

## Research Article

# An Energy-Efficient Data Delivery Scheme for Delay-Sensitive Traffic in Wireless Sensor Networks

**Harshavardhan Sabbineni and Krishnendu Chakrabarty**

*Department of Electrical and Computer Engineering, Duke University, Durham, NC 27705, USA*

Correspondence should be addressed to Harshavardhan Sabbineni, hvardhan@ee.duke.edu

Received 15 June 2009; Revised 10 November 2009; Accepted 20 December 2009

Copyright © 2010 H. Sabbineni and K. Chakrabarty. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a novel data-delivery method for delay-sensitive traffic that significantly reduces the energy consumption in wireless sensor networks without reducing the number of packets that meet end-to-end real-time deadlines. The proposed method, referred to as SensiQoS, leverages the spatial and temporal correlation between the data generated by events in a sensor network and realizes energy savings through application-specific in-network aggregation of the data. SensiQoS maximizes energy savings by adaptively waiting for packets from upstream nodes to perform in-network processing without missing the real-time deadline for the data packets. SensiQoS is a distributed packet scheduling scheme, where nodes make localized decisions on when to schedule a packet for transmission to meet its end-to-end real-time deadline and to which neighbor they should forward the packet to save energy. We also present a localized algorithm for nodes to adapt to network traffic to maximize energy savings in the network. Simulation results show that SensiQoS improves the energy savings in sensor networks where events are sensed by multiple nodes, and spatial and/or temporal correlation exists among the data packets. Energy savings due to SensiQoS increase with increase in the density of the sensor nodes and the size of the sensed events.

## 1. Introduction

Wireless sensor networks (WSNs), consisting of large numbers of sensor nodes, have enabled a new monitoring and sensing paradigm for large geographical areas. WSN applications include detection of forest fires, habitat monitoring, and target tracking in the battlefields [1–3]. In a sensor network, the data generated by individual sensor nodes is forwarded to the sink for further processing. The sensed information in a sensor network typically has the following characteristics.

- (1) Sensor network applications are data centric and are mainly concerned with the data generated by the sensor nodes and not the ids of the reporting sensor nodes.
- (2) The generated data can have different real-time deadlines. For example, in a forest fire detection application, the data that indicates the presence of a forest fire may have a shorter real-time deadline to reach the sink node than the data that consists of periodic temperature readings.

- (3) The data generated by sensor nodes has both spatial and temporal correlation. Such correlation of the data can be leveraged by performing in-network data aggregation, thereby potentially reducing the number of packet transmissions and hence, extending the lifetime of the network.

Timely delivery of sensor-generated data in an energy-efficient way is the subject of this paper. We are interested in event-driven sensor networks, where multiple detections of the same event can occur within a short time period in a relatively close geographic region. Many applications of sensor networks such as habitat monitoring and target tracking fit this model, as events in these sensor networks often occur in specific geographic regions. For example, in a habitat monitoring sensor network application, a query might look like this: notify me within 2 minutes whenever the number of animals in a geographic region  $[x, y]$  increases above 50?

Traditional service differentiation protocols such as IntServ and DiffServ [4] for wired networks support real-time traffic with latency constraints through end-to-end

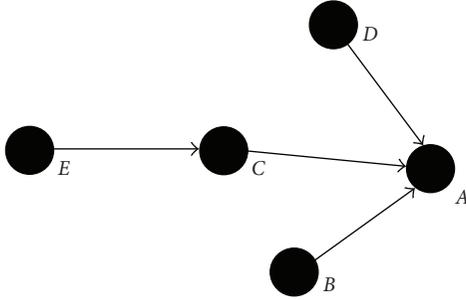


FIGURE 1: In a sensor network, performing data aggregation may increase the latency of the data packets. In this connected topology, data generated by nodes  $B$ ,  $C$ ,  $D$ , and  $E$  is correlated. If node  $C$  waits for packets from node  $E$  to perform data aggregation, this may increase the delay for its packets to reach the sink node  $A$ .

signaling and resource reservation. However, such protocols are not suitable for wireless sensor networks due to several reasons, including dynamic topological changes such as node addition, failure, and node mobility.

Designing an energy-efficient data delivery scheme with latency constraints is a difficult challenge in wireless sensor networks. There is a trade-off between performing in-network data aggregation and achieving timeliness. Structures that optimize data aggregation by improving path sharing increase the queuing delay at relay nodes due to increased waiting time for the arrival of the data packets to be aggregated. In real-time applications, such increased queuing delays typically result in longer packet delivery latencies and can make the packets miss their timeliness deadlines and thus can overshadow the energy savings of the in-network aggregation. An example is shown in Figure 1. In Figure 1, if each node that sensed the event sends a packet to the sink, there will be five packets from nodes  $B$ ,  $C$ ,  $D$ , and  $E$  traveling towards the sink node  $A$ . However, if node  $C$  aggregates the data from itself and node  $E$ , only four packets have to travel towards the sink. However, this may cause packets from node  $C$  to miss their real-time deadline.

We have the following goals for designing a data delivery scheme for delay-sensitive traffic in sensor networks.

- (i) *Localized Algorithms*. It should only use localized algorithms and not depend on global state information for data delivery.
- (ii) *Energy Efficiency*. The data delivery scheme should be energy efficient as sensor nodes are resource constrained.
- (iii) *Scalability*. The scheme should be scalable to large and dense sensor networks that consist of thousands of nodes.

In this paper, we propose SensiQoS, a novel data delivery scheme for delay-sensitive traffic that significantly reduces the energy consumption in wireless sensor networks without reducing the number of packets that meet end-to-end real-time deadlines. SensiQoS leverages the inherent properties of the data generated by events in a sensor network, such as

spatial and temporal correlation, and realizes energy savings through application-specific in-network aggregation of the data. SensiQoS maximizes the energy savings by adaptively waiting for packets from upstream nodes to perform in-network processing without missing the real-time deadline of the data packets. It uses a distributed scheduling scheme, where nodes make localized decisions on when to schedule a packet for transmission to save energy and to which neighbor they should forward the packet to meet its end-to-end real-time deadline. We also present a randomized algorithm where nodes adapt locally to the network traffic to maximize energy savings in the network.

The rest of the paper is organized as follows. Section 2 summarizes related work. Section 3 presents the proposed QoS protocol. Section 4 presents the analysis of the energy savings of the protocol. Section 5 presents a performance evaluation of the proposed protocol. Finally, Section 6 concludes the paper.

## 2. Related Work

SPEED [5] protocol is designed to provide soft end-to-end deadline guarantees for real-time packets in sensor networks. However, it provides only one network-wide speed, which is not suitable for differentiating various traffic with different deadlines. MMSPEED [6], scheme of Felemban et al., utilizes the concept of SPEED to provide real-time guarantees in both the reliability and timeliness domains. In MMSPEED, each node takes localized decisions to provide both timeliness and reliability guarantees using location information. Each node supports multiple speeds to its neighbors, and a packet is kept in a queue that can make the packet meet its deadline. To support reliability guarantees, MMSPEED sends the packet through multiple routes to support the needed reliability. SensiQoS differs from MMSPEED. MMSPEED does not take advantage of the correlated nature of the data generated in sensor networks and routes each packet to the sink to meet its deadline. SensiQoS allows nodes to aggregate the data while still allowing the packets to meet their timeliness deadline.

Several data aggregation techniques [7, 8] were proposed in the literature to save communication energy. Finding the optimal data aggregation tree is an NP-hard problem [9]. GIST [10] proposes a semistructured approach to construct a group-independent spanning tree in polynomial time. Cristescu [11] considers the problem of joint rate allocation and transmission structure optimization for network-correlated data gathering purposes. The work in [11] proposes two solutions based on a combination of Slepian-Wolf coding and clustering. However, these schemes do not consider latency constraints during data aggregation.

