Energy-Aware QoS Provisioning for Wireless Sensor Networks: Analysis and Protocol

Muhammad Mahbub Alam, Md. Abdur Razzaque, Md. Mamun-Or-Rashid, and Choong Seon Hong

Abstract: **Wireless sensor networks (WSNs) are envisioned to facilitate information gathering for various applications and depending on the application types they may require certain quality of service (QoS) guarantee for successful and guaranteed event perception. Therefore, QoS in WSNs is an important issue and two most important parameters that hinder the goal of guaranteed event perception are time-sensitive and reliable delivery of gathered information, while a minimum energy consumption is desired. In this paper, we propose an energy-aware, multi-constrained and multipath QoS provisioning mechanism for WSNs based on optimization approach. Hence, a detailed analytical analysis of reliability, delay and energy consumption is presented to formulate the optimization problem in an analytical way. A greedy algorithm is proposed to achieve the desired QoS guarantee while keeping the energy consumption minimum. Also, a simple but efficient retransmission mechanism is proposed to enhance the reliability further, while keeping the delay within delay bound. Simulation results demonstrate the effectiveness of our scheme.**

Index Terms: **Delay guarantee, energy-awareness, performance analysis, quality of service (QoS) provisioning, reliability, wireless sensor networks (WSN).**

I. INTRODUCTION

Wireless sensor networks are envisioned to serve a variety of applications including homeland security, environmental monitoring, human imaging and tracking, biomedical research, and military surveillance. These networks are composed of autonomous tiny sensor nodes, which are able to measure physical parameters in a hostile environment to accomplish the task of a specific application. However, applications such as real-time target tracking, industrial process control, and time critical monitoring require certain level of quality of service (QoS) for the successful event perception and collaborative actions. Also, sensor networks are extremely energy constrained; so, efficient use of energy is one of the prime concerns in each protocol stack.

A. Quality of Service in WSNs

Wireless sensor networks are different from traditional data networks with specific application requirements, and the traffic may have different demands depending on the application types. For example, traditional network is usually expected to be reliable; whereas, due to the vulnerable wireless environment, reliability plays an important role in wireless sensor network (WSN). Since the sensor nodes usually sense co-related data due to their dense deployment, the reliability in WSNs might not be important for individual packet; rather it might collectively depend on all packets (or a subset) sent by the sources [1]. Further, the bandwidth requirement for a group of nodes in a certain period might be important in WSNs instead of that for individual nodes [1]. Therefore, QoS in WSNs depends on the specific requirements of the applications. For some applications, the path delay might not critically be important when data arrive at the sink with a certain delivery ratio. However, a multimedia application, where video or image data require certain delay guarantee, might accept packet loss up to a certain level. Moreover, a particular type of application may require both the reliability and delay guarantee and pose multiple constraints on routing. Finding a route with multiple constraints is an NP-complete problem [2], and thus, it is a reasonable way to solve it using a heuristic or approximation technique [3]. The solution for WSN also needs to be energy-aware in order to extend the network lifetime.

B. Problem Description

We investigate both reliability and delay guarantee in sensor networks. The reliability in sensor networks is defined as the ratio of the number of unique packets received by the sink(s) to the number of unique packets sent by the sources [3], [4]. Whereas, the delay (more specifically, end-to-end delay) associated with a packet is the time that it takes to reach the sink from the source.

Suppose, the reliability and delay requirements associated with a packet is R_{req} and D_{req}, respectively. Then, a path can meet the delay requirement, if it has end-to-end delay less than or equal to D_{req}. If the constraint is so aggressive that no path meets the delay bound, QoS requirement of the packet is not supported by the network. On the other hand, if no path meets the reliability requirement, the reliability of the network can be improved to provide the required reliability. In wireless networks, typically MAC layer retransmission is adopted to improve the reliability. However, retransmission increases the delay at each hop, and consequently, the end-to-end delay of a packet increases. Thus, retransmission based reliability improvement might affect the other metric. Moreover, the MAC layer retransmission does not effectively increase the reliability in sensor networks (especially in loaded network), since the retransmitted packets increase the medium contention [5]. It also decreases the transmission efficiency and consumes energy. So, it is required to handle the reliability and delay requirements separately.

In this paper, we focus on *multi-path routing* [6] to achieve the required reliability, where a node sends copies of a packet over

---

Manuscript received February 14, 2008; approved for publication by Felipe A. Cruz-Perez, Division II Editor, March 27, 2009.

This work was partially supported by the KOSEF grant funded by MEST (No. R01-2008-000-20801-0) and the MKE, Korea, under the ITRC support program supervised by the IITA (IITA-2009-(C1090-0902-0002)). Dr. C. S. Hong is corresponding author.

The authors are with the Computer Engineering Department of Kyung Hee University, South Korea, email:{mahbub, razzaque, mamun}@networking.khu.ac.kr, cshong@khu.ac.kr.

multiple paths. If the sinks receive at least one copy of a packet, the packet is assumed to be delivered successfully. However, to meet both reliability and delay requirements using multi-path routing, every selected path should satisfy the delay bound; so that, whichever copy is received by the sinks, delay requirement is met. Therefore, multi-path routing needs to select one or more paths to achieve the QoS, where each path satisfies the delay requirement individually, but all the paths provide the reliability collectively.

Existing proposals (for example, [3], [7]) select multiple paths to provide hop-by-hop QoS, where the forwarding node at each hop determines the one-hop QoS requirements and selects one or more next hops (paths) to satisfy the requirements. Thus, a forwarding node does not consider whether the next hop can provide the required QoS or not. In the worst case, to meet the reliability, a downstream node might create many copies of a packet. On the other hand, a downstream node might not find a delay satisfying path while forwarding a packet, as the queueing delay is expected to be higher near the sink. Furthermore, hop-by-hop QoS-aware path selection is vulnerable to engender routing loops [3]. To overcome the problems of the hop-by-hop QoS, our proposed mechanism selects paths to provide end-to-end QoS. In end-to-end QoS, a single node selects path(s) for a packet based on end-to-end delay and reliability of the available paths, and thus, forwards a packet only if it can meet the requirements. Though, there exist proposals in the literature for providing end-to-end QoS in wired, wireless, or even sensor networks; to the best of our knowledge, none of these mechanisms provides end-to-end QoS for multiple constraints with minimum energy consumption for WSNs.

Therefore, the path selection mechanism requires exact information about the path quality (i.e., reliability and delay) to select the desired paths. However, it is almost impossible to measure the exact path quality for wireless networks, since the network status changes very frequently. Furthermore, to keep updated information at each forwarding node, nodes require to periodically exchange information. This is very infeasible for resource constrained sensor networks. Therefore, providing hard QoS is not feasible for sensor networks and only soft QoS provisioning is achievable. Our aim is to provide soft QoS; where QoS is guaranteed with certain probability.

Energy is a very scarce resource for sensor networks. A path selection mechanism needs to select the subset of paths those consume minimum energy but satisfy both reliability and delay requirements. We use an optimization approach for energy-efficient path selection. The energy consumption in a path depends on the number of transmission attempts experienced by a packet in the path. Thus, the aggregate energy consumption for a packet in multi-path routing is proportional to the number of total transmission attempts experienced by all the copies of a packet. This implies that two subsets of paths, having same number of paths in each subset, might have different energy consumption. As a result, only minimizing number of paths might not ensure minimum energy consumption. Therefore, the objective of the optimization problem is to select the subset of paths that requires minimum expected transmission attempts, and thus, consumes minimum energy.

Finally, the unique nature of sensor networks, where destination of all packets is the sink node, might create problems for multi-path routing in single-sink WSNs. Since multi-path routing increases the network load and multiple paths converge near the sink, both the contention and congestion are increased. Therefore, we propose multi-sink multi-path routing, where multiple paths are spatially separated from each other. Note that this will consume energy in a spatially-balanced way and extend the network lifetime.

C. *Contributions of the Paper*

The key contributions of the paper are as follows: i) We propose a detailed analytical analysis of reliability, delay, and energy consumption for both singl-path and multi-path routing, ii) an energy-aware QoS provisioning mechanism is proposed using optimization approach and a greedy algorithm is presented, iii) to further enhance the QoS, we propose a simple and efficient retransmission method, which utilizes packet classification and priority queueing to improve the reliability, while keeping the end-to-end delay within delay bound, and iv) simulation results of the proposed mechanisms are presented.

