Relocation of Gateway for Enhanced Timeliness 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 have witnessed an increasing interest in applications of wireless sensor networks that demand certain quality of service (QoS) guarantees. To respond to the needs of such emerging applications, new routing protocols have been proposed for providing energy-efficient real-time relaying of data. All of these protocols assumed a stationary sink node and did not consider any possible movement of the sink node for performance purposes. In this paper, we propose possible relocation of sink (gateway) for improving the timeliness of real-time packets when the network is congested. Our approach searches for a location close to the most loaded last hop node or one of its siblings. The gateway is then relocated to the new location so that the load of that node is alleviated and the real-time traffic can be split. As long as the gateway stays within the transmission range of all last hop nodes, it can be moved to that location without affecting the current route setup. However, if such movement gets the gateway out of the transmission range of one of those hops, routes are adjusted by introducing new forwarders picked from the nodes that are not involved in data traffic. Simulation results demonstrate the effectiveness of the proposed approach for popular performance metrics.

1 Introduction

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 [1][2][3][4][5][6]. Sensors in such applications are deployed, usually unattended, in sheer numbers for data gathering and processing. Sensor nodes are usually miniaturized and constrained in energy supply, bandwidth, on-board memory and processing capacity. The constraints in energy supply have 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 sources to a particular sink (gateway). 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. 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 proposed. The proposed routing protocols not only ensure soft real-time delay guarantees through the duration of a connection but also provide the use of most energy efficient path [13][14]. While such protocols achieve soft end-to-end delay guarantees, their service can diminish with the increasing volume of real-time data. In such cases, most of the packets can start to miss their specified deadlines. In order to enhance timeliness in such situations, one of the solutions is to explore gateway's ability to move to a location where volume of real-time data is high. An example of this scenario is when the gateway is a laptop computer or other portable devices on the backpack of a rescue crew who is not expected to travel long distances. Such relocation in those circumstances can balance the traffic load among multiple nodes and hence increase the hit ratio of real-time packets.

In this paper, we present a novel mechanism for relocation of the gateway under QoS traffic for enhanced on-time delivery in sensor networks. End-to-end delay bound for real-time data is achieved through the use of a Weighted Fair Queuing (WFQ) based packet scheduling technique in each sensor node [15][16]. 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 [17]. When average hit ratio for real-time packets starts to decrease, our approach considers moving the gateway to a better position in order to maintain the same or even better level of timeliness. To determine the new gateway's position, we consider locations on or close to heavily loaded last hop node and try to split the incoming traffic passing through that node without extending the delay experienced by real-time packets over other routes. As long as the gateway remains within the transmission range of all the last hop nodes, we maintain the same routes that were set initially. If it is expected that the new location will put the gateway out of the transmission range of some of the last hop nodes in the current routes, new forwarder nodes that are not involved in any routing activity are selected. Such unused nodes will introduce very little queuing delay, which is desirable for timeliness of all real-time packets using those nodes as relays. If such nodes cannot be found, relocation is either not considered or a new network topology is set at the new location depending on the overhead rerouting will introduce relative to the gain in timeliness.

In the balance of this section we describe the sensor network architecture that we consider and summarize the related work. In section 2, we first summarize the underlying WFQ based approach for achieving on-time delivery of real-time data and then describe our approach for handling issues related to the relocation of the gateway. Section 3 discusses performance evaluation and simulation results. Finally we conclude the paper with a summary in section 4.

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.