QAWSN [12] considers the practical limits on achievable energy improvements using correlation-aware aggregation structures over correlation-unaware aggregation structures. The authors conclude that the energy improvements in Steiner Minimum Trees (SMTs) and Shortest Path Trees (SPTs) are not very different, and SPT is more suited for sensor networks. Synchronization of multiple levels of data

fusion [13] considers this problem but it is based on a centrally calculated wait-time. A geographic gossip algorithm [14] that exploits geographic information has been proposed for achieving data aggregation in sensor networks. By routing packets to far away nodes in the network, geographic gossip achieves rapid diffusion of information and aggregation.

Data aggregation subject to latency constraints is a more difficult problem, and only a few solutions have been proposed in the literature [15]. A centralized solution to the problem of data aggregation subject to latency constraints is proposed in [16]. The authors propose a Weighted Fair Queueing methodology to schedule packets at relay nodes. When real-time traffic is involved, the bandwidth ratio for each relay node on the path from the source to the gateway is determined at the gateway. This solution is not scalable to situations where multiple sinks are involved in large sensor networks.

An interesting study that considers data aggregation subject to latency constraints is reported in [17]. The work in [17] considers the problem of providing QoS for hierarchical sensor networks. The problem of multipath relaying is modeled as a linear programming problem, and two centralized and optimal solutions are proposed for multipath relaying with and without delay constraints. A distributed QoS routing [18] is proposed that selects a network path with sufficient resources to satisfy a given delay criterion using reinforcement learning technique. The problem of soft QoS provisioning is modeled using integer programming, and a distributed hop-based routing algorithm is proposed [19] for providing end-to-end QoS in wireless sensor networks. A QoS-MAC protocol is proposed in [20]. The work in [21] considers the problem of multiconstrained QoS multipath routing for wireless sensor networks. The worst-case delay in a sensor network is dependant on the node degree. However, achieving soft-QoS provisioning using [19] involves carefully tuning parameters to estimate the link reliability, and inaccuracies in tuning may result in many packets missing their real-time deadline. SensiQoS does not involve any parameters that need to be tuned and hence is more robust.

### 3. SensiQoS Design

In a sensor network, data sensed by sensors present in geographically close proximity is more likely to be correlated. Similarly, data sensed by sensor nodes within a certain time interval after the occurrence of the event is more likely to be correlated. SensiQoS leverages such spatial and temporal correlation present in the sensed data in a sensor network by aggregating correlated data packets that belong to the same interest at a relay node. SensiQoS employs adaptive algorithms to determine the wait-time of data packets at each relay node such that they do not miss their real-time deadline.

In SensiQoS, sink sends interest packet describing the data it seeks along with the priority and timeliness requirements and an application-specific in-network aggregation function. A sample interest used in SensiQoS is shown as follows.

*Type:* four-legged animal.

*Monitoring duration:* 5 minutes.

*Real-time deadline:* 5 seconds.

*Region of interest:*  $[-100, -100, 100, 100]$ .

*Aggregation function:* maximum.

*Priority class:* high.

The design of SensiQoS meets the following goals. First, SensiQoS schedules the transmission of packets such that each packet is able to meet its real-time deadline. Second, it aggregates packets belonging to the same query, thereby increasing network lifetime but without suffering an increase in the number of packets that miss their real-time deadline. Third, it improves the energy savings realized through aggregation using a modification of the shortest path tree algorithm. Fourth, it only uses information local to each node in a decentralized manner for packet scheduling and forwarding.

SensiQoS is proactive: each node periodically broadcasts the HELLO messages that contain the node's state (buffer space information). In addition to periodic beaconing, SensiQoS uses two types of on-demand beacons, namely, a delay estimation beacon and a backpressure beacon. Delay estimation beacon is used to estimate the delay between two neighbor nodes and backpressure beacon is used to quickly notify upstream nodes about traffic congestion in the network. SensiQoS interacts with both the MAC layer and the routing layer in the sensor node.

In existing ad hoc networks, packets are typically forwarded in First Come First Serve (FCFS) order. FCFS scheduling does not work well in real-time networks where packets have different priorities and different end-to-end deadlines. In SensiQoS, packet scheduling policy is both aggregation aware and deadline aware. Aggregation aware means that packets are forwarded to nodes that may have data to aggregate from other packets, and packet transmission is delayed if there is an opportunity to aggregate its contents with other incoming packets. Deadline aware means that packets are scheduled such that they do not miss their real-time deadline.

SensiQoS is energy efficient. Correlated data packets present in a node's cache that belong to the same interest are aggregated using an application-specific aggregation function. This in-network aggregation of data reduces the number of data packets that travel from source to the sink and consequently the total number of data transmissions. As the cost of communication forms a major part of energy utilization for a node, SensiQoS saves a significant amount of energy and extends the network lifetime.

SensiQoS is adaptive as well. Each node adaptively determines the delay each packet can tolerate at that node and attempts to maximize the energy savings by delaying packets that have a chance of getting aggregated because of arrival of packets belonging to the same interest. SensiQoS also adapts to the network conditions by probabilistically dropping packets of low priorities in the event of congestion.

Source	Destination	Packet deadline	Packet priority	Interest ID	Timestamp
--------	-------------	-----------------	-----------------	-------------	-----------

FIGURE 2: Packet header for SensiQoS.

**3.1. Location Information.** SensiQoS uses location information to provide energy-efficient packet delivery. The location information used in SensiQoS protocol may be provided by the Global Positioning System (GPS) [22] or any other localization scheme. Several localization schemes are also available in the literature for wireless sensor networks. In [23], a light-weight localization for supporting range and distance queries is proposed. This scheme uses received signal strength information (RSSI) as the ranging method and works without any external infrastructure. Localization using a new radio interferometric positioning technique that provides both high accuracy and long range is proposed in [24]. In [25], a scheme is presented to estimate the relative location of nodes by having only a few nodes in the sensor network with GPS capability. It uses the received signal strength information (RSSI) as the ranging method to obtain accurate location estimates. The work in [26] uses an ad hoc localization technique called *Calamari* in combination with a calibration scheme to calculate distance between two nodes using a fusion of RF-based RSSI and acoustic time of flight (TOF). Acoustic ranging [27] can be used to get fine-grained position estimates of nodes. The work in [28] proposes a low-cost localization technique that uses time-of-arrival ranging. Recursive schemes such as [29] can also be used to get fine-grained position estimates of sensor nodes with error within 28 cm for nodes of 40 m radio range. Spotlight [30] uses spatiotemporal properties of well-controlled events such as light for localization and achieves a high amount of accuracy to within 10 cm localization error in under 10 minutes.

**3.2. Packet Header Format.** Nodes running the SensiQoS protocol use the following header for their data packets. It consists of both the sourceID and sourceLocation in the source field as well as the destinationID and the destinationLocation in the destination field. It also contains packet deadline and packet priority, two parameters that determine the quality-of-service the packet receives from the network. Having packet priority as a separate parameter assists SensiQoS in determining the quality-of-service for a packet when there is contention for resources with the higher priority packet receiving better service. Finally, it contains the InterestID that is used by the relay nodes in identification during the the aggregation process. Packets in a node cache with the same InterestID are aggregated if there is a high degree of correlation among the packets as determined by an application-specific aggregation function. The fields in a SensiQoS packet header are shown in Figure 2.