The paper is organized as follows. Section II describes the system model and assumptions. In Section III, we present the proposed analytical formulation. The proposed QoS provisioning scheme is presented in Section IV. Performances of the proposed scheme are explained in Section V. Section VI describes the related works and we conclude in Section VII.

II. SYSTEM MODEL AND ASSUMPTIONS

A. *Cluster-Based Hierarchical Network*

We consider a cluster-based hierarchical network, where $\mathcal{N}$ homogeneous sensors are evenly distributed over the area of interest. Sensors are static and send the sensed data to the cluster-head (CH). The CHs send the data to the sink(s) using multi-hop routing. We assume that clusters are already formed using any existing method (for example, [8]–[10]). However, our QoS provisioning mechanism does not depend on clustering or the route finding mechanism. In fact, it can work with a general purpose routing protocol, if the routing protocol can find multiple paths for each source node. We choose cluster based network to utilize the inherent benefits of the hierarchical environment. Since, only few nodes perform the task of routing in cluster-based network, overhead of network state maintenance and route selection is expected to be minimum. The set of CHs are represented by C, and the ith CH is represented by C_i, where $i = 1, 2, \cdots, |C|$. We assume that all sensor nodes use CSMA/CA protocol in the MAC layer.

B. *Multi-Sink Multi-Path Routing*

As mentioned, multi-path routing in a single-sink WSN introduces some unique problems. Since the destination of all data is the sink node, data packets in WSNs converge near the sink. The extra load (i.e., the redundant packets) of multi-path routing makes this problem further complicated. To study the impact of multi-path routing in a single-sink WSN, we run simulations where data are generated in a cluster of 10 sensor nodes. All nodes send their data to the CH, and the CH sends the data to the

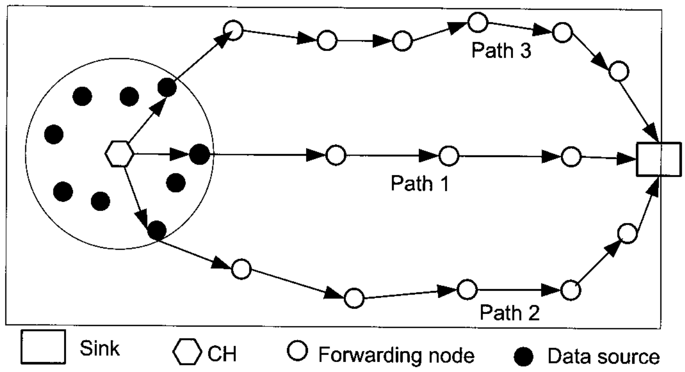


Fig. 1. Multi-path routing in single-sink WSN, paths converge near the sink.

sink using single-path or multi-path routing as shown in Fig. 1. The general simulation setup and parameters are explained in Section V. We consider three paths from the CH to the sink, where path 1 individually interferes with paths 2 and 3, but paths 2 and 3 do not interfere with each other. The results are summarized in Table 1. The simulation results clearly show that if only one sink is used, then multi-path routing does not increase the delivery ratio effectively. If the paths interfere with each other, the delivery ratio might decrease due to increased packet dropping.

The reason behind the packet dropping is the extra load (i.e., the redundant packets) of the multi-path routing, which increases both the contention and congestion near the sink. Furthermore, nodes near the sink run out of energy quickly; which in turn, reduces the network lifetime. Therefore, to overcome both the problems (i.e., to handle the extra load and to extend the network lifetime), we use multiple sinks in the network, which increases the network capacity and spatially balances the energy consumption. We assume that sinks are well-separated from each other, so that multiple copies of a packet do not interfere with each other. Many proposals exist in the literature, where multiple sinks are used to overcome the *hot spot* problem. For example, Siphon [11] proposes multiple virtual sinks in the network to avoid congestion near the sink and DEED [12] dynamically adds sinks in the network to reduce the collision and decrease the delay. We use multiple sinks to avoid the increased contention and congestion due to packet redundancy. Note that extra resources of multiple sinks facilitate to achieve the required QoS of the application, which is not achievable with single sink. Like [13], we assume that sinks are connected to a central data collection center. So, it does not matter to which sink, a particular data packet is delivered. The data collection center can uniquely identify and receive every delivered packet from the sinks.

C. Routing Assumptions

We assume that each cluster head knows the set of available paths to all the sinks. Also, the next hop node and hop count of each path are known to the cluster heads. Usually, these information can be extracted from the clustering algorithm very easily. Multi-path routing decision is made by the source cluster-head, where we define a CH as source cluster-head (SCH) for all of its member sensors. Intermediate nodes in the path, just forward the data packets.

Table 1. Delivery ratio for single-sink multi-path routing without MAC layer retransmission.

Routing	Path(s)	No of hops	Delivery ratio
Single-path	Path 1	5	74%
	Path 2	7	61%
	Path 3	8	57%
Multi-path	Path 1 & 2	–	63%
	Path 1 & 3	–	58.5%
	Path 2 & 3	–	69.5%
	Path 1, 2 & 3	–	54%

III. PROPOSED ANALYTICAL APPROACH

As mentioned, our proposed mechanism provides energy-efficient QoS provisioning based on optimization. Therefore, in this section, we address the probabilistic formulation of reliability, delay, and energy consumption for both single-path and multi-path routing. The results of the analyses are presented to show the effectiveness of multi-sink multi-path routing. Further, based on these analyses and results, we formulate the optimization problem in Section IV. We assume that multi-path routing does not adopt any MAC layer retransmission; whereas, single-path routing uses MAC layer retransmission.

A. Reliability Constraint Analysis

If the number of unique packets sent by the sources is N_s and the number of unique packets received by the sinks is N_r, the reliability, denoted as R, is $R = N_r/N_s$.

The reliability of a path is a multiplicative metric. Thus, if the probability that a packet is dropped at the jth hop of a h-hop path is p_j^c, the probability that the packet is received by the sink is

$$p(h) = \prod_{j=1}^{h}(1 - p_j^c) \tag{1}$$

where $p(h)$ is the probability of success for a path of h hops.

A.1 Reliability of Multi-path Routing

In multi-path routing, if there are L paths and the hop count of the ith path is h_i, the probability that at least one copy of a packet is successfully received by the sinks is

$$p(L) = 1-\prod_{i=1}^{L}[1-p_i(h_i)] = 1-\prod_{i=1}^{L}\left[1 - \prod_{j=1}^{h_i}(1 - p_{i,j}^c)\right] \tag{2}$$

where $p(L)$ is the probability of success for multi-path routing with L paths, $p_i(h_i)$ is the probability of success for the ith path defined in (1) and $p_{i,j}^c$ is the probability that a packet is dropped at the jth hop of the ith path.

Since a packet is received with probability $p(L)$, thus the number of total packets received by the sink has binomial distribution, which is given by

$$P[N_r = n] = \binom{N_s}{n}[p(L)]^n[1 - p(L)]^{N_s-n}.$$

Suppose, the required reliability, R_{req}, is achieved when the number of unique packets received by the sink is N_{req}, then $N_{\text{req}} = R_{\text{req}} N_s$. And to achieve this, N_r must be greater than or at least equal to N_{req}. Then, the probability that the reliability is met in multi-path routing (i.e., $\text{P}[N_r \geq N_{\text{req}}]$), $p_{\text{req}}^{\text{multi}}$, is

$$p_{\text{req}}^{\text{multi}} = \sum_{n=N_{\text{req}}}^{N_s} \binom{N_s}{n} [p(L)]^n [1 - p(L)]^{N_s - n}. \tag{3}$$

A.2 Reliability of Single-Path Routing

In single-path routing, at each hop, a packet is retransmitted to increase the reliability and a packet is dropped after M transmission attempts (i.e., initial transmission and $M - 1$ retransmissions), where M is the maximum retransmission limit. Therefore, the probability, denoted as $p_j(M)$, that the jth hop successfully forwards a packet within M transmission attempts is given by

$$p_j(M) = 1 - (p_j^c)^M.$$

And the probability that a packet is successfully received by the sink in single-path routing with hop count h, using (1), is

$$p(h) = \prod_{j=1}^{h} [p_j(M)] = \prod_{j=1}^{h} \left[1 - (p_j^c)^M\right].$$

