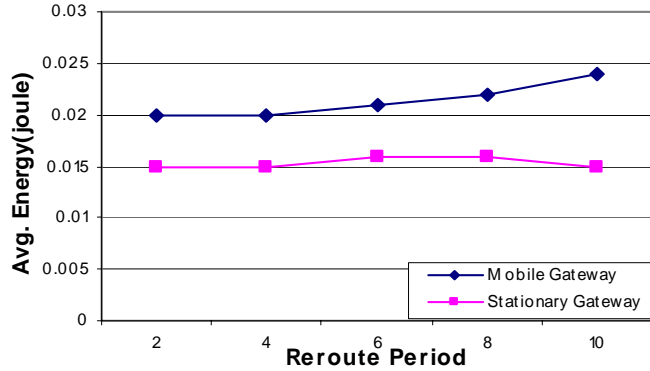


Fig. 12: Average energy per packet under different reroute periods.

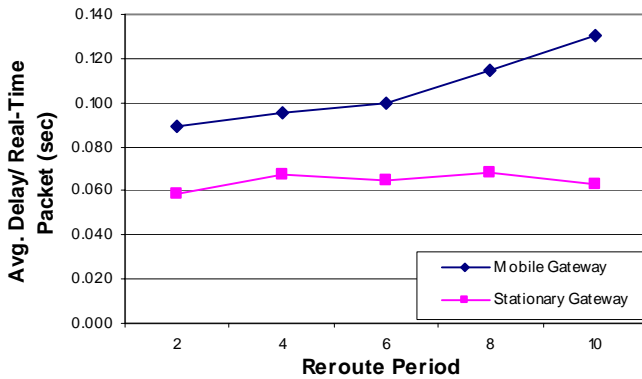


Fig. 13: Average delay per real-time packet for different reroute periods.

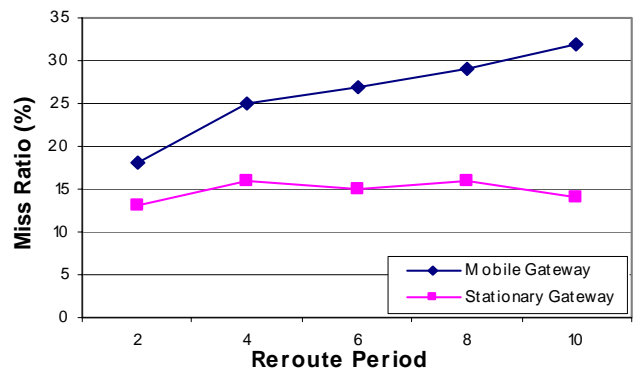


Fig. 14: Effect of the reroute period on the miss ratio for real-time packets.

Yet another important factor that affects the performance of our approach in the mobile gateway architecture is the speed of the gateway. We have varied the gateway speed in order to see the effect of mobility on the miss ratio and delay. The experiment results have shown that with the increased gateway speed, it becomes difficult to meet the deadlines of real-time packets as seen in Fig. 15. Increased gateway speed will cause the gateway to go out of range quickly, resulting new hops to be created immediately. These new hops will introduce extra delay and cause some of the real-time packets to miss their deadlines. The best performance in this case is when the gateway speed is 0, which means a stationary gateway.

Based on the observations from figures discussed in this subsection, we can conclude that a routing

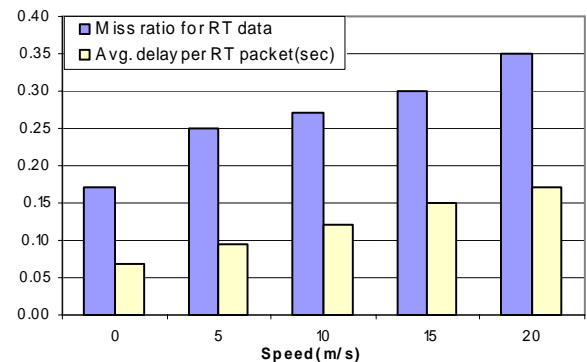


Fig. 15: Miss ratio for real-time packets and average delay per real-time packet grows with increased gateway speed.

protocol that handles the mobility of the gateway should consider minimizing the number of reroutes i.e., increase the number of reroute period without negatively affecting the delivery of real-time packets on time. Moreover, fast moving gateways can negatively affect the network efficiency and even make it impossible to get on-time data delivery. Our approach for providing energy-aware delay-constrained routing with a mobile gateway can be adjusted through the tuning the reroute period in order to give the best performance in terms of energy usage and timeliness. We envision that such tuning has to be based on the application of the sensor network.

5 Conclusion

In this paper, we have presented a new energy-aware delay-constrained routing approach for sensor networks. The proposed approach finds energy-efficient paths for real-time data that is subject to end-to-end delay requirements. WFQ, which allows service sharing for real-time and non-real-time flows, is employed in order to support both best effort and delay-constrained traffic. A service ratio r is derived for calculating the amount of bandwidth to be dedicated to the real-time and non-real-time traffic on a particular outgoing link. Since WFQ provides per flow upper bounds on end-to-end delay, each sensor generating real-time data is considered as a different flow. However, rather than employing a different queue and link share for each real-time flow, a single queue is used to accommodate all real-time data from different flows. The real-time service ratio r for that queue is calculated as the summation of the link shares of real-time flows passing through that node. Such service rate estimation mechanism is used along with a leaky bucket constrained packet generation model so that end-to-end delay bounds are met. The approach is further extended to support a mobile gateway (sink) considering dynamic update of routes when the gateway is on the move. Once the gateway moves out of range, we find new sensor nodes within the communication range of the gateway to relay the packets and the routes are updated by adding those new hops. If the gateway moves close to the source nodes, it overhears on-going traffic to limit the end-to-end delay and save unnecessary relaying. The updated routes are used until the next reroute where efficient routes are set based on the new location of the gateway.