**3.3. SPEED Protocol.** SensiQoS achieves energy savings by aggregating correlated packets in the sensor network while delivering packets to their destination before the real-time deadline. To ensure the delivery of a packet to its destination

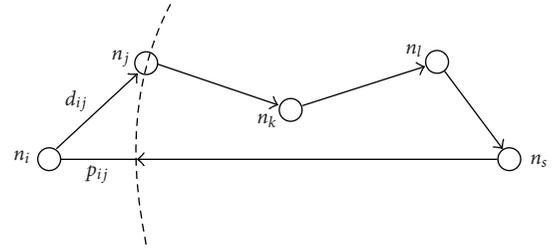


FIGURE 3: Illustration of SPEED protocol.

within the real-time deadline, SensiQoS leverages the idea of a single network-wide speed provided by the SPEED protocol [5]. We briefly describe SPEED protocol through an example. Consider a data packet  $p$  generated by a source node  $n_i$ . Suppose that nodes  $n_j$ ,  $n_k$ , and  $n_l$  are three of the intermediate nodes that  $p$  passes through to reach its final destination node  $n_s$  as shown in Figure 3. The geographical distances from  $n_i$ ,  $n_j$ , and  $n_k$  to  $n_s$  are  $d_{i,s}$ ,  $d_{j,s}$ , and  $d_{k,s}$ , respectively. Suppose that the delay experienced by  $p$  at node  $n_i$  including queuing, processing, and MAC collision resolution to reach node  $n_j$  is  $\text{delay}_{i,j}$ . This means that packet  $p$  has traveled an Euclidean distance of  $d_{i,s} - d_{j,s}$  towards the final destination node  $n_s$ . The speed with which the packet  $p$  has progressed towards the destination node  $n_s$  in unit time is then given by

$$S_{i,j}^k = \frac{d_{i,s} - d_{j,s}}{\text{delay}_{i,j}}. \quad (1)$$

SPEED guarantees that packet  $p$  will travel each hop with a chosen preset speed. To ensure its single network-wide speed guarantee, SPEED maintains two sets of nodes for each query. There are a neighbor set that consists of all nodes that are within the transmission range and a forwarding set that consists of nodes that are closer to the destination than the node itself. SPEED also maintains the speed with which a packet can be forwarded to each neighbor node through the use of delay beacons. Each node in the network is able to forward the packet to a downstream node such that the progress speed is greater than the preset speed  $S$ . However, due to congestion in the sensor network, the queuing and channel contention delays can interfere with real-time guarantees. SPEED achieves congestion control by probabilistically dropping packets to maintain at least one node that satisfies the progress speed criterion. Nodes also inform upstream nodes to reduce congestion by sending backpressure beacons. SensiQoS leverages this network-wide speed guarantee provided by SPEED to determine wait-time for data packets at a node using its packet scheduler.

**3.4. SensiQoS Packet Scheduler.** A key component of SensiQoS is the packet scheduler that determines the amount of time a packet can wait at a relay node. Each data packet may traverse multiple relay nodes during its journey from the source to the sink. Upon its arrival at a relay node, SensiQoS protocol calculates in a localized way the amount of time the packet can afford to wait at that relay node.

Better aggregation can be achieved by waiting for a longer time as potentially more packets arrive from the upstream nodes to this relay node. However, this will increase the amount of delay experienced by the packet, and determining the amount of wait-time without a packet missing its real-time deadline is a challenging problem. SensiQoS adaptively determines the wait-time such that the packet is able to meet its real-time deadline.

Consider a packet  $p$  belonging to an event  $E$  generated by a source node, say node  $n_g$ , and has a real-time deadline of  $t_D$  seconds. Suppose that the network-wide speed supported by the sensor network is  $S$  m/s. Suppose the Euclidean distance between the source and the sink node  $n_s$  is  $L$  m. If the packet progresses towards the sink node  $n_s$  at a speed of  $S$  m/s, the amount of time it takes the packet to reach the sink  $n_s$  is  $t_{g,s}$  and is given by the following equation:

$$t_{g,s} = \frac{L}{S}. \quad (2)$$

If  $t_{g,s} > t_D$ , the network cannot support the speed required by this packet to reach its destination within its real-time deadline. However, if  $t_{g,s} < t_D$ , the packet will arrive at the sink node earlier than its real-time deadline. We claim that by distributing the remainder time  $t_D - t_{g,s}$  in an intelligent way among the relay nodes overall data aggregation can be improved without packets missing their real-time deadlines. We denote this remainder time as  $\text{rem\_time}(p)$ . With the single network-wide speed guarantee of  $S$  m/s, a packet can spend  $\text{rem\_time}(p)$  in the network and can still meet its real-time deadline. The wait-time for  $p$  at a relay node  $n_r$  is a function of  $\text{rem\_time}(p)$ :

$$\text{wait\_time}(p, n_r) = \phi(\text{rem\_time}(p)). \quad (3)$$

Nodes running SensiQoS delay the transmission of packets for an interval equal to this wait-time. This delay in transmission of packet  $p$  provides a “window of opportunity” for the node to aggregate  $p$  with other packets of similar interest that may arrive at a relay node  $n_r$  during the “window of opportunity”. These delayed packets are stored in the node’s local cache. When packets that belong to the same query arrive at the sensor node, they are aggregated using an application-specific aggregation function specified in the query. The result of the aggregation is then forwarded to the downstream neighbor towards the sink.

The time a packet waits at a relay node is determined adaptively at each hop, instead of waiting for a fixed amount of time at each hop. Adaptively determining the wait-time ensures that a packet meets its real-time deadline in spite of local changes to the traffic conditions. Specifically, the SensiQoS module of a relay node  $n_r$  calculates the Euclidean distance  $L_{r,u}$  traveled by packet  $p$  from an upstream node  $n_u$  and determines the wait-time using the following equation:

$$\text{wait\_time}(p, n_r) = \frac{t_D}{L} L_{r,u} - \frac{L_{r,u}}{S}, \quad (4)$$

where  $L_{r,u}$  is given by

$$L_{r,u} = L_{u,s} - L_{r,s}, \quad (5)$$

$t_D$  is the real-time deadline of the packet  $p$ , and  $L$  is the distance from the source node to the destination node.

**3.5. Data Aggregation.** The energy efficiency of a data gathering scheme depends on the amount of correlation that exists between the sensed data. If the sensed data is perfectly correlated and multiple data packets can be reduced to the size of a single data packet after aggregation, the Steiner Minimum Tree (SMT) is an optimal routing structure [12]. However, [12] has shown that correlation-unaware approaches such as shortest path trees provide a comparable performance for many network scenarios with significantly less overhead. Based on the above observation, SensiQoS uses shortest path routes between nodes for aggregation.

In a data gathering application that performs aggregation, the delay at each hop of the aggregation tree should include transmission delay, contention delay, and aggregation delay. In the aggregation process, data packets may need to be held for some time at intermediate relay nodes to perform aggregation. Aggregation delay in SensiQoS comprises of the processing time for aggregation at each node and the SensiQoS determined wait-time.

The wait-time of the aggregated packet at a relay node  $n_r$  is the smallest remaining wait-time among all packets getting aggregated:

$$\text{wait\_time}(p_{\text{agg}}, n_r) = \min_{j=1}^M \text{rem\_wait\_time } p_j, n_r, \quad (6)$$

where  $M$  is the total number of packets that are being aggregated,  $\text{rem\_wait\_time}(p_j, n_r)$  is the remaining wait time of packet  $p_j$ , and  $\text{wait\_time}(p_{\text{agg}}, n_r)$  is the wait time of the aggregated packet  $p_{\text{agg}}$ .

**3.6. MAC-Layer QoS Support.** The proposed SensiQoS protocol determines when to schedule a packet for transmission but depends on the low level MAC layer to support prioritized access to the shared medium depending on the priority class of the packet.

To provide service differentiation, the 802.11e MAC protocol [31] provides a differentiated channel access mechanism called Enhanced Distributed Channel Access (EDCA). EDCA provides QoS support by associating different channel access parameters with different traffic categories. The following access parameters are associated with each traffic category. SensiQoS maps each traffic category to a priority class.

- (i) *Arbitration Interframe Spacing (AIFS)*. It is the minimum time interval to wait before backing off. A high priority traffic category has a shorter AIFS length.
- (ii) *Transmission Opportunity (TXOP)*. It is the maximum duration for which the node can transmit. A TXOP allows a node to transmit multiple data frames without the need to restart the channel acquisition mechanism.
- (iii) *Contention Window Parameters ( $CW_{\min}$  and  $CW_{\max}$ )*. These parameters determine the number of random slots to wait before starting transmission. Assigning smaller values of  $CW_{\min}$  to high priority traffic category gives it more TXOPs compared to a low priority traffic category. Similar is the case with  $CW_{\max}$ .