Finally, similar to (3), the probability that reliability is met in single-path routing with retransmission limit M, $p_{\text{req}}^{\text{single}}$, is

$$p_{\text{req}}^{\text{single}} = \sum_{n=N_{\text{req}}}^{N_s} \binom{N_s}{n} [p(h)]^n [1 - p(h)]^{N_s - n}.$$

B. Delay Constraint Analysis

The total delay, denoted as d, experienced by a packet in a path of hop count h is the sum of the delays at the intermediate nodes, d_j (where $j = 1, 2, \cdots, h$), and is given by

$$d = \sum_{j=1}^{h} d_j. \tag{4}$$

Based on the assumptions that the propagation and processing delays are negligible, d_j can be calculated as follows

$$d_j = d_{\text{trans}} + d_{\text{MAC}} + d_{\text{que}}$$

where d_{trans} is the transmission delay, d_{MAC} is the medium access delay and d_{que} is the queueing delay of a packet.

B.1 Transmission Delay (d_{trans})

Transmission delay of a packet depends on the packet size (including the PHY and MAC overhead [14]), P_{size}, and transmission rate or link bandwidth, BW. Assuming all packets have same size and all sensors are homogeneous having same radio, d_{trans} is same for all packets and is given by

$$d_{\text{trans}} = \frac{P_{\text{size}}}{\text{BW}} + d_{\text{SIFS}} + d_{\text{ACK}}$$

where d_{SIFS} and d_{ACK} are the duration of the short interframe space (SIFS) [14] and the ACK transmission time [14], respectively.

B.2 Medium Access Delay (d_{MAC})

When a node starts to transmit a packet, it needs to go through a backoff process as part of the CSMA/CA medium access procedure [14]. For a single transmission attempt, MAC delay is the duration of the backoff process and is also known as backoff delay (BO). However, if retransmission is adopted, then MAC delay is the sum of the backoff delays of each transmission attempt and the transmission delays of each unsuccessful attempt.

At the beginning of the backoff process of the mth transmission attempt (where $m = 0, 1, \cdots, M - 1$), a node uniformly selects a backoff value, B_m, in the range $(0 \sim W_m - 1)$. Here, W_m is the size of the *contention window* for the mth attempt and is given by $W_m = 2^m CW_{\min}$, where $CW_{\min}$ is the size of the *minimum contention window* [14].

During backoff, a node decrements the value of B_m by '1' in each free slot and freezes the backoff process when it hears any transmission from the neighbors. It resumes the decrement process, if it finds the medium free again for more than a DIFS period [14]. The node finally transmits when B_m reaches zero. Thus, the backoff delay in the mth attempt is the sum of the duration of B_m empty slots (i.e., generic slots) and zero or more busy periods; where a busy period indicates a successful (only one neighbor transmits) or an unsuccessful (more than one neighbor transmit) transmission by the neighbors.

Like [15], we consider the busy periods also as slots. Thus, a slot duration, denoted as t_{slot}, can have any one of three values, namely i) duration of an empty slot, σ, ii) duration of a successful transmission, T_s, and iii) duration of an unsuccessful transmission, T_c. Let p_t denote the probability that a slot is busy (i.e., there is a transmission by at least one neighbor); whereas, p_s and p_c denote the probability of successful and unsuccessful transmission in busy slot, respectively.

Therefore, the backoff process can be represented as a Bernoulli process [16], where in each trail the value of B_m is decremented by '1.' Further, the number of slots (both empty and busy) required in the kth trail, T_k, is geometrically distributed with parameter $(1 - p_t)$ [16]. Thus, the number of total slots required for B_m to reach zero, denoted as B'_m, is given by

$$B'_m = \sum_{k=1}^{B_m} T_k. \tag{5}$$

Since the T_k is i.i.d. random variable, we drop the subscript in T_k. We calculate the expected value of B'_m using Wald's equation [17] and the variance using [17]. Then,

$$\begin{aligned} E[B'_m] &= E[B_m]E[T], \\ Var[B'_m] &= E[B_m]Var[T] + Var[B_m](E[T])^2 \end{aligned} \tag{6}$$

where $E[B_m]$ and $Var[B_m]$ are the expected value and variance of B_m, respectively. As B_m is a uniform random variable, so, $E[B_m] = \frac{W_m - 1}{2}$ and $Var[B_m] = \frac{W_m^2 - 1}{12}$ [17]. Also, as T is geometrically distributed, the expected value and variance of T are $E[T] = \frac{1}{1 - p_t}$ and $Var[T] = \frac{p_t}{(1 - p_t)^2}$, respectively [17].

As B'_m is the number of slots (empty and/or busy) that a node needs to wait in the backoff period, the backoff delay in the mth attempt, denoted as BO_m, is the sum of the duration of B'_m

slots, i.e., $BO_m = \sum_{k=1}^{B'_m} t_{\text{slot}}$. Similar to (6), the expected value and variance of BO_m can be given by

$$E[BO_m] = E[B'_m]E[t_{\text{slot}}],$$
$$Var[BO_m] = E[B'_m]Var[t_{\text{slot}}] + Var[B'_m](E[t_{\text{slot}}])^2 \quad (7)$$

where $E[t_{\text{slot}}]$ and $Var[t_{\text{slot}}]$ are the expected value and variance of t_{slot}, respectively. Since, an empty slot with duration σ, successful slot with duration T_s and unsuccessful slot with duration T_c occur with probability $(1-p_t)$, p_tp_s, and p_tp_c, respectively; therefore, the expected value and variance of t_{slot} are [17]

$$E[t_{\text{slot}}] = (1-p_t)\sigma + p_tp_sT_s + p_tp_cT_c,$$
$$Var[t_{\text{slot}}] = (1-p_t)(\sigma - E[t_{\text{slot}}])^2 + p_tp_s(T_s - E[t_{\text{slot}}])^2 + p_tp_c(T_c - E[t_{\text{slot}}])^2.$$

Since multi-path routing achieves the required reliability using path diversity, therefore, we assume that no MAC layer retransmission is used. Thus, the MAC delay for multi-path routing, denoted as $d_{\text{MAC}}(\text{multi})$, is the backoff delay of the initial transmission attempt (i.e., when $m = 0$), which is $d_{\text{MAC}}(\text{multi}) = BO_0$. The expected value and variance of $d_{\text{MAC}}(\text{multi})$ can be found from (7), putting $m = 0$.

Whereas, for single-path routing, a packet is retransmitted until it is successfully delivered to the next hop or the number of retransmission reaches M. Thus, the number of retransmissions required for a successful delivery, denoted as M', has truncated geometric distribution and the probability mass function (PMF) is

$$P[M' = m] = \frac{p_tp_s(1-p_tp_s)^m}{1-(1-p_tp_s)^M}, \qquad m = 0, 1, \cdots, M-1.$$

Therefore, the MAC delay for single-path routing, d_{MAC}(single), becomes a function of M'. Let $d_{\text{ACC}}(M')$ denote the MAC delay of a packet that experiences M' retransmissions, then, $d_{\text{MAC}}(\text{single}) = d_{\text{ACC}}(M')$, where $M' = 0, 1, \cdots, M-1$. The delay $d_{\text{ACC}}(M')$ includes the backoff delay of the initial transmission attempt, BO_0, and M' unsuccessful transmissions each followed by a backoff delay BO_m (where $m = 1, 2, \cdots, M'$), and is given by

$$d_{\text{ACC}}(M') = \sum_{m=0}^{M'} BO_m + \sum_{m=1}^{M'} T_c.$$

The expected value and variance of $d_{\text{ACC}}(M')$ are given by

$$E[d_{\text{ACC}}(M')] = \sum_{m=0}^{M'} E[BO_m] + M'T_c,$$
$$Var[d_{\text{ACC}}(M')] = \sum_{m=0}^{M'} Var[BO_m].$$