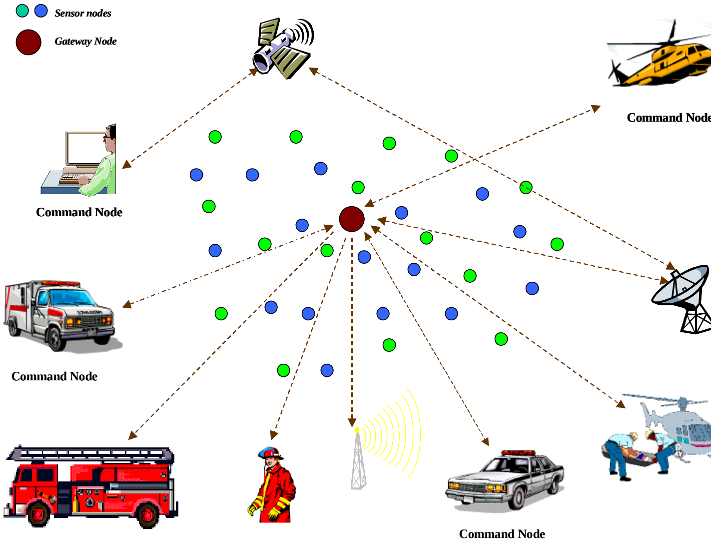


Fig. 1: Three-tier sensor network architecture

While sensor nodes are stationary, we are considering a limited mobility model for the gateway. 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, processing and radio circuits can be powered on and off. In addition, the radio's transmission power is assumed to be programmable 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. [18]. 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.

1.2 Related Work

While contemporary best-effort routing approaches address-unconstrained traffic, 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.

Another QoS routing protocol for sensor networks that provides soft real-time end-to-end guarantees is SPEED [13]. 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.

The approach of [19] finds a least cost and energy efficient path that meets end-to-end delay requirements. The link cost used is a function that captures the nodes' energy reserve, transmission energy, error rate and other communication parameters. In order to support both best effort and real-time traffic at the same time, a class-based queuing model is employed. The queuing model allows service sharing for real-time and non-real-time traffic. The bandwidth ratio r, is defined as 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. As a consequence, the throughput for normal data does not diminish by properly adjusting such "r" value. The protocol finds a list of least cost paths by using an extended version of Dijkstra's algorithm and picks a path from that list which meets the end-to-end delay requirement.

All of the described protocols assumed a stationary gateway and did not consider relocation of the gateway node for enhanced performance. The only work to consider relocation of gateway node in wireless sensor networks is reported in [20]. This work considers relocation of the gateway in order to reduce average energy per packet and hence increase the lifetime of the network by checking the traffic density of the nodes that are one-hop away from the gateway and their distance from the gateway. Once the total transmission power for such nodes is guaranteed to reduce more than a certain threshold and the overhead of moving the gateway is justified, the gateway is relocated to new location. While the presented approach aims at optimizing the energy consumption and the lifetime of the network, it works only for unconstrained data traffic and does not fit to the needs of real-time traffic. Our approach is an extension to that work which works under the involvement of real-time traffic as well. It periodically checks for possible performance degradation and triggers relocation for the gateway node when needed. To the best of our knowledge, our work is the first to introduce relocation of gateway for enhanced end-to-end delay and miss ratio when real-time traffic is involved.

2 Relocation of the Gateway for Enhanced Performance

Since the gateway is the final destination for the data packets, its location can be very influential to the networking performance. In a multi-hop wireless networks, the throughput, average delay and energy consumed in packet routing depends on the positions of the sources of the data and the destination. The location of the gateway becomes even more influential when real-time traffic is involved. For instance, in some circumstances most of the real-time path establishment requests can be denied or the hit ratio for real-time packets can decrease significantly due to congested routes through the gateway. Traffic congestion can be caused by the increasing the number of real-time data packets coming from nodes close to a recently event. In such circumstances, it may not be feasible to meet the requirements for real-time data delivery. This scenario can be unacceptable for applications like disaster management and target tracking, since failing to provide real-time traffic within the required time period may negatively impact the application. If the gateway has a capability to move, it can be beneficial to relocate the gateway close to the occurring event in order to spread the traffic on more hops and increase the feasibility of meeting the real-time delivery requirements. Even a limited motion of gateway can make big difference in terms decreasing end-to-end delay and miss ratio for real-time packets.

Based on the observations above, we define the problem as follows: Given the location of the gateway and established routes for on time delivery of real-time data, we are interested in relocating the gateway to a new position such that better end-to-end delay and hit ratio are achieved without affecting the energy usage of the network. In this section, we first describe the underlying routing mechanism for providing on-time delivery of real-time data and then explain our approach of when, where and how to relocate the gateway for enhanced performance.