In addition to prioritized access to the shared medium, SensiQoS also requires the following services from the MAC layer.

- (i) *One-Hop Delay.* We have modified MAC 802.11e such that it maintains a one-hop delay for each priority class to each node in the forwarding set. Each packet is annotated with a timestamp when it is sent out. When the ACK for the packet is received, another time stamp is associated with it. The difference between the two timestamps is taken as the one-hop delay of the packet to the destination node.
- (ii) *Next-Hop Node.* SensiQoS also utilizes the MAC layer information for making routing decisions. When MAC layer receives an RTS, it updates SensiQoS with the packet's priority class and its destination. SensiQoS prefers to route packets of a priority class to a node amongst the forwarding set that has received a packet of that priority class. If there are multiple nodes that have received packets of that priority class, the node that has received the packet latest is considered first. This ensures that packets have a better chance of getting aggregated at the downstream node.

**3.7. Feedback-Based Congestion Control.** SensiQoS protocol adaptively delays a packet's transmission such that it does not miss its real-time deadline. However, a node may still miss its deadline due to dynamic network conditions such as congestion, node failure, or node mobility. SensiQoS uses feedback-based congestion control to support soft real-time services when traffic congestion increases in the network and nodes fail to support the preset network-wide speed. SensiQoS uses the backpressure beacons provided by SPEED to propagate the feedback about packets that missed their deadlines to the upstream node that sent the packet. SensiQoS organizes packets belonging to different priority classes in different queues while they wait in a node's local cache. When a congestion notification through a backpressure beacon is received from a downstream node, SensiQoS drops packets probabilistically from the lowest priority class queue followed by the next lower priority class queue and so on. This localized feedback ensures that SensiQoS recovers from dynamic network conditions. Figure 4 shows an example of how packets are organized in different priority queues for a sensor network application that generates packets that belong to three different priority classes.

#### 4. Analysis

In this section we discuss the various factors that affect the selection of the network-wide speed as well as the real-time deadlines for the events in the sensor network and analyze the energy savings realized by using SensiQoS protocol. Suppose that the set of  $k$  sensor nodes is denoted by  $G = \{n_1, n_2, \dots, n_k\}$ . Suppose that  $\mathbf{n}_i$  is the location vector for the location of the sensor node  $n_i$  where  $\mathbf{n}_i = \langle x_i, y_i \rangle$  where  $x_i$  and  $y_i$  are the  $x$  and  $y$  coordinates for the location of the

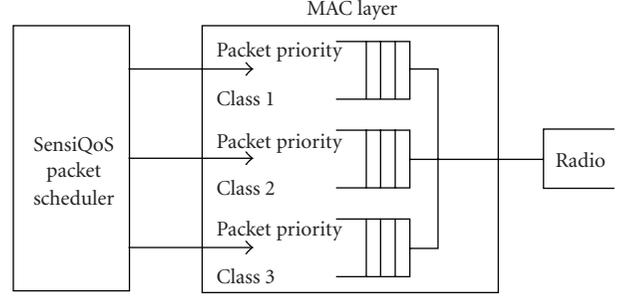


FIGURE 4: Packet organization in SensiQoS.

sensor node  $n_i$ . Suppose that  $d_{ij} = \|\mathbf{n}_i - \mathbf{n}_j\|$  is the distance between the sensor nodes  $n_i$  and  $n_j$ .

**4.1. Network-Wide Speed.** Suppose that  $t_i^p$  represents the propagation time for a packet  $p$ ,  $t_i^w$  represents the wait time as determined by SensiQoS, and  $t_i^q$  represents the queuing time as well as the channel contention delay for a sensor node  $n_i$ . Thus, the total time a packet spends at a relay node is given by

$$t_i = t_i^p + t_i^w + t_i^q. \quad (7)$$

This is a combination of the propagation delay, waiting time, and the channel acquisition delay. The propagation delay depends on the laws of physics. Waiting time depends on the SensiQoS algorithms, and the channel acquisition delay depends on the congestion in the network. From (4),  $t_i^w$  clearly depends on the distance to the sink, the real-time deadline of the packet, as well as the network-wide speed supported by the network. The summation of the time a packet spends at all the relay nodes must be less than the real-time deadline of the packet:

$$\sum_{i=1}^R t^i \leq t_D, \quad (8)$$

$$\sum_{i=1}^R t_p^i + \sum_{i=1}^R t_w^i + \sum_{i=1}^R t_q^i \leq t_D,$$

where  $R$  is the number of relay nodes between the source node and the sink. The minimum speed the network needs to support arises when the wait-time determined by SensiQoS is zero, that is,

$$\sum_{i=1}^R t_w^i = 0. \quad (9)$$

Note that  $t_i^q$  depends on the traffic conditions and the congestion in the network. Hence the sensor network designers must choose the network-wide speed that can support the desired network traffic. Network designers must also choose a real-time deadline for each event based on the nature of the event. When appropriate, choosing a longer real-time deadline will let SensiQoS provide a greater

amount of aggregation by allowing the packets to stay in the network for a longer period of time. For each event, the network-wide speed must be able to transport a packet within its real-time deadline. The minimum network-wide speed guarantee required is the maximum over all the minimum required speeds for each event. In other words,

$$S = \max_{i=1,2,\dots,e} S_i, \quad (10)$$

where  $S_i$  is the minimum network-wide speed required for event  $i$ , and the number of events is  $e$ .

**4.2. Expected Number of Transmissions.** Suppose, for the ease of analysis, that the nodes in the network are organized in the form of a  $d$ -ary tree with the sink node as the root of the tree. Suppose that each node generates packets using a Poisson process with rate  $\lambda$  and exponential packet transmission times and injects those packets into the network. Suppose that each packet has a real-time deadline of  $t_D$  seconds, and suppose that the network supports a network-wide speed of  $S$  m/s. To simplify the analysis, we suppose that all nodes at each level are equidistant from the sink and all nodes except the sink generate data.

Since the packets are generated using a Poisson process with rate  $\lambda$ , the packet generation times at a node is an exponentially distributed random variable with parameter  $\lambda$ ,  $\lambda > 0$ , with a probability density function  $f(x)$  and a cumulative distribution function  $F(x)$  as

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0, \\ 0, & \text{if } x < 0, \end{cases} \quad (11)$$

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases}$$

Consider a  $d$ -ary tree with  $h$  levels. Nodes present at level  $h$  are the leaf nodes. More packets get aggregated at level  $h-1$  than at level  $h-2$  as more nodes are present at level  $h-1$ .

The distance from a node  $n_i$  present at level  $h$  to the sink is given by

$$d_{i,s} = \|\mathbf{n}_s - \mathbf{n}_i\|. \quad (12)$$

When a packet travels from a node  $n_i$  at level  $h$  to a downstream node at  $n_j$  present at level  $h-1$ , it makes a progress of

$$L_{ij} = \|\mathbf{n}_s - \mathbf{n}_i\| - \|\mathbf{n}_s - \mathbf{n}_j\|. \quad (13)$$

The time taken by packet  $p$  to travel from a node at level  $h$  to the sink is  $L_h/S$ . The minimum time taken by the packet  $p$  to travel from level  $h$  to level  $h-1$  is  $L_{ij}/S$  where  $S$  is the network-wide speed supported by the underlying SPEED protocol.

The  $\text{wait\_time}(p, n_j)$  at the relay node  $n_j$  calculated by SensiQoS is

$$\text{wait\_time}(p, n_j) = \frac{t_D}{L/L_{ij}} - \frac{L_{ij}}{S}. \quad (14)$$

We use the following two properties of Poisson distribution [32] for determining the expected number of packets that are aggregated in SensiQoS:

- (1) the sum of  $n$  Poisson processes with parameter  $\lambda_i$  is a Poisson process with parameter  $\lambda = \sum_{i=1}^n \lambda_i$ ;
- (2) the number of packets generated by Poisson distribution in an interval  $T$  is a random variable whose expectation is  $\lambda T$ .