The effectiveness of the proposed routing algorithm is validated through simulation. Simulation results have shown that our approach for the stationary gateway model consistently performs well in terms of miss ratio and average delay, compared to a baseline mechanism in which same link cost function and routing algorithm are used without performing any service differentiation. Our approach provides at least 80% on-time delivery of real-time data even when the network is congested through adjusting the real-time data service rate. Moreover, the approach enhances timeliness without major negative impact on the consumed energy. The simulation results also have demonstrated that the mobility of the gateway can be handled efficiently at the expense of a small increase in the average energy consumed per packet and miss ratio for real-time packets. Our experiments have indicated that for the optimal performance in terms of timeliness and energy consumption the reroute frequency should be carefully selected.

References

- [1] I. F. Akyildiz et al., "Wireless sensor networks: a survey", *Computer Networks*, Vol. 38, pp. 393-422, 2002.
- [2] D. Estrin, et al., "Next Century Challenges: Scalable Coordination in Sensor Networks," in the *Proceedings of International Conference on Mobile Computing and Networks (MobiCOM '99)*, Seattle, WA, August 1999.
- [3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," in *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 *Proceedings of International 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] W. Rabiner Heinzelman et al., "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," in the *Proceedings of Hawaii International Conference on System Sciences (HICSS '00)*, January 2000.

- [8] W. Heinzelman et al., "Adaptive protocols for information dissemination in wireless sensor networks," in the *Proceedings of 5th Annual ACM/IEEE MobiCom'99*, Seattle, WA, August 1999.
- [9] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the *Proceedings of the 6th Annual ACM/IEEE MobiCom'00*, Boston, MA, August 2000.
- [10] R. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", in the *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, March 2002.
- [11] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in the *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA)*, Atlanta, GA, October 2002.
- [12] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," in *Elsevier Journal of Ad Hoc Networks* (to appear).
- [13] A. K. Parekh and G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case", *IEEE Trans. on Networking*, vol. 1, no. 3, pp. 344-357, June 1993.
- [14] A. Demers et al., "Analysis and simulation of a fair queuing algorithm" in *Journal of Internetworking Research and Experience*, pp 3-26, October 1990.
- [15] A. K. Parekh and G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks Services Networks: The Multiple Node Case", in the *Proceedings of IEEE INFOCOM 1993*.
- [16] "Data sheet for the Acoustic Ballistic Module", SenTech Inc., <http://www.sentech acoustic.com/>
- [17] Z. Wang and J. Crowcraft, "QoS-based Routing for Supporting Resource Reservation," *IEEE Journal on Selected Area of Communications*, Sept 1996.
- [18] Q. Ma and P. Steenkiste, "Quality-of-Service routing with Performance Guarantees," in the *Proceedings of the 4th International IFIP Workshop on Quality of Service*, May 1997.
- [19] 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.
- [20] C. Zhu and M. S. Corson, "QoS routing for mobile ad hoc networks," *In the Proceedings of IEEE INFOCOM*, 2002.
- [21] T. He et al., "SPEED: A stateless protocol for real-time communication in sensor networks," in the *Proceedings of International Conference on Distributed Computing Systems*, Providence, RI, May 2003.
- [22] H. Luo et al., "TTDD: Two-tier Data Dissemination in Large-scale Wireless Sensor Networks," *ACM/Kluwer Mobile Networks and Applications (MONET), Special Issue on ACM MOBICOM (2002)*.
- [23] G. Feng et al., "Performance Evaluation of Delay-Constrained Least-Cost Routing Algorithms Based on Linear and Nonlinear Lagrange Relaxation," In the *Proceedings of the IEEE ICC'2002*, New York, April 2002.
- [24] 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.
- [25] Q.V.M. Ernesto et al., "The K shortest paths problem," *Research Report, CISUC*, June 1998.
- [26] M. Younis et al., "Energy-Aware Routing in Cluster-Based Sensor Networks", in the *Proceedings of IEEE/ACM MASCOTS2002*, Fort Worth, Texas, October 2002.
- [27] H. Yang and B. Sikdar, "A Protocol for Tracking Mobile Targets using Sensor Networks," in the *Proceedings of the 1st IEEE Workshop on Sensor Networks Protocols and Applications (SNPA 2003)*, Anchorage, AK, June 2003.
- [28] E. Shih et al., "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in the *7th Annual International Conference on Mobile Computing and Networking*, 2001.
- [29] P. Havinga and G. Smit, "Energy-efficient TDMA medium access control protocol scheduling," in the *Proceedings of the Asian International Mobile Computing Conference (AMOC 2000)*, November 2000.
- [30] J.B. Andersen et al., "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, Vol. 33, No. 1, January 1995.
- [31] A. Chandrakasan, et al., "Power Aware Wireless Microsensor Systems", Keynote Paper ESSCIRC, Italy, 2002.
- [32] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in T. Imelinsky and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.
- [33] 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.