2.1 Real-time Routing Mechanism

Our routing approach for supporting real-time traffic is based on the model defined in [14]. Each node employs a packet scheduling discipline that approximates Generalized Processor Sharing (GPS) [15]. 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. The packetized version of GPS is called Weighted Fair Queuing (WFQ). One interesting property of WFQ is that when combined with leaky bucket constrained sources, it can provide upper end-to-end delay bounds for each flow [16][17]. Assuming flow *i* is constrained by a leaky bucket with parameters (σ_i, ρ_i), the maximum end-to-end delay (transmission + queuing delay) for a packet of flow *i* under WFQ, given in [17], is:

$$D(i) \le \frac{\sigma_i}{g(i)} + \sum_{m=1}^{M-1} \frac{P_{\max}(i)}{g_i^m} + \sum_{m=1}^{M} \frac{P_{\max}}{C} \qquad [1]$$

$$g_i^m = \frac{\Phi_i^m}{\sum_{j=1}^{n} \Phi_j^m} C \qquad [2]$$

where,

- C is the link bandwidth
- $P_{max}(i)$ is maximum packet size for flow i
- P_{max} is maximum packet size in the network
- g_i^m is the service rate on node m for flow i
- $g(i)$ is the minimum of all service rates for flow i
- M is the number of nodes on path of flow i
- Φ_i^m is the link share on node m for flow i

Since WFQ is flow based, the approach of [14] uses an approximation of WFQ by considering each imaging sensor node as a source of different real-time flow with only one real-time queue to accommodate the real-time data coming from these multiple flows (Fig. 2). This model 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. The service ratio "*r*" is the bandwidth ratio 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. This 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. In this case, r_m for the real-time queue on a node is the summation of link shares $(\Phi_1^m, \Phi_2^m, ..., \Phi_n^m)$ of all real-time flows passing through that node as shown in Fig. 2. Each link share is calculated through the formula (2).

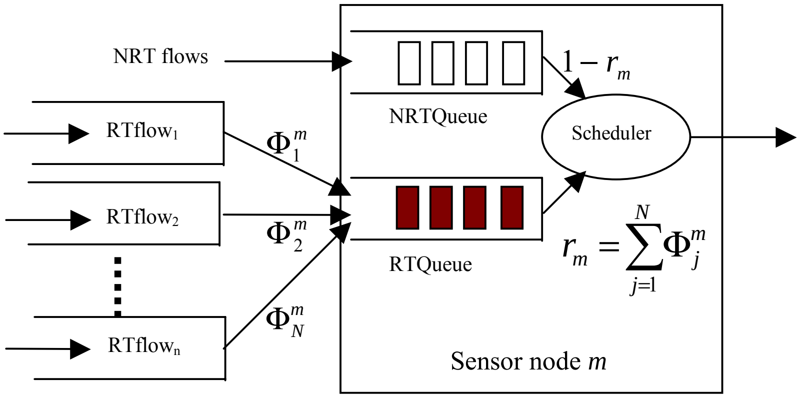


Fig. 2: Queuing model on a sensor node

In order to setup real-time routes to the gateway, the approach first obtains a set of energy-efficient paths for each real-time source. Then, these paths are further checked 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. Once an r_m between 0 and 1 can be found for each node, the real-time connection is established for that path.

2.2 Relocation Approach

There are three issues to consider in gateway relocation. These are when to relocate the gateway, where to put it and how to handle its motion without disrupting data traffic. In this subsection, we will discuss these issues and then state our algorithm to handle such issues.