A consequence of these properties is that the expected number of transmissions from a node present at level  $h$  to nodes at level  $h-1$  during the wait-time of a packet at a relay node  $n_r$  at level  $h-1$ ,  $A_{h,h-1}$  is given by

$$A_{h,h-1} = \lambda \times \text{wait\_time}(p, n_r). \quad (15)$$

All packets that belong to the same interest will be aggregated at level  $h-1$ . The total number of packet transmissions at level  $h-1$ ,  $A_{h-1}$  is given by

$$A_{h-1} = \frac{1}{\text{wait\_time}(p, n_r)} \times (d^h + d^{h-1}). \quad (16)$$

In a  $d$ -ary tree, the number of packet transmissions at level  $h-2$  and lower is much smaller than the number of packet transmissions at level  $h-1$ .

Thus the expected number of packet transmissions,  $A_{\text{Total}}$ , for the total time  $T$  is given by

$$A_{\text{Total}} = N \approx \frac{T}{\text{wait\_time}(p, n_r)} \times (d^h + d^{h-1}). \quad (17)$$

This equation shows that the total number of packet transmissions in SensiQoS is inversely proportional to the amount of wait-time at a downstream sensor node. The longer the wait-time, the fewer the number of packet transmissions and consequently the larger will be the energy savings. It is also proportional to the speed supported by the sensor network.

**4.3. Impact of Localization Error.** So far we have assumed that the sensor nodes have perfect knowledge of their geographic location. An error in the location estimation will affect the minimum network-wide speed supported by the sensor network. We investigate the effect of location error on the energy savings of SensiQoS.

Suppose that a single node  $n_i$  at hop level  $h-1$  has its location estimated incorrectly by the localization scheme. Suppose that the location vector for the estimated location  $\mathbf{n}_i^{\text{est}}$  is  $\mathbf{n}_i^+$  or  $\mathbf{n}_i^-$ . The location  $\mathbf{n}_i^+$  is farther from the sink than the actual location  $\mathbf{n}_i$  while the location  $\mathbf{n}_i^-$  is closer to the sink than the actual location  $\mathbf{n}_i$ . We consider two cases. First, let  $\mathbf{n}_i^{\text{est}} = \mathbf{n}_i^+$ . The change in the estimated progress toward the sink because of the location error is given by

$$L_{i^+j}^{\text{est}} = \|\mathbf{n}_s - \mathbf{n}_i^+\| - \|\mathbf{n}_s - \mathbf{n}_j\|, \quad (18)$$

where

$$L_{i^+j}^{\text{est}} = L_{ij} + L_{i^+}. \quad (19)$$

The wait-time for packet  $p$  at node  $n_j$  is given by

$$\text{wait\_time}(p, n_j) = L_{jst} \times \left( \frac{D}{L} - \frac{1}{S} \right). \quad (20)$$

The increase in wait-time because of the error in location estimation is given by

$$\text{wait\_time}^\delta(p, n_j) = L_{iit} \times \left( \frac{D}{L} - \frac{1}{S} \right). \quad (21)$$

In this case, the calculated wait-time is longer than the actual wait-time. This will result in the packet staying longer at the downstream node  $n_j$  and consequently missing the deadline. Now consider the case when  $\mathbf{n}_i^{\text{est}} = \mathbf{n}_i^-$ . The change in the estimated progress toward the sink because of the location error is given by

$$\begin{aligned} L_{ij}^{\text{est}} &= \|\mathbf{n}_s - \mathbf{n}_i^-\| - \|\mathbf{n}_s - \mathbf{n}_j\|, \\ L_{ij}^{\text{est}} &= L_{ij} - L_{iit}. \end{aligned} \quad (22)$$

The reduction in wait-time because of the error in location estimation is given by

$$\text{wait\_time}^\delta(p, n_j) = L_{iit} \times \left( \frac{D}{L} - \frac{1}{S} \right). \quad (23)$$

In this case, the calculated wait-time is shorter than the correct wait-time. This will result in the packet getting transmitted early at the upstream node  $n_j$  and consequently losing the opportunity to potentially aggregate more packets.

The change in the number of packets that will not be aggregated is given by

$$\lambda \times \text{wait\_time}^\delta(p, n_i). \quad (24)$$

**4.4. Space Complexity.** SensiQoS uses buffer space available in sensor nodes to store data packets for aggregation. The amount of buffer space required is directly proportional to the number of interests present in the node's local cache and the amount of correlation among data packets. For statistical queries such as min, max, and avg, two pieces of data can be combined and reduced to the same size as that of a single piece. In such cases, the amount of buffer space required is equal to the number of interests present in the cache. If data packets that arrive at a node are only partially correlated and in-network aggregation of the data packets results in more than one packet, then the amount of buffer space required is  $O(p)$ , where  $p$  is the number of packets resulting after aggregation. Techniques such as those in [33] have been developed in the literature to overcome the memory limitations in sensor networks by leveraging Flash-based virtual memory. If there is no correlation amongst data packets and aggregation is not possible, transmitting a single packet with all the data items can still provide savings for SensiQoS provided header length is much larger compared to the size of each data item.

**4.5. Time Complexity.** As can be seen in Algorithm 1, the processing in the procedure *recvDataPacket* depends on the time complexity of the application-specific aggregation function. Wait-time calculation can be done in  $O(1)$  time at any relay node for any source-destination pair.

## 5. Performance Evaluation

To measure the effectiveness of SensiQoS, we conducted extensive simulations of the proposed SensiQoS protocol using the J-SIM network simulator. J-SIM is an open-source network simulator that provides a modeling and simulation environment for wireless sensor networks [34]. The performance of SensiQoS is compared with the following protocols.

- (i) *MMSPEED*. This is the standard MMSPEED protocol without aggregation in either the application layer or the MAC layer. MMSPEED provides QoS guarantees in both the timeliness domain and the reliability domains using localized algorithms for sensor networks.
- (ii) *MMSPEED-AGG*. This is the MMSPEED protocol with opportunistic aggregation. When a packet arrives at a relay node, aggregation of the data packets is performed at both the application layer as well as the MAC layer with packets that are already present in the cache waiting for transmission. However, no effort is made to wait at a relay node for aggregation with packets from upstream nodes.
- (iii) *RTPAW*. RTPAW [35] is a real-time power-aware framework and protocol stack. Nodes are grouped into clusters and a cluster head communicates with the sink node through relay nodes. An aggregation layer is present between MAC layer and the routing layers, and the cluster head uses the aggregation layer to collect the data from cluster nodes. Protocol overhead involves maintaining the clusters, election of the cluster head and the relay nodes and the periodic beacon messages. In our simulations, we have used a cluster size of 5 cluster nodes per cluster head.

In our experiments, we assume that wireless links are perfectly reliable as we would like to evaluate the timeliness and aggregation properties of SensiQoS with other protocols.

**5.1. Energy Model.** Each sensor node is assumed to have a radio range of 20 m. The bandwidth of the radio is assumed to be 20 Kbps. The sensor characteristics are given in Table 1. These values are taken from the specifications for the TR1000 radio from RF Monolithics [36].

**5.2. Simulation Environment.** The general simulation environment is drawn mainly from MMSPEED and is summarized in Table 2. Each sensor node that generates traffic for the sensor network maintains a CBR traffic of 5 packets/second throughout the simulation. The simulation

```

Procedure recvDataPacket (srcNode, SENSIQOS packet, nextHopVector)
Result: Schedule the packet for transmission
(1) currentClass ← getTOS (Packet) ;
(2) if isDestination then                                     /* if the packet is for self */
    /* Forward the packet to the application layer */;
(3) doSending (upPort, packet) ;
(4) else
(5)   body ← getBody (packet) ;
(6)   timestamp ← getTimestamp (packet) ;
(7)   progressDistance ← distance (sender, destination) - distance (self, destination) ;
(8)   timeS ← progressDistance/networkSpeed;
(9)   deadline ← getDeadline (packet) ;
(10)  qid ← getQueryID (packet) ;
    /* Retrieve packets that belong to the query */;
(11)  pkts ← lookupPacketsinCache (qid) ;
(12)  if pkts ≠ 0 then
(13)    cancel timeout for the packets;
    /* Aggregate packets using the aggregation function */;
(14)    Aggpacket ← doAggregate (pkts, p) ;
(15)    waitTime ← GetWaitTime (timeS, deadline, progressDistance) ;
(16)    if waitTime == 0 then
(17)      ForwardToDestination(packet);
(18)    else
(19)      Set timer for the waitTime;
      /* Store the packet in the node's memory */;
(20)      InsertPacketsinCache (Aggpacket)
(21)    end
(22) end
(23) end

```

ALGORITHM 1: Procedure recvDataPacket.