Finally, the expected value and variance of $d_{\text{MAC}}(\text{single})$ for maximum retransmission limit M, are given by

$$E[d_{\text{MAC}}(\text{single})] = \frac{p_tp_s}{1-(1-p_tp_s)^M}\sum_{m=0}^{M-1}(1-p_tp_s)^m \cdot E[d_{\text{ACC}}(m)],$$
$$Var[d_{\text{MAC}}(\text{single})] = \frac{p_tp_s}{1-(1-p_tp_s)^M}\sum_{m=0}^{M-1}(1-p_tp_s)^m \cdot [Var[d_{\text{ACC}}(m)] + (E[d_{\text{ACC}}(m)] - E[d_{\text{ACC}}])^2].$$

B.3 Queueing Delay (d_{que})

Queueing delay at any node depends on the queue service time and the packet arrival pattern. For wireless sensor networks, the service time of the queue, s, is the sum of the transmission delay d_{trans} and medium access delay d_{MAC}. Therefore, the expected service time ($1/\mu$, where μ is the service rate) and its variance are given by

$$E[s] = E[d_{\text{MAC}}] + d_{\text{trans}} = 1/\mu, \quad (8a)$$
$$Var[s] = Var[d_{\text{MAC}}] = \sigma_s^2. \quad (8b)$$

Suppose, the cluster-head at the jth hop of a path, C_j has n_j sensors as cluster members and data generation of each sensor has Poisson distribution with rate g_j [18], [19]. The arrival rate at the distant cluster-head (having hop count H), that does not forward packets of other CHs, is $\lambda_H = n_Hg_H$. However, the intermediate CHs forward the data of the upstream CHs, so the arrival rate at the intermediate CHs also has the Poisson distribution due to the superposition property of the Poison process [16]. Let us denote U_j be the number of upstream CHs for C_j. Then, the arrival rate at C_j, where C_j is h hops away from the sink, is

$$\lambda_j = n_jg_j + \sum_{u=1}^{U_j}\sum_{k=h+1}^{H} n_kg_k \prod_{l=h+1}^{k}(1-p_b) \quad (9)$$

where p_b is the blocking probability of a packet. Now, the jth cluster-head can be modeled as $M/G/1$ queueing system with arrival rate, λ_j, expected service time, $E[s_j]$ and variance of service time, $\sigma_{s_j}^2$. The expected number of packets in any CH can be found by using Pollaczek-Khinchin [20] formula

$$E[q] = \rho + \frac{\rho^2 + \lambda^2\sigma_s^2}{2(1-\rho)}$$

where $\rho = \lambda/\mu$ is the traffic intensity at the CH. We calculate the expected sojourn time of a packet by applying Little's Law [20] from above equation. Note that sojourn time includes all the delays (i.e., d_{trans}, d_{MAC}, and d_{que}) of a packet in the CH [20]. Therefore, the expected delay in the jth hop is

$$E[d_j] = \frac{E[q_j]}{\lambda_j} = \frac{1}{\lambda_j}\left(\rho_j + \frac{\rho_j^2 + \lambda_j^2\sigma_{s_j}^2}{2(1-\rho_j)}\right).$$

Finally, the expected end-to-end delay of a packet for a path of h hops is measured as

$$E[d] = \sum_{j=1}^{h} E[d_j] = \sum_{j=1}^{h}\frac{1}{\lambda_j}\left(\rho_j + \frac{\rho_j^2 + \lambda_j^2\sigma_{s_j}^2}{2(1-\rho_j)}\right). \quad (10)$$

C. Energy Consumption Analysis

The expected energy consumption of a packet is the product of the expected number of transmission attempts and the energy consumption for each transmission attempt, which is $E[E_{\text{total}}] = E[T_{\text{total}}]E_{\text{hop}}$, where E_{total} and T_{total} are the total energy consumption and the total transmission attempts for a packet, respectively. E_{hop} is the energy consumption in one transmission attempt and is constant for fixed size packet.

C.1 Energy Model

We assume sensors are homogeneous, so each sensor has same transmission radius (r) and they consume equal energy to transmit a bit. We consider, for simplicity, a simple radio model where the radio dissipates E_{elec} energy per bit to run the transmitter or receiver and E_{amp} energy per bit for the transmit amplifier. We also assume r^2 energy loss due to channel transmission. Then, the energy consumption to send a k-bit message to next hop is, $E_{\text{hop}} = 2E_{\text{elec}}k + E_{\text{amp}}kr^2$.

C.2 Energy Consumption in Multi-path Routing

In multi-path routing, L paths are selected and L copies of a single packet are sent. Let $T_{\text{path}}(i)$ denote the number of transmission attempts for a packet in path i. Then, total expected transmission attempts, $E[T_{\text{total}}(m)]$, for multi-path routing is

$$E[T_{\text{total}}(m)] = \sum_{i=1}^{L} E[T_{\text{path}}(i)]. \tag{11}$$

Now, $T_{\text{path}}(i)$ depends on the hop count h_i of path i and the probability of packet dropping ($p^c_{i,j}$, where $j = 1, 2, \cdots, h_i$) at each hop in the path. Also, a packet can be either delivered to the sink and thus experienced h_i attempts or dropped in any intermediate hop and experienced less than or equal to h_i attempts. Thus, the PMF of $T_{\text{path}}(i)$ is given by

$$P[T_{\text{path}}(i) = t] = \begin{cases} p^c_{i,t} \prod_{j=1}^{t-1}(1 - p^c_{i,j}), & t = 1, 2, \cdots, h_i \\ \prod_{j=1}^{t}(1 - p^c_{i,j}), & t = h_i \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, the expected number of transmission attempts for a packet in path i is given by

$$E[T_{\text{path}}(i)] = \sum_{t=1}^{h_i} \left[t p^c_{i,t} \prod_{j=1}^{t-1}(1 - p^c_{i,j}) \right] + h_i \prod_{j=1}^{h_i}(1 - p^c_{i,j}). \tag{12}$$

Finally, combining (11) and (12), the expected total transmission attempts for a packet in multi-path routing is

$$E[T_{\text{total}}(m)] = \sum_{i=1}^{L} \left[\sum_{t=1}^{h_i} \left[t p^c_{i,t} \prod_{j=1}^{t-1}(1 - p^c_{i,j}) \right] + h_i \prod_{j=1}^{h_i}(1 - p^c_{i,j}) \right]. \tag{13}$$

Table 2. Average success rate per transmission attempt with different network loads (packets per second generated by the sensor nodes).

Routing	PPS = 2	PPS = 6
$L = 1, M = 1$	0.892	0.875
$L = 2, M = 1$	0.881	0.821
$L = 3, M = 1$	0.873	0.793
$L = 4, M = 1$	0.859	0.772
$L = 1, M = 2$	0.864	0.667
$L = 1, M = 3$	0.831	0.521
$L = 1, M = 4$	0.792	0.485

C.3 Energy Consumption in Single-Path Routing

In single-path routing, at each hop, a packet is either delivered to the next hop within M transmission attempts or dropped after M unsuccessful transmission attempts. Note that if a packet is dropped at any hop, it experienced exactly M transmission attempts at that hop.