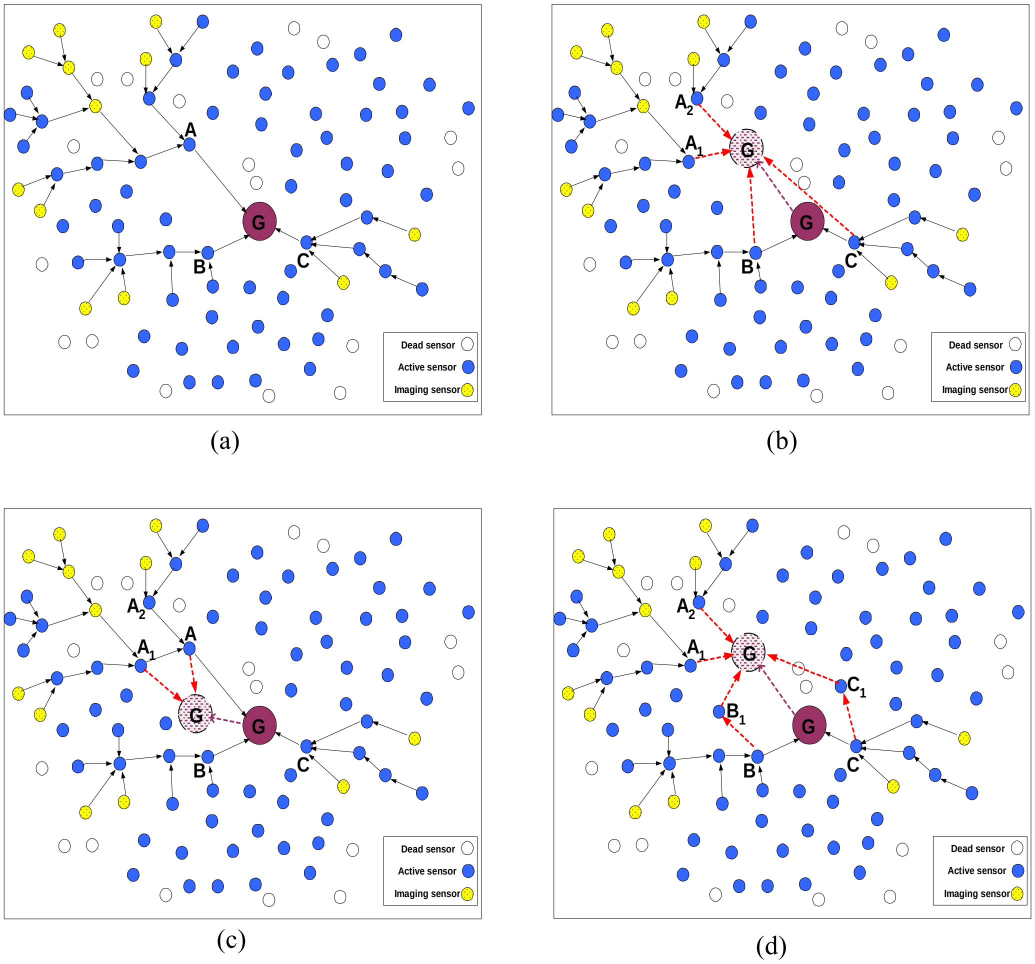
When to move the gateway: First of all, the time for such relocation should be decided based on some performance observations such as miss ratio. The miss ratio is indirectly related to finding proper r-values in our underlying approach. Note that, our approach for real-time routing basically tries to find an r-value by considering every path from the source of real-time traffic to the gateway. In cases where a proper r-value between 0 and 1 cannot be found, the connection is simply rejected and the path is not established. Moreover, even proper r-values are found, the miss ratio can start to decrease due to involvement of new events that cause generation of additional real-time packets and hence more congestion. Therefore, the trigger for relocation will be the unacceptable increase in the miss ratio of real-time packets. In our approach, the gateway will set a threshold for the maximum level of miss ratio by maintaining such statistics periodically and consider relocating the gateway to a better location when such threshold is reached. Note that, the gateway can be relocated more than once whenever necessary during the data traffic cycles.

Where to move the gateway: After deciding that the gateway is to be relocated, our approach will consider searching a new location for the gateway. The main aim here is to move the gateway towards the loaded nodes in terms of real-time traffic so that the end-to-end delay can be decreased. Therefore, our approach first searches the last hop nodes i.e. the nodes that are directly transmit to the gateway in order to designate the hop with the biggest r-value and consider relocating the gateway at the position of that hop. This will helpful in twofold: (1) It will help in decreasing the average end-to-end delay since the number of hops for data packets to travel will be decreased. (2) It will help in admitting more real-time flows since the load is alleviated by splitting the traffic.