TABLE 1: Radio characteristics [36].

Radio mode	Power consumption (mW)
Transmit ( $T_x$ )	14.88
Receive ( $R_x$ )	12.50
Idle	12.36
Sleep	0.016

TABLE 2: Simulation environment parameters.

Bandwidth	30 Kbps
Terrian	200 × 200 m
Number of nodes	100
Node placement	Uniform
Radio range	30 m

is run for 500 seconds, and the results are taken from the average of 100 runs of the simulation and are shown with 95% confidence intervals. Confidence intervals are calculated using the method of independent replications [32] for these simulations unless otherwise mentioned. We obtain a single output variable in each of the simulation runs, and its distribution is not known. Hence we use statistical inference based on normal distribution (because of the Central Limit Theorem) to determine the confidence

intervals. The following formula is used to obtain the 100(1 -  $\alpha$ )% confidence interval of the population mean  $\mu$ :

$$\bar{\theta} - z_{1-\alpha/2} \frac{S_n}{\sqrt{n}} < \mu < \bar{\theta} + z_{1-\alpha/2} \frac{S_n}{\sqrt{n}} \quad (25)$$

where  $\bar{\theta}$  is the sample mean,  $S_n$  is the sample variance,  $n$  is the number of simulation runs, and  $z_{1-\alpha/2}$  is the (1 -  $\alpha/2$ ) quantile of the standard normal distribution  $N(0; 1)$ . The vertical lines on the graphs in this section refer to the confidence intervals.

Although SensiQoS relies on a localization scheme, we do not consider it in our simulator for simplicity. Instead, we make use of the geographic locations of sensor nodes provided by our simulator to determine the delay for each packet at a sensor node. Simulation results show that SensiQoS not only performs well compared to MMSPEED in providing real-time guarantees but also saves significantly more energy and extends the network lifetime. SensiQoS even outperforms MMSPEED with in-network aggregation in energy savings.

**5.3. Service Differentiation.** To demonstrate the service differentiation provided by SensiQoS, two events,  $E_1$  and  $E_2$ , are created in the network. This simulation consists of nodes sending data packets that belong to two different events with two different deadlines. Low Priority Event (LPE) has

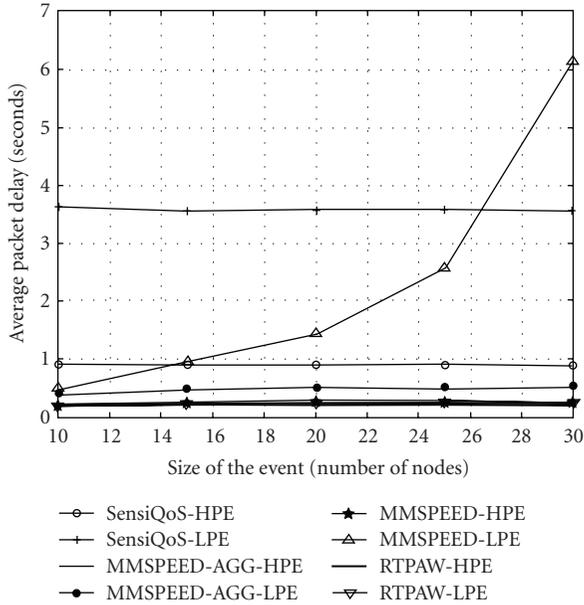


FIGURE 5: Service differentiation.

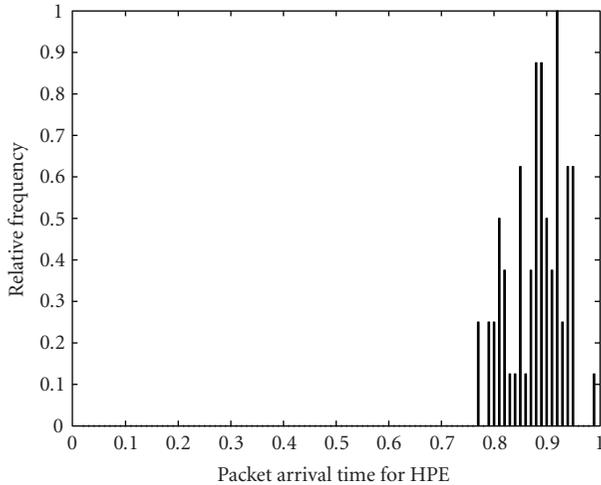


FIGURE 6: Normalized histogram of the packet arrival times for the high-priority event.

a deadline of 4.0 seconds while High Priority Event (HPE) has a deadline of 1.0 second [6]. The sink or the base station is located at one corner of the network. Nodes generate data from the other end of the network within an event radius. The results are shown in Figure 5. While both protocols provide service differentiation, SensiQoS provides a much greater differentiation for the two events. The reason for this is that SensiQoS allows packets at relay nodes to remain in the network for a longer time by determining the wait-time based on the real-time deadline of the packet. This improves the chances of a packet getting aggregated with packets of the same event. A histogram of the packet arrival times for the high-priority event is shown in Figure 6.

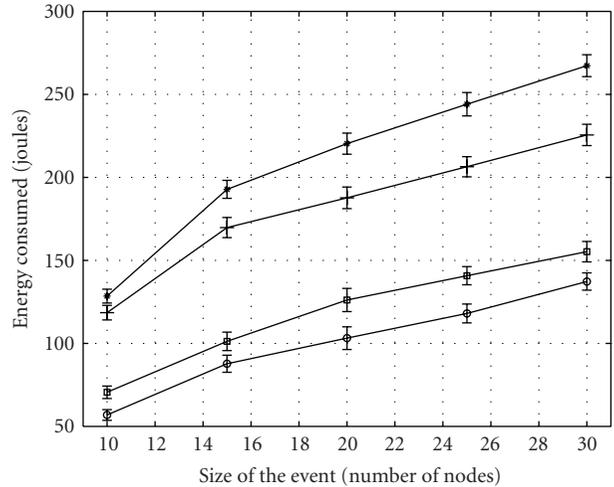


FIGURE 7: Energy consumption.

**5.4. Energy Savings.** Figure 7 shows the total energy consumption for each protocol. Total energy consumed for all the protocols is directly proportional to the number of transmissions, which is the sum of the number of data packets sent and the number of control packets sent per node. MMSPEED protocol without any aggregation consumes the most energy as expected. SensiQoS consumes the least amount of energy as SensiQoS aggregates more number of packets compared with the other two protocols and results in fewer number of transmissions. In fact, SensiQoS consumes 50 percent less energy than MMSPEED protocol with opportunistic aggregation and 70 percent less energy compared with the MMSPEED protocol without any aggregation.