Let $T_{\text{hop}}(j)$ denote the number of transmission attempts experienced by a packet at the jth hop, which is forwarded to the next hop within M transmission attempts. Then, $T_{\text{hop}}(j)$ has truncated geometric distribution with the following PMF

$$P[T_{\text{hop}}(j) = t] = \frac{(p^c_j)^{t-1}(1 - p^c_j)}{1 - (p^c_j)^M}, \quad t = 1, 2, \cdots, M.$$

And expected number of transmission attempts at the jth hop is

$$E[T_{\text{hop}}(j)] = \frac{1 - (p^c_j)^M[1 + (1 - p^c_j)M]}{(1 - p^c_j)[1 - (p^c_j)^M]}.$$

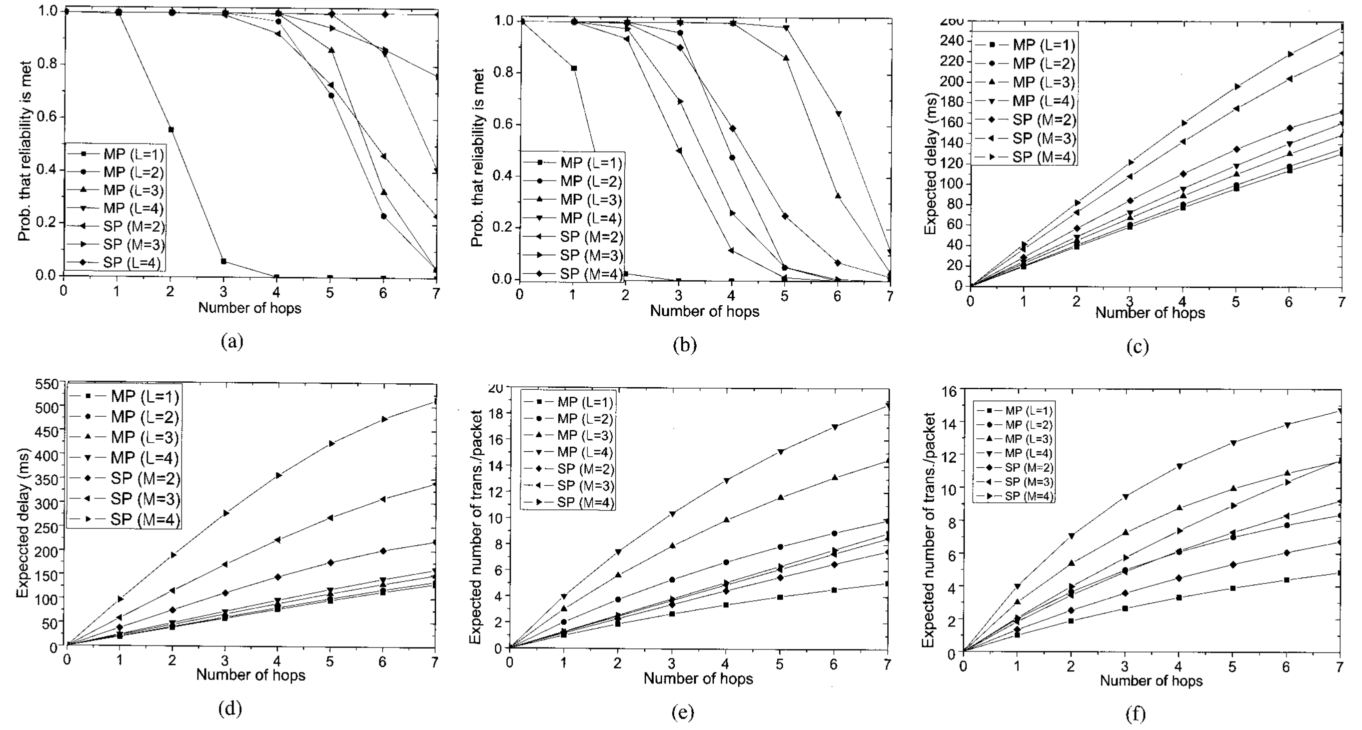
Suppose, at the jth hop, a packet is either dropped with probability p^d_j or delivered to the next hop with probability $(1 - p^d_j)$, where $p^d_j = (p^c_j)^M$. Let h' denote the number of hops traveled by a packet before being dropped at any intermediate node or delivered to the sink. Then, for a path of h hops, the PMF of h' is given by

$$P[h' = t] = \begin{cases} p^d_{t+1} \prod_{j=1}^{t}(1 - p^d_j), & t = 0, 1, \cdots, h - 1 \\ \prod_{j=1}^{t}(1 - p^d_j), & t = h \\ 0, & \text{otherwise.} \end{cases}$$

And total number of transmission attempts experienced by a packet is a function of h', that is $T_{\text{total}}(h')$ and is given by

$$T_{\text{total}}(h') = \begin{cases} \sum_{j=1}^{h'} T_{\text{hop}}(j) + M, & h' = 0, 1, \cdots, h - 1 \\ \sum_{j=1}^{h'} T_{\text{hop}}(j), & h' = h. \end{cases}$$

Finally, the expected number of total transmission attempts



(a) (b) (c)

(d) (e) (f)

Fig. 2. Results of the analysis for multi-path (MP) and single-path (SP) routing with different packet generation rates. Probability that the reliability requirement is met for (a) PPS $= 2$, (b) PPS $= 6$. Expected delay experienced by a packet for (c) PPS $= 2$ and (d) PPS $= 6$. Expected energy consumption for a packet for (e) PPS $= 2$ and (f) PPS $= 6$. L indicates the number of paths and M indicates the maximum retransmission limit.

for a packet in single-path routing, $T_{\text{total}}(s)$, is given by

$$E[T_{\text{total}}(s)] = \sum_{t=0}^{h-1}\left[\sum_{j=1}^{t} E[T_{\text{hop}}(j)] + M\right] \cdot p_{t+1}^{d}\prod_{j=1}^{t-1}(1-p_j^{d}) + \prod_{j=1}^{h}(1-p_j^{d})\sum_{j=1}^{h} E[T_{\text{hop}}(j)].$$

D. Results and Discussions

To get the results of the analysis, we need to know the probability of success of a single transmission attempt for different traffic conditions and different values of M and L. Since the detailed analysis of the probability of success is out of the scope of the paper, we have used long run average success rate of single transmission attempt found in simulation for the results and left the detailed analysis as a future work.

Table 2 shows the success rate found in simulation for a single transmission attempt in single-path and multi-path routing. The simulation results are taken for two different traffic rates; that is, the number of generated packets per second (PPS) by every sensor node in the network are 2 and 6. The data enlisted in Table 2 show one important property that the collision increases with retransmission. As a result, the success rate per transmission attempt decreases, even though overall success rate may increase. When the network is lightly loaded, the success rates with retransmission are higher than the rates when the network is highly loaded (network works in saturation). With lower source rates, usually the network operates below saturation and retransmission does not decrease the success rate heavily. In this case, the retransmitted packets utilize the unused capacity of the network. However, with high source rates, the network is already in saturation and retransmission increases the contention of the medium and success rate drops sharply. Fig. 2 shows the results of the analyses presented in the previous subsections.

Figs. 2(a) and 2(b) show the probability of meeting the reliability with PPS equal to 2 and 6, respectively. We assume that a delivery ratio of 85% will meet the reliability requirement. Note that according to (3), a probability of around 0.6 indicates that the expected number of packets received by the sinks just meets the reliability. Fig. 2(a) shows that the reliability of the sensor network is increased with retransmission. In this case, the network is lightly loaded and the overall success rate increases with retransmission. Also, the performance is better than multi-path routing when the hop count is high. Fig. 2(b) shows that in a loaded network, multi-sink multi-path routing outperforms single-path routing. The reason is that the retransmitted packets increase contention and overall success rate does not increase.

The expected end-to-end delay for a packet is shown in Figs. 2(c) and 2(d) with PPS equal to 2 and 6, respectively, where for multi-path routing we consider the delay of the longest path. As expected, multi-path routing exhibits less delay than single-path routing. Fig. 2(c) shows that the differences between delays are small for PPS equal to 2, since in a lightly loaded network success rate is high and usually few retransmissions are required. On the other hand, Fig. 2(d) shows that expected delay is much higher in single-path routing with retransmission and delay increases with the increasing value of M.

The expected energy consumption for a single packet is shown in Figs. 2(e) and 2(f). The figures show the expected number of transmission attempts for a packet, instead of the expected energy consumption. Note that the expected number of transmission attempts is proportional to the expected energy consumption of a packet. In multi-path routing, we assume that the hop counts of all paths are same. In reality, most of the cases hop counts will vary and so the number of transmission attempts will be less. Fig. 2(e) shows single-path routing with different retransmission limits require almost same energy consumption for a lightly loaded network. The reason is the almost same success rates of the lightly loaded network, which require almost equal number of retransmissions irrespective of the value of M. And as expected, multi-path routing consumes more energy than single-path routing. Since the success rate is high, a packet is supposed to travel longer distance and it consumes more energy. Fig. 2(f) shows the energy consumption for PPS equal to 6 and note that the difference in energy consumption per packet between multi-path and single-path routing is less here. The reason is that in a loaded network retransmission increases the number of collisions and almost at each hop a packet requires retransmissions. Furthermore, in reality average path length will be less in multi-path routing and the gap will be even smaller.