However, finding such new location for the gateway does not necessarily mean that the gateway will be relocated. Before moving the gateway, an analysis should be performed in order to assess the potential overhead that will be introduced when gateway is relocated at the new location. Such overhead will be due to any necessary route adjustment when the gateway goes out of the transmission range of some of the other last hop nodes. If this is the case, new relay nodes should be found to maintain uninterrupted data delivery and consequently some routes will need to be updated accordingly. While involvement of such new relay nodes will introduce additional delay for real-time packets and increase energy consumption, forming new network topology based on the new location of the gateway can be undesirable due to its overhead in terms of control traffic and energy [21][22].

Therefore, when a new location is found, our approach first checks whether at the new location the gateway will go out of range or not. This is also used for breaking the ties when multiple alternative nodes with the same *r*-value are found. In such circumstances, if the other last hop nodes can still reach the gateway by increasing their transmission power, relocation can be performed safely. A pictorial illustration of this situation is depicted in Fig. 3a and 3b. If we assume that there is an occurring event on north-west of node A, many of the imaging sensors that are sources of real-time traffic will be turned on in that region, increasing the traffic density flowing through node A (Fig. 3a). In this case, most of the paths passing through node A will be rejected since the limited service rate on node A increases the end-to-end queuing delay. However, the load on node A can be alleviated by moving the gateway to the location of A. If the gateway is still reachable by the nodes B and C, they just increase their transmission ranges for uninterrupted data delivery as seen in Fig 3b.

If the new location requires some topology changes then our approach considers alternative positions that might be feasible for the gateway to be positioned at without causing any effect on the network topology. Such alternatives include the location of one of the siblings of the node with the biggest *r*-value or some other location that will be closer to that sibling and enable data relaying without affecting the



(a)

(b)

(c)

(d)

Fig. 3: (a) Initial routes (b) Gateway is relocated to the location of A if it is not out of range of C and B. (c) If that is not possible, gateway is relocated to a location close to B. (d) Otherwise unused forwarders B_1 and C_1 are found for relaying of traffic from B and C

latency of other real-time packets. Here the selection of the sibling node is decided based on its *r*-value since targeting the sibling with the biggest *r*-value can help best in splitting the real-time traffic. An example of this situation is shown in Fig. 3c. In this case, A_1, which is the sibling of A with the biggest *r*-value, is picked and gateway is relocated closer to that node so that the load of A is split between A and A_1. Such relocation does not require any route updates for the current topology. Note that, here the new position of the gateway should be within the communication range of A_1 and the other nodes A, B and C so that the traffic passing through A_1 can be directed to the gateway to alleviate the load of A. At the same time nodes A, B and C can still reach the gateway without any need to adjust the routes.

In cases where a new location that will not cause the gateway to be out of range cannot be found, our approach strives to minimize the overhead of adjusting the routes. Therefore, new forwarder nodes that are currently not involved in real-time traffic relaying are picked for each of the last hop node in order to provide uninterrupted delivery of real-time packets. Since such nodes will not introduce extra queuing delay for real-time data, this will not affect the end-to-end delay of real-time data using that path. An example is shown in Fig. 3d. In this case, nodes B_1 and C_1 that are not part of any existing routes are designated as forwarders for nodes B and C respectively.

Failure to find such forwarders will result in keeping the gateway in its current position since the possible gain in relocation can be degraded due to overhead explained above. It should be noted that in some network architectures the gain achieved by the gateway relocation is valued more than the rerouting overhead. In such architectures it would make sense not to give up on gateway relocation if the network topology cannot be maintained. Since this scenario depends highly on the routing protocol and network management strategy, we have decided to flag such case for further architecture specific analysis or to simply seize the gateway movement.

How to move the gateway: The data transmission during the movement of the gateway is also a concern. It will be unacceptable to increase packet losses on some data paths. Our gateway motion handling mechanism tries to maintain continual packet delivery to the gateway through the adjustment of the transmission power of some sensors or via designating some forwarders to extend the current routes.