**5.5. Packet Deadline Miss Ratio.** The deadline miss ratio is an important metric in soft real-time systems. In the simulation, some packets are lost due to congestion or forced drops. We also consider this situation as a deadline miss. The packet deadline miss ratio for each protocol is shown in Figure 8 with 95% confidence intervals. Packet deadline miss is a Bernoulli distributed random variable. Hence the packet deadline miss ratio is binomially distributed, and the confidence interval can be determined using the following relationship:

$$k_0(p) < s_n < k_1(p), \quad (26)$$

where  $k_0$  is the largest integer such that

$$\sum_{k=0}^{k_0} b(k; n, p) = B(k_0; n, p) \leq \frac{\alpha}{2} \quad (27)$$

and  $k_1$  is the smallest integer such that

$$\sum_{k=k_1}^n b(k; n, p) = 1 - B(k_1 - 1; n, p) \leq \frac{\alpha}{2}. \quad (28)$$

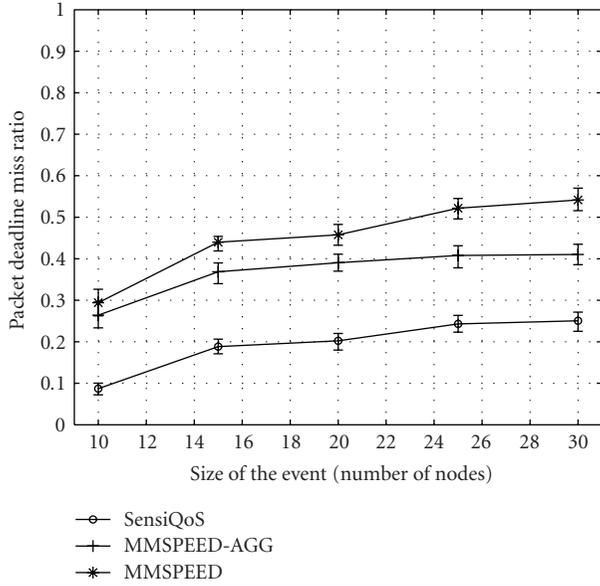


FIGURE 8: Packet deadline miss ratio with increasing number of flows.

The calculation of confidence intervals is done using the SEMSTAT software package [37].

As the number of flows increases in the network, MMSPEED drops more packets to meet the real-time deadline for the remaining packets in the network. However, SensiQoS drops far fewer packets due to its ability to aggregate better, which results in reduction in network contention, as shown in Figure 8.

**5.6. Node Density.** We now consider the effect of node density on SensiQoS. In this experiment, we increase the number of nodes present in the sensor grid. As the event size remains the same, the number of nodes that report the occurrence of the event increases with increase in node density. Correlation among data packets is likely to increase with increase in node density as the internode separation distance reduces and nodes get closer to each other.

Figure 9 compares the energy savings of SensiQoS with MMSPEED protocols with increasing node density. We show the average delay for the packets delivered to the sink as a function of the node density in Figure 10 with 95% confidence intervals. The packet delay in the same simulation run is correlated. Hence we use the method of batch means [32] to determine the confidence intervals. An initialization period of 5 seconds is used and a batch length of 50 seconds is used to obtain the confidence intervals. As the node density increases, average delay of the packets delivered by MMSPEED increases. This is due to contention in the network and the increased delay in acquiring the channel. This does not increase the aggregation as much as seen by the energy savings in Figure 9. SensiQoS enables the packets to remain in the network despite increased contention. The ability to wait in the network and aggregate enables SensiQoS to achieve greater energy savings. This is shown in Figure 10

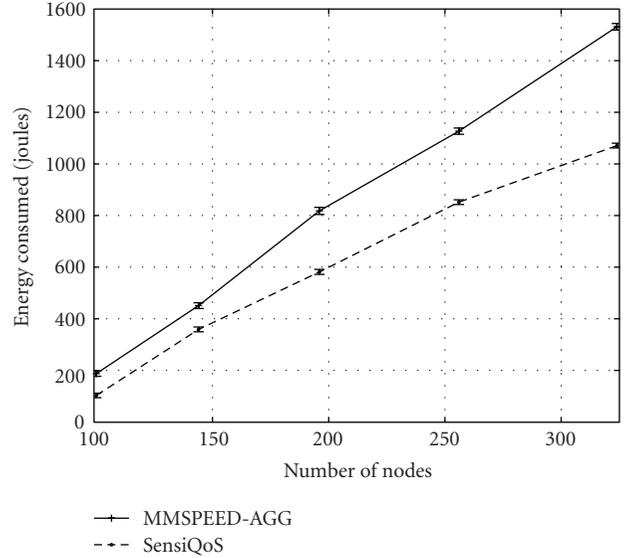


FIGURE 9: Energy consumed with increasing node density.

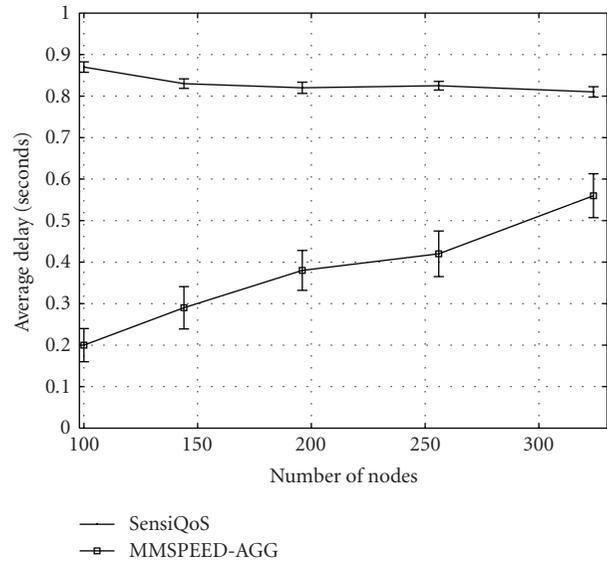


FIGURE 10: Average delay with increasing node density.

that as the number of nodes in the sensor network increases, the average delay of the packets delivered by SensiQoS remains approximately the same.

**5.7. Impact of Aggregation Factor.** So far we have assumed that the data has perfect correlation, and hence the resulting number of bytes after aggregating two different packets is a data packet of single-packet size. However, this can vary based on the aggregation factor of the corresponding aggregation method chosen by the application. We define the aggregation factor as the ratio of total number of bytes generated after aggregation over the size of a single packet when two packets are aggregated. In our simulations the size of a single packet is 30 bytes. As an example, with

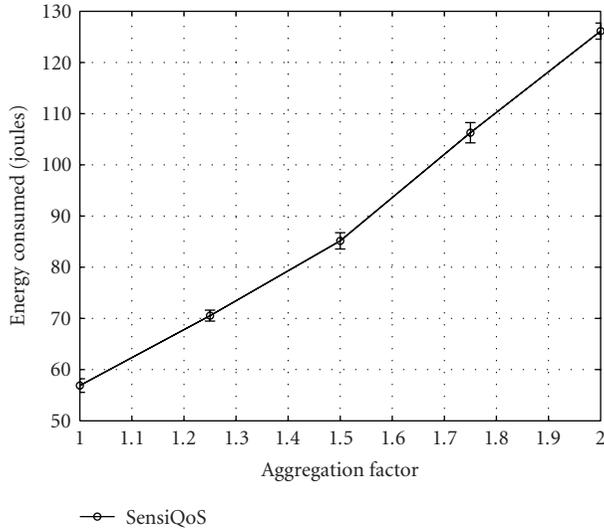


FIGURE 11: Energy consumed versus aggregation factor.

an aggregation factor of 1.2, the result of aggregating two packets is 36 bytes. We do not send fractional packets in our simulations. When enough fractional packets accumulate within the wait-time that can be combined into a single packet, we transmit the packet to the downstream node. The energy consumed as a function of the aggregation factor is shown in Figure 11.

**5.8. Impact of Event Occurrence.** So far, we have examined the effect of uniform event generation in the sensor network. We now investigate the effect of nonuniform event generation on the energy savings of SensiQoS. In this simulation, events are generated at random locations in the sensor network. The event radius is chosen as 50 m, and events occur at randomly chosen locations in the sensor network every 5 minutes. Each event lasts for 5 seconds at a Poisson arrival rate of 5 packets/second. When an event occurs at a specific location, all the sensor nodes within the range of the event radius generate data related to that event. The energy consumed as a function of the event frequency is shown in Figure 12. For each event frequency, one data packet of each priority is generated by the source sensor nodes. The simulations are run for one hour, and the results shown are the average energy consumed over five runs.