The results of the analysis indicate that the conventional retransmission mechanism is unable to meet both the QoS requirements, specially in a loaded network. Further, as mentioned in Section II-B, single-sink multi-path routing does not increase the reliability effectively. Whereas, multi-sink multi-path routing can meet both the QoS requirements, with reasonable energy consumption. These observations lead us to design the proposed QoS provisioning mechanism in Section IV.

However, multi-path routing should select the subset of paths, those consume minimum energy; since energy consumption varies with the number of paths and the paths in the subset. Therefore, our proposed mechanism selects the desired paths using optimization, where the goal of the optimization is to select the subset of paths, those collectively consume minimum energy subject to both reliability and delay constraints. We formulate the optimization problem in Section IV, where the objective function and the constraints are formulated based on the analysis presented in this section. Also, as shown in the results, the energy consumption in multi-path routing is proportional to the total hop counts of the selected paths, so, we propose a greedy algorithm in Section IV, which selects the paths having minimum total hop counts.

Finally, the reliability in both single-path and multi-path routing decreases for longer paths, therefore, the analysis motivates us to propose the controlled and effective retransmission mechanism (Section IV-F).

IV. PROPOSED QOS PROVISIONING MECHANISM

In this section, we propose the energy-aware QoS provisioning mechanism. We first formulate the optimization problem, then we propose a greedy algorithm to solve the optimization problem. Since the path selection mechanism requires the estimated reliability and delay of the available paths, we also propose the estimation methods for both reliability and delay. Finally, we propose an effective retransmission mechanism to further enhance the reliability.

A. Problem Formulation

Objective Function: The objective of the optimization is to select a subset of paths that consumes minimum energy. The energy consumption of a packet is proportional to the number of transmission attempts that all the copies of the packet experienced in the selected paths. In (13), we have shown the expected number of transmission attempts that all copies of a packet experienced in a subset of paths and the objective function minimizes this.

Constraints: The reliability constraint ensures that the selected subset of paths meets the desired reliability with certain probability. In (3), we have shown the probability of meeting the reliability by a subset of paths in multi-path routing. The reliability constraint enforces that the probability in (3) must be greater than or equal to P_{req}. On the other hand, the delay constraint ensures that the delay of every selected path is less than or equal to the delay requirement. In (10), we have shown the expected delay of a packet in one path. So, the delay constraint enforces that the expected delay in (10) must be less than or equal to D_{req}, for every selected path.

We assume that each cluster-head has L paths toward L different sinks,[1] where the hop count of the ith path is h_i. We define a binary variable, x_i, which equals 1, if the ith path is selected, otherwise 0. Then, the optimization problem is formulated as minimize

$$\sum_{i=1}^{L} x_i \left[\sum_{t=1}^{h_i} [t p^c_{i,t} \prod_{j=1}^{t-1} (1 - p^c_{i,j}) + h_i \prod_{j=1}^{h_i} (1 - p^c_{i,j}) \right] \qquad (14)$$

subject to

$$\sum_{n=N_{\text{req}}}^{N_s} \binom{N_s}{n} [p(L)]^n \, [1 - p(L)]^{N_s - n} \geq P_{\text{req}}, \qquad (15)$$

$$x_i \sum_{j=1}^{h_i} \frac{1}{\lambda_j} \left(\rho_j + \frac{\rho_j^2 + \lambda_j^2 \sigma_{s_j}^2}{2(1 - \rho_j)} \right) \leq D_{\text{req}} \quad i \in \{1, 2, \cdots, L\} \qquad (16)$$

$$x_i \in \{0, 1\}. \qquad (17)$$

In (15), $p(L) = 1 - \prod_{i=1}^{L} [1 - x_i p_i(h_i)]$ and is given in (2), where $i \in \{1, 2, \cdots, L\}$.

For a packet, only the source cluster-head solves this optimization problem to select the path(s), whereas the intermediate nodes just forward the packet. Many algorithms exist in the literature to solve this type of optimization problems and any one of these can be applied. We use the branch-and-bound approach [21] to solve the optimization problem. Since, solving an optimization problem is computationally expensive for the resource constrained sensor nodes, we propose a greedy algorithm in the next subsection to get a near optimal result.

B. Energy-Aware QoS Provisioning Algorithm

The greedy algorithm for energy-aware QoS provisioning finds a near optimal solution with reduced complexity and requires less information exchange. It is designed based on two

---

[1] Note that the proposed mechanism is also applicable for singe-sink WSNs, where all L paths are toward the only sink.

key approximations. First, the objective function in (14) requires the expected number of transmission attempts that a packet experienced for each of the available paths. An SCH needs the success rates of each of the downstream nodes in a path to find the expected total transmission attempts for that path. Therefore, the key approximation of the greedy algorithm is to minimize the number of total hops, instead of number of total expected transmission attempts for a packet. Thus, the greedy algorithm finds the subset of paths those have minimum total hop counts. Second, the greedy algorithm finds a near optimal solution using the following three steps.

- It selects the candidate paths those meet the delay bound. Let L_c denote the number of candidate paths.
- It finds the minimum number of paths, L_m, required to meet the reliability. Note that there could be a subset of paths having more paths but less total hop counts. The greedy algorithm will not consider those subsets of paths. Though, for WSNs, existence of such a subset is very unlikely and even if it happens, it will be very infrequent.
- If multiple feasible subsets exist with L_m paths, it selects the subset which has the minimum total hop counts.

Algorithm 1 presents the mechanism in detail. Lines 01–03 select the candidate paths those meet the delay requirement. If no path meets the delay requirement (i.e., $L_c = 0$), the algorithm drops the packet and exits (lines 04-05). In Section IV-G, we explain a way to extend the proposed mechanism to handle this.

The algorithm then finds the minimum number of paths (among the candidate paths) required to meet the reliability. We assume that the routing table in the CHs are sorted according to the estimated success rates of the paths. Lines 07–12 find the minimum number of paths, that meets the reliability requirement. If the required reliability is not met, the algorithm drops the packet and exits (lines 13–14).

The procedure *check* (L_m), defined in lines 40–51, checks the reliability of a subset of paths. Since, calculating the probability of meeting the reliability using (3) is computationally expensive, the greedy algorithm finds the probability of the successful delivery of a packet, $p(L)$, using (2). Note that the expected number of total packets received by the sinks is $E[N_r] = p(L)N_s$, due to its binomial distribution. Thus, the reliability is met, when $p(L) \geq R_{\text{req}}$. The procedure also counts the total hop counts of the subset of paths.

Finally, the algorithm finds the subset of L_m paths, those meet the reliability requirement and have minimum total hop counts. Note that the number of feasible subsets, those meet the delay requirement, is "L_c *choose* L_m." Lines 16–38 find each of the subsets and call procedure *check* (L_m). The algorithm terminates in line 39.

C. Complexity Analysis

In this sub-section, we obtain the worst case running time complexity of the proposed QoS provisioning algorithm.

Lines 01–05 select the candidate paths and take $O(L)$ time. The loop in lines 07–14 finds the minimum number of paths required to meet the reliability, L_m, among L_c candidate paths. It takes $O(L)$ time.