Once the new location is determined, the gateway explores two options based on the information of whether it will be out of range at the new location or not: If the gateway can still be reachable by the last hop nodes when relocated, it will simply instruct these nodes to adjust the transmission power of their radio to cover the gateway's new location and starts moving there. 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 at the new location, it explores the option of employing sensor nodes to forward the packets. Ideal forwarder nodes should not be currently involved in relaying real-time traffic as stated earlier. The gateway starts moving to that location and assumed to move in strides to reach intermediate positions. The strides are to be in straight line. At each intermediate position, the gateway checks whether it can still be reachable by the last hops while traveling on the next stride and inform the last hop nodes about its situation. Once, it detects that forwarder nodes are needed, the routes are extended by those nodes and that information is sent to relevant nodes.

The proposed algorithm for gateway relocation is shown in Fig. 4. The gateway monitors the miss ratio for real-time data periodically to detect situations where there is need for relocation (line 1-2). If the gateway motion is justified, a new position is identified (line 3) and the gateway starts to move to its new location (line 4). In order to designate a new location for the gateway, the algorithm finds the node with the biggest *r*-value in line 6. In case of ties, the one closest to the gateway is picked in lines 7-8. Once the new location is found, the algorithm checks whether moving to that location will require employing forwarder nodes (line 9). If that is the case, alternative locations are searched in line 10. If no location is available without involvement of new forwarders, the algorithm looks for nodes (line 18-25) that are not currently part of any data paths in line 12 and starts moving the gateway to the new position in line 15. When such forwarders cannot be identified, relocation is not performed (line 13-14). It should be noted that analysis specific to the network architecture and the detailed routing protocol can be triggered at this stage to further check the suitability of the move by comparing the significance of the gains of

```
Check the miss ratio at a predefined period
if (Miss ratio < Threshold)
        SearchForNewLocation()
Start moving gateway to NewLocation(G)

SearchForNewLocation()
  Find the node s.t. $m \leftarrow \max_{(j \in LastHopNodes)} (rval^{j})$ /*LastHopNodes: within one hop neighbor of G*/
  if (more than one m is found)
        Get the one closest to G
  /*Check if gateway goes out of transmission range for any of the nodes in LastHopNodes when
  relocated at the position of node m*/
  if (G goes out of transmission range of any of the nodes in LastHopNodes)
        Consider moving it closer to one of its siblings having the biggest r-val
        if (G still goes out of range of any node $j \in LastHopNodes$ )
                FindForwarder(j)
                if (no forwarder can be found)
                    break; /* Do not relocate (or check whether the overhead of the move
                              and of setting new topology is justified) */
                NewLocation(G) ← Pos( m)
else NewLocation(G) ← Pos(m)

FindForwarder(j)
    /*Find a relay node i between the new position of G and node j which is free */
  Find Forwarder i  s.t.{Dist(j,i)<TRange[j] && Dist(i,$G_{new}$)<TRange[i] && $rval^{i}$==0 &&
  Remain_energy(i) > $\delta$}
  Update RouteTable[j];
  if (i not in LastHopNodes) /*Update the list of LastHopNodes accordingly*/
     begin
       Add i to LastHopNodes
       UpdateRouteTable[i];
     end
  Remove j from LastHopNodes
```

Fig. 4: Relocation algorithm for the gateway under QoS traffic

repositioning the gateway to the overhead incurred in setting new topology at the new location. Such analysis can be easily augmented.

3 Experimental Validation

The effectiveness of our approach is validated through simulation. This section describes the underlying network operation, simulation parameters, performance metrics and experimental results.

3.1 Network Operation

We have adapted the network operational model of [21] 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.

In addition, the gateway broadcasts the routing table to all sensors prior to starting or resuming data transmission. Routes are set by using 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*.**

3.2 Environment Setup

In the experiments, the network consists of varying number of sensor nodes (50 to 200) randomly placed 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 [23] with the capacity set to 2Mbps. 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 [24].