## 6. Conclusions

In this paper, we have presented SensiQoS, a distributed packet scheduling scheme for wireless sensor networks that reduces energy consumption without significantly increasing the number of packets that miss their real-time deadlines. SensiQoS adaptively varies the wait-time of each packet and transmits the packets in the order of their priorities. Each packet is forwarded to the node that can carry the packet towards the destination node with maximum speed using geographic routing. A next step in this research is to use optimistic approaches for calculating the wait-time of a

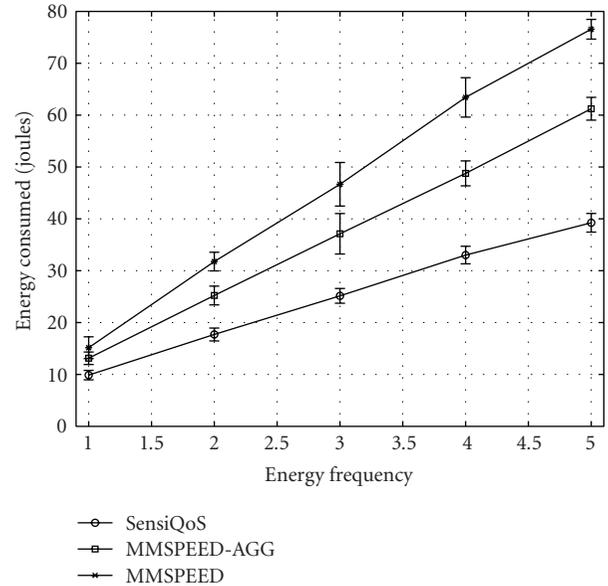


FIGURE 12: Energy consumed versus event frequency.

packet based on  $\text{rem\_time}(p)$  to achieve even more energy savings in dense sensor networks.

## Acknowledgments

The authors thank Professor John Board and Dr. John Zachary for helpful comments on an earlier draft of this paper. They are indebted to Professor Kishor Trivedi, who helped improve the results through the use of statistical inference methods.

## References

- [1] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, D. Estrin, and L. Girod, "Habitat monitoring: application driver for wireless communications technology," in *Proceedings of the ACM Workshop on Data Communication in Latin America and the Caribbean*, 2001.
- [2] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.
- [3] J. Warrior, "Smart sensor networks of the future," *Sensors Magazine*, March 1997.
- [4] I. Mahadevan and K. M. Sivalingam, "Quality of service architectures for wireless networks: IntServ and DiffServ models," in *Proceedings of the 4th International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN '99)*, pp. 420–425, 1999.
- [5] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems*, pp. 46–55, May 2003.
- [6] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: multipath Multi-SPEED Protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–753, 2006.

- [7] W.-H. Liao, Y. Kao, and C.-M. Fan, "Data aggregation in wireless sensor networks using ant colony algorithm," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 387–401, 2008.
- [8] H. Chen, H. Mineno, and T. Mizuno, "Adaptive data aggregation scheme in clustered wireless sensor networks," *Computer Communications*, vol. 31, no. 15, pp. 3579–3585, 2008.
- [9] J. Gao, L. Guibas, N. Milosavljevic, and J. Hershberger, "Sparse data aggregation in sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 430–439, April 2007.
- [10] L. Jia, G. Noubir, R. Rajaraman, and R. Sundaram, "GIST: group-independent spanning tree for data aggregation in dense sensor networks," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems*, pp. 282–304, June 2006.
- [11] P. Von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC '04)*, pp. 60–66, October 2004.
- [12] Y. Zhu, K. Sundaresan, and R. Sivakumar, "Practical limits on achievable energy improvements and useable delay tolerance in correlation aware data gathering in wireless sensor networks," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 328–339, September 2005.
- [13] W. Yuan, S. V. Krishnamurthy, and S. K. Tripathi, "Synchronization of multiple levels of data fusion in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 1, pp. 221–225, December 2003.
- [14] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: efficient aggregation for sensor networks," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 69–76, April 2006.
- [15] Y. Hu, N. Yu, and X. Jia, "Energy efficient real-time data aggregation in wireless sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '06)*, pp. 803–808, July 2006.
- [16] K. Akkaya and M. Younis, "An energy-aware qos routing protocol for wireless sensor networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCSW '03)*, pp. 710–715, 2003.
- [17] R. Benkoczi, H. Hassanein, S. Akl, and S. Tai, "QoS for data relaying in hierarchical wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet '05)*, pp. 47–54, October 2005.
- [18] N. Ouferhat and A. Mellouck, "QoS dynamic routing for wireless sensor networks," in *Proceedings of the 2nd ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet '06)*, pp. 45–50, October 2006.
- [19] X. Huang and Y. Fang, "Multi-constrained soft-QoS provisioning in wireless sensor networks," in *Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine '06)*, vol. 191, p. 14, 2006.
- [20] N. Saxena, A. Roy, and J. Shin, "Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks," *Computer Networks*, vol. 52, no. 13, pp. 2532–2542, 2008.
- [21] X. Huang and Y. Fang, "Multiconstrained QoS multipath routing in wireless sensor networks," *Wireless Networks*, vol. 14, no. 4, pp. 465–478, 2008.
- [22] B. Hoffman-Wellenhof, H. Lichteneger, and J. Collins, *Global Positioning System: Theory and Practice*, Springer, New York, NY, USA, 4th edition, 1997.
- [23] U. Bischoff, M. Strohbach, M. Hazas, and G. Kortuem, "Constraint-based distance estimation in ad-hoc wireless sensor networks," in *Proceedings of the 3rd European Workshop on Wireless Sensor Networks (EWSN '06)*, pp. 54–68, 2006.
- [24] B. Kusy, A. Ledeczki, M. Maroti, and L. Meertens, "Node-density independent localization," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 441–448, ACM, Nashville, Tenn, USA, April 2006.
- [25] N. Patwari and R. J. O'Dea, "Relative location in wireless networks," in *Proceedings of the 53rd IEEE Vehicular Technology Conference (VTC '91)*, pp. 1149–1153, 1991.
- [26] K. Whitehouse and D. Culler, "Calibration as parameter estimation in sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 59–67, September 2002.
- [27] L. Girod and D. Estrin, "Robust range estimation for localization in ad hoc sensor networks," Tech. Rep. CS-TR-2000XX, UCLA, 2000.
- [28] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 166–179, July 2001.
- [29] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *Proceedings of the International Conference on Network Protocols (ICNP '01)*, pp. 35–41, November 2001.
- [30] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke, "A high-accuracy, low-cost localization system for wireless sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 13–26, 2005.
- [31] "IEEE standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 11: wireless lan medium access control (mac) and physical layer (phy) specifications amendment 8: medium access control (mac) quality of service enhancements," IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)), pp. 1–189, 2005.
- [32] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, John Wiley & Sons, Chichester, UK, 2002.
- [33] A. Lachenmann, P. J. Marrón, M. Gauger, D. Minder, O. Saukh, and K. Rothermel, "Removing the memory limitations of sensor networks with flash-based virtual memory," *SIGOPS Operating Systems Review*, vol. 41, no. 3, pp. 131–144, 2007.
- [34] A. Sobeih, J. C. Hou, L.-C. Kung et al., "J-Sim: a simulation and emulation environment for wireless sensor networks," *IEEE Wireless Communications*, vol. 13, no. 4, pp. 104–119, 2006.
- [35] E. Toscano, O. Mirabella, and L. L. Bello, "An energy-efficient real-time communication framework for wireless sensor networks," in *Proceedings of the 6th International Workshop on Real-Time Networks (RTN '07)*, pp. 60–69, 2007.

- [36] RF Monolithics Inc, "Ash transceiver's designers guide," 2003, <http://www.rfm.com/>.
- [37] International SEMATECH Inc, "Semstat statistical software package," <http://www.itl.nist.gov/div898/handbook/dataplot.html>.