The procedure in lines 40–51, *check*(L_m), finds the expected reliability and total hop counts of a subset of paths. It takes $O(L_m)$ time. Finally, lines 16–38 find the subset of paths which has minimum total hop counts but meets the reliability. As mentioned, there are "L_c *choose* L_m" or $\frac{n!}{(L_c-L_m)!L_m!}$ subsets. For each subset the algorithm calls the procedure *check* (L_m) and

Algorithm 1 Energy-aware QoS provisioning.

```
Input: $L$ paths in $path[]$; delay ($del$), hop count ($hop$) and reliability ($rel$) of each path are given
Initialization: $flag = 0$ and $h_{\min} = 999$
// Paths in $path[]$ are sorted according to prob. of success
01: for ($i = 0$, $L_c = 0$; $i < L$; $i$++) //$L_c$ is no. of candidate paths
02: if ($path[i].del \leq D_{\text{req}}$)
03:     $candPath[L_c$++$] = path[i]$;
04: if ($L_c == 0$)   //no path meet the delay bound
05:   Drop the packet and exit
06:   //min no of paths, $L_m$, required to meet reliability
07: for ($i = 0$, $L_m = 0$, $i < L_c$; $i$++) // paths in $candpath[]$
08:   $tempPath[L_m$++$] = candPath[i]$;
09:   if (check ($L_m$) == 1)
10:     $flag = 1$; break;
11:   endif
12: endfor
13: if ($flag == 0$)  //existing paths do not meet the reliability
14:   Drop the packet and exit
15: //select $L_m$ paths having minimum total hop counts
16: for ($i = 0, l = 0; i \leq L_c - L_m; i++$) //each candidate path
17:   for ($j = i, count = 0; j < L_m;$   ) {
18:     $tempPath[count$++$] = candPath[$j++$]$
19:     if ($count == L_m$) //$tempPath[]$ contains $L_m$ paths
20:       if (check ($L_m$) == 0)
21:         $j = L_c$;
22:       if ($j < L_c$)]
23:         $count$ −−;
24:       else
25:         $loopcount = 0$;
26:     endif
27:     if ($j \geq L_c$)
28:       $loopcount$++;
29:       if ($count == L_m$)
30:         $count - = 2$;
31:       else
32:         $count - = loopcount$;
33:       if ($count < 0$)
34:         Break;
35:       $j = candPath[count] + 1$;
36:     endif
37:   endfor
38: endfor
39: End. // $L_m$ paths in selectPath[] are the selected paths
40: int check(int $l$) {
41:   $h = 0$; $p^c(L) = 1$; //$p^c(L)$ is prob. of packet loss
42:   for ($i = 0$; $i < l - 1$; $i$++)
43:     $h$+=$tempPath[i].hop$;$p^c(L)$*=($1 - tempPath[i].rel$);
44:   endfor
45:   $p(L) = 1 - p^c(L)$;
46:   if ($p(L) < R_{\text{req}}$)
47:     return 0;
48:   if ($h < h_{\min}$)
49:     $h_{\min} = h$; copy paths in tempPath[] to selectPath[]
50:   return 1;
51: }
```

thus takes $O\left(\frac{n!}{(L_c-L_m)!L_m!}L_m\right)$ times. However, as we consider the available paths are toward different sinks for a packet, therefore, we assume that the number of paths L is bounded by the number of sinks. Also, for $(L_c \leq 7)$, it is found that $\frac{n!}{(L_c-L_m)!L_m!} < L_c^2$. Thus, for at most 7 sinks in the networks, the worst case time complexity is $O(L_m L^2)$. Therefore, the proposed method is a feasible solution for sensor networks.

D. Reliability Estimation

Each node estimates the loss rate for each outgoing link toward a particular sink and uses hop-by-hop propagation of success rates through local information exchange.

D.1 Link Loss Rate Estimation

Each node estimates the link loss rate for every outgoing link using the weighted average loss interval (WALI) method discussed in [22]. It uses interval (in number of packets) between loss events to estimate the loss rate of a link. We denote the interval between the mth and the $(m+1)$th loss for the outgoing link of the ith path (i.e., the first hop of the path) as $l_{i,1}(m)$. Then, for the recent $1 \leq m \leq n$ losses, the average loss interval, $\hat{l}_{i,1}$, is

$$\hat{l}_{i,1(1,n)} = \frac{\sum_{m=1}^{n} l_{i,1}(m)w_m}{\sum_{m=1}^{n} w_m}$$

$$\hat{l}_{i,1(0,n-1)} = \frac{\sum_{m=0}^{n-1} l_{i,1}(m)w_m}{\sum_{m=1}^{n} w_m}$$

$$\hat{l}_{i,1} = \max(\hat{l}_{i,1(1,n)}, \hat{l}_{i,1(0,n-1)})$$

where $l_{i,1}(0)$ is the interval since the most recent loss and w_m is the weight given to each loss interval in the history. We have used $n = 8$ and $w = [1, 1, 1, 1, 0.8, 0.6, 0.4, 0.2]$ as our parameters. Intuitively, these choices give greater weight to recent packet loss, lesser weight to distant packet loss. Then, we compute the average success rate of the first hop of the ith path, $\hat{p}_{i,1}$, using average loss rate, $\hat{p}_{i,1}^c = 1/\hat{l}_{i,1}$, as

$$\hat{p}_{i,1} = 1 - \hat{p}_{i,1}^c. \tag{18}$$

D.2 Multihop Success Rate Estimation

The advantage of broadcast communication nature of the wireless networks enables to measure the success rate of a path by passive information exchange. When a node forwards a packet in a path, it includes the success rate of the path in the packet. The upstream nodes overhear the success rate of the downstream node and calculate their success rate for the path. Note that only the CHs overhear the success rates of the downstream nodes and it is not required that every node measures and overhears the success rates; thus, it ensures minimum overhead. The success rate of the ith path of a node, $\hat{p}_i(h_i)$, is given by

$$\hat{p}_i(h_i) = \prod_{j=1}^{h_i} \hat{p}_{i,j} = \hat{p}_{i,1} \prod_{j=2}^{h_i} \hat{p}_{i,j}$$

where $\hat{p}_{i,j}$ is the success rate of the jth hop and $\hat{p}_{i,1}$ is the success rate of the first hop (i.e., $j = 1$) of the node. Whereas, $\prod_{j=2}^{h_i} \hat{p}_{i,j}$ is the success rate of the path from the downstream node and the node overhears this from the forwarded packets of the downstream node.

D.3 Estimation of Success Rates of the Downstream Nodes

To measure the expected number of transmission attempts of the ith path using (12), $E[T_{\text{path}}(i)]$, an SCH needs the success rates, $\hat{p}_{i,j}$, of the downstream nodes in the path. Note that every downstream node knows the success rate of its outgoing link (see Section IV-D.1). However, exchanging the success rates is not feasible for WSN. Therefore, every SCH estimates the success rates from the success rate of a path. Since, path reliability, $\hat{p}_i(h_i)$, is the product of the success rates of the downstream nodes, so, the average success rate of the nodes in the path is the geometric mean of $\hat{p}_i(h_i)$. Therefore, the average success rate of the nodes in the ith path, $\hat{p}_{\text{path}}(i)$, is

$$\hat{p}_{\text{path}}(i) = [\hat{p}_i(h_i)]^{1/h_i}.$$

Also, note that for WSNs, due to the many-to-one routing [23], success rate gradually decreases. To estimate the individual success rate, we use average decrement/increment of success rate at successive downstream nodes. Since $\ln[\hat{p}_{\text{path}}(i)] = [\sum_{j=1}^{h_i} \ln(p_{i,j})]/h_i$, success rate at the jth hop is estimated as

$$\ln(\hat{p}_{i,j}) = \ln(\hat{p}_{i,1}) - \ln\left(\frac{\hat{p}_{i,1}}{\hat{p}_{path}(i)}\right)^{\frac{2(j-1)}{h_i-1}}, \quad j = 2, 3, \cdots, h_i$$

where $\frac{2}{h_i-1}\ln(\frac{\hat{p}_{i,1}}{\hat{p}_{\text{path}}(i)})$ is the average decrement/increment of the success rates of the successive nodes.

E. Delay Estimation

A simple way of path delay estimation is to estimate the link delay and multiply it with the hop count of the path, assuming the delays at each node are almost same. However, in sensor networks, data usually converge as they get closer to the sink, which increases the queueing delay at the downstream nodes, and as a result, total delay increases. Therefore, we separately estimate the queue service time and the average arrival rate at the downstream nodes to estimate the multihop delay.

E.1 Queue Service Time Estimation

Each node estimates the expected service time in (8a) and the variance of the service time in (8b), similar to the RTT estimation of standard TCP, using EWMA as

$$E[s] = (1-\alpha)E[s] + \alpha s,$$

$$\sigma_s^2 = (1-\beta)\sigma_s^2 + \beta\,|s - E[s]|^2$$

where α and β are used as tuning parameters to smooth the variations. Through extensive simulation, we have set the value

$\alpha = 0.12$ and $\beta = 0.20$, which produce the best estimation of long term average of service time and its variance. Note that for realistic sensor networks, this is reasonable because the service time depends on the historical behavior.