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. [18]. 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 the cost function defined for each link is using remaining energy as part of the cost.

We assume that the network is tasked with a multi-target tracking mission in the experiment and the gateway can move with a maximum speed of 5m/sec when needed. 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. 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 [25]. 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. Any target remains active until it leaves the deployment region area.

3.3 Performance Metrics

Given our interest in enhancing on-time delivery of real-time data by employing gateway relocation, 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.
- *Network Throughput:* Defined as the total number of data packets received at the gateway divided by the simulation time.

3.4 Performance Results

In this section we present some performance results obtained through simulation. As a baseline approach, we have used the same underlying routing mechanism with a stationary gateway for i.e., without considering any movement.

Delay and Timeliness: When qualifying the impact on the end-to-end delay and miss ratio for real-time packets we have observed that the relocation approach significantly decreases average delay per real-time packet and provides at least 20% decrease in the deadline miss ratio as shown in Fig. 5 and 6. This is due to the decreased number of hops and queuing delay for the data coming from highly loaded areas when gateway goes closer to those areas. The decrease in the miss ratio is even more significant when the number of sensors is increased, suggesting the positive effect on network scalability.

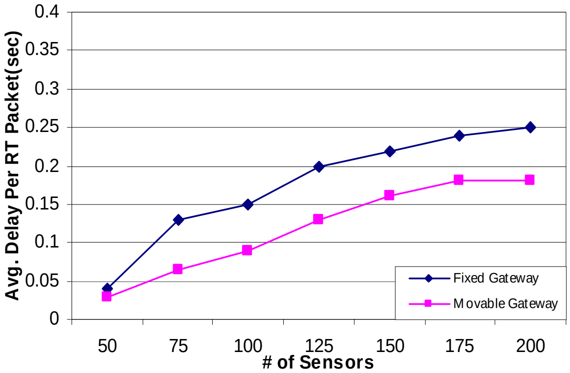


Fig. 5: Average delay per real-time packets with different number of sensors.

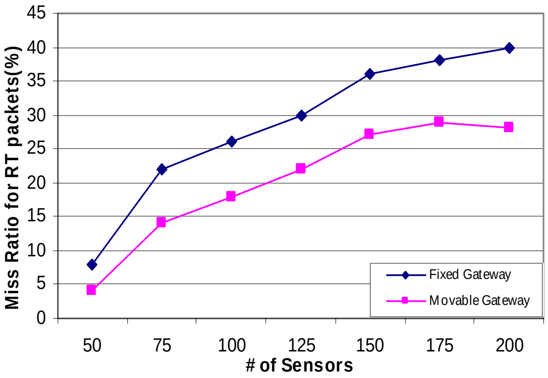


Fig. 6: Deadline miss ratio for real-time packets with different number of sensors.

Energy Consumption: In order to capture the effect of relocation on energy consumption, we have looked at the average energy per packet. The results, depicted in Fig. 7, demonstrate that our approach does bring very negligible energy overhead to the network. Although either some of the last hop nodes increase their transmission range or routes are extended through additional forwarders, the potential energy overhead due to such adjustments is compensated through the energy gain when gateway is relocated towards loaded nodes.

Throughput: When we have looked at the throughput for real-time data, we have observed that relocating the gateway increases real-time data throughput by about 20% compared to the stationary gateway real-time data throughput, as shown in Fig. 8. This is expected since moving the gateway towards heavily loaded nodes and splitting the traffic will help in boosting the number of admitted flows for real-time relaying.

In summary, the simulation results clearly confirm the effectiveness of our approach. The gateway relocation under real-time traffic can provide substantial enhancement in hit ratio of real-time packets without introducing any extra overhead in terms of energy. Such enhancement is more significant when traffic load is high. Moreover, the throughput for real-time data has significantly benefited from such a move.

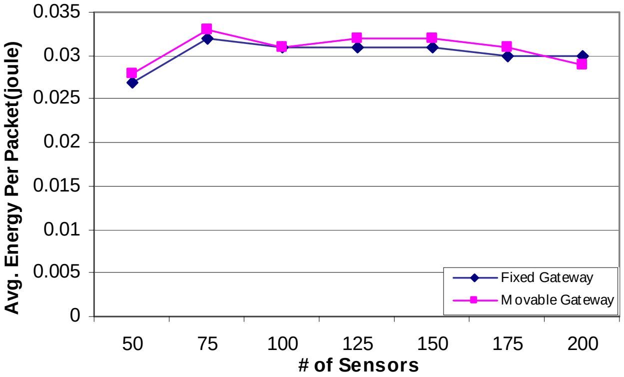


Fig. 7: Average energy per packet consumed under different number of sensors.

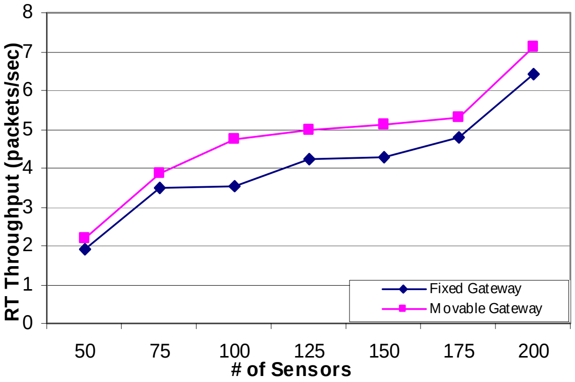


Fig. 8: Real-time data throughput for different number of sensors.

4 Conclusion

In this paper, we have presented an algorithm for effective gateway relocation in order to enhance the performance of routing QoS traffic in wireless sensor networks. At each sensor node, WFQ-based packet scheduler is employed. WFQ allows service sharing for real-time and non-real-time flows and achieves certain delay bounds when used along with leaky bucket constrained sources. Our approach periodically checks the miss ratio for real-time packets and triggers a relocation stimulus for the gateway if the miss ratio is above a certain threshold. In order to designate a new gateway location, our approach finds the node that routes the largest number of real-time packets and checks whether moving to that location or close to that location affects the current routes or not. Relocation is performed only if the new location will decrease the end-to-end delay for most real-time packets without increasing the energy consumption of the network.

The effectiveness of our approach is validated through simulation. Simulation results have demonstrated the effectiveness of our approach showing at least 20% decrease in the deadline miss ratio and real-time data throughput when compared to a fixed gateway model. Moreover, The simulation has indicated that our approach boosts these performance metrics without major negative impact on the consumed energy.

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 Internation 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, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocols for Wireless Microsensor Networks,” *Hawaii International Conference on System Sciences (HICSS '00)*, January 2000.

[8] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in the *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (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 International Conference on Mobile Computing and Networking (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] 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.

[14] K. Akkaya and M. Younis, “Energy-aware delay constrained routing in wireless sensor networks,” Technical Report, Dept. of Computer Science, University of Maryland, Baltimore County, 2003.

[15] 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.

[16] A. Demers et al., “Analysis and simulation of a fair queuing algorithm” in *Journal of Internetworking Research and Experience*, pp 3-26, October 1990.

[17] 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*, Vol. 2, pp. 521-530.

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

[19] K. Akkaya and M. Younis, “An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks,” *Proc. of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003)*, Providence, RI, May 2003.

[20] M. Younis, M. Bangad and K. Akkaya, “Base-Station Repositioning For Optimized Performance of Sensor Networks,” in the *Proceedings of the IEEE VTC 2003 - Wireless Ad hoc, Sensor, and Wearable Networks*, Orlando, Florida, October 2003.

[21] M. Younis, M. Youssef, K. Arisha, “Energy-Aware Routing in Cluster-Based Sensor Networks”, in the Proceedings of IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), Fort Worth, Texas, October 2002.

[22] 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.

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

[24] A. Chandrakasan, et al., "Power Aware Wireless Microsensor Systems", Keynote Paper ESSCIRC, Florence, Italy, September 2002.

[25] K. Danilidis et al., “Real-time Tracking of Moving Objects with an Active Camera,” in *Real-Time Imaging Journal,* Vol. 4, No.1 pp.3-20. February 1998.