E.2 End-to-End Delay Estimation

Under the assumption that the traffic generation rates of all sensor nodes are same and each CH has almost equal number of sensors, the arrival rate at each forwarding node can be estimated using (9). Once a node can estimate the arrival rate of the downstream nodes and variance of the service time of the queue, it can easily estimate the end-to-end delay using (10).

F. *Further Enhancement of QoS*

As mentioned in Section III-D, a controlled and effective retransmission scheme can be used to achieve a high delivery ratio. We propose a simple and very efficient (almost guaranteed) retransmission scheme for sensor networks. We use packet classification and priority queueing to reduce the delay of certain packets, so that the intermediate nodes can retransmit them keeping the delay within delay bound. This also allows an SCH to forward a packet, when the available paths do not provide the required reliability.

F.1 Packet Classification Mechanism

We classify the packets in two different priorities.[2] In multipath routing, if L paths are selected, then L copies of a packet are sent. Among the L copies, one copy is marked as high priority packet and the rest are considered as normal packets. Finally, successive marked packets, from a single SCH, are forwarded in all L paths equally, so that every path has around $1/L$ marked packets.

F.2 Delay Reduction by Priority Queueing

We use priority queueing at each forwarding node to reduce the delay of the marked packets, which in turn allows us to retransmit them.

The queue of a sensor node is divided into number of virtual queues. We use four virtual queues ($VQ_0 - VQ_3$), each of which is a link list of fixed sized memory space (we refer this as a buffer). A packet is stored in a buffer. The purpose of the virtual queues are: i) VQ_0 stores the priority packets, ii) VQ_1 stores normal packets, iii) VQ_2 stores the priority packets those require retransmission and iv) VQ_3 keeps track of the empty buffers. When a packet arrives, a node detaches a buffer from the head of VQ_3 and stores the packet in it. The buffer is attached to the tail of VQ_0, if the packet is a priority packet; otherwise, to the tail of VQ_1. After successful transmission, the buffer of the packet is attached to the tail of VQ_3. Whereas, after unsuccessful transmission the buffer is attached to the tail of VQ_2, if the packet is a priority packet; otherwise, it is attached to the tail of VQ_3.

The packets in the virtual queues are processed in a weighted round robin fashion. The weight of VQ_0 and VQ_1 are set to $\frac{2}{3}$ and $\frac{1}{3}$, respectively. Though, a different set of weights can be assigned based on the QoS requirement. So, the delay of the priority packets are reduced, which allows retransmission of the priority packets in the intermediate nodes.

F.3 Efficient Retransmission

We use a reservation-based retransmission mechanism, where a priority packet is retransmitted after reserving the channel. Therefore, a sender does not retransmit a priority packet immediately after the unsuccessful initial attempt. Instead, it requests the receiver through the subsequent packets (until it gets the acceptance from the receiver) that it needs to retransmit a packet. For this, a 1-bit control flag is used in the header of the DATA and ACK packets. We define this as *Retransmission Flag* and denote this as ReTx. The sender turns On the ReTx flag of the data packet, if there is any priority packet to be retransmitted (i.e., VQ_2 is non-empty). After receiving the DATA packet, every node checks the ReTx flag. If the ReTx flag is ON, the DATA packet is used as the request packet for the retransmission. However, the receiver may or may not accept the request.[3] If the receiver accepts the request, then it turns on the ReTX flag in the ACK packet and this completes the reservation. The sender then retransmits a packet (from the head of VQ_2) after a duration of SIFS, without waiting for the DIFS period and the backoff process. Neighbors of the receivers freeze their backoff process, when they hear any ACK packet with ReTx flag turned ON. Whereas, neighbors of the sender freeze their backoff process, if they hear the transmission after SIFS period. Since all the neighbors of sender and receiver are well-aware about the transmission, there will be almost no collision. On the other hand, if the sender does not get the ACK packet or gets the ACK packet without the ReTx flag ON, then it does not retransmit the packet and tries the same procedure with the next packet. If the sender does not initiate the transmission after SIFS, the neighbors of the sender continue their backoff process after DIFS, considering either the previous transmission was unsuccessful or the receiver did not allow the retransmission. Fig. 3 shows the complete procedure.

Finally, a priority packet is retransmitted at a node as long as the remaining allowable delay is less than or equal to the estimated delay of the remaining path from that node. Therefore, we add a field in the header of each packet, which stores the remaining allowable delay of the packet. The SCH sets this value. An intermediate node updates the value by subtracting the delay of the packet at that node. This will also facilitate to discard a packet early and save energy.

G. *Discussions*

Some minor details of the proposed QoS provisioning scheme are worth mentioning. First, the number of sinks required for a specific sensor network. Since the maximum number of paths depends on the number of sinks used in the network, it plays an important role in terms of reliability. If the network requires more reliability then more sinks are better. However, the size of

---

[2] If the QoS requirement of the network is high, then more priority classes can be considered and our priority queueing mechanism can support this without much overhead.

[3] It may depend on the implementation, for example, based on the buffer status a receiver may not allow a transmitter to retransmit a packet. We assume that the receiver always allows the transmitter to retransmit one priority packet in each request.

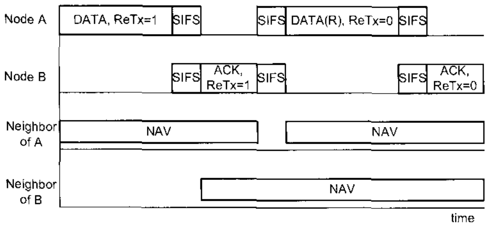


Fig. 3. Simple and efficient retransmission of priority packets.

the network is also a factor in determining the number of sinks, so that the multiple paths from a node do not interfere with each other. Also, the sinks should be placed with little care and with sufficient distance between them, so that any two paths from a sensor node do not interfere with each other.

Second, As mentioned in Section IV-B, an SCH drops a packet, if the available paths do not satisfy the delay requirement. A simple solution for this is to add a one bit control flag in the header of the DATA packet. The SCH can select the shortest path for that packet, and turn ON the flag. The forwarding nodes take special care to reduce the delay of the packet so that end-to-end delay decreases. One way to reduce the delay is to add one more virtual queue with the highest priority and store the packets in that queue in all forwarding nodes. But a general solution could be to increase the number of virtual queues and implementing a delay differentiation mechanism in processing the queues, where the packets those travel more hops get higher priority. Also, as mentioned in Section III-D, the reliability of a path decreases with the increase of the hop count, so a delay differentiation mechanism will allow more retransmissions for a packet having higher hop counts and thus increase the reliability. Since the queue management depends on the implementation, it should be decided before implementation. And based on the analysis presented in Section III, a specific implementation can be designed for certain reliability and delay requirements.

Third, so far we have considered that the nodes in the network generate data continuously. But, without any modification, our scheme can work with event based sensor networks. In the absence of an event, we assume that the sensors generate packets in a low rate and when an event occurs sensors generate packets with high rates. The event packets are QoS packets and the SCHs use multi-path QoS routing for those packets.

Finally, we considered that the sensor network is cluster-based, and thus, the CHs might consume more energy than a member and run out of energy quickly. We assume that the clustering mechanism rotates the CHs to overcome this. On the other hand, our protocol is equally applicable for a tree-based WSN (for each sink, there is a sink-rooted tree). In this case, the source of a packet can execute the multi-path QoS routing instead of the SCH. Furthermore, for balanced energy consumption, the parent nodes (or, the forwarding nodes) can be rotated based on their remaining energy.

V. PERFORMANCE EVALUATIONS

A. Simulation Environment

Table 3. Parameters used in simulation and their values.

Parameters	Value	Parameter	Value
Link bit rate	512 kbps	α	0.12
PHY header	192 μs	β	0.20
MAC header	224 bits	Buffer size	30 packets
ACK packet	112 bits	Packet size	32 Bytes
Slot time	20 μs	Initial energy	5 J/node
SIFS	10 μs	E_{elec}	50 nJ/bit
DIFS	50 μs	E_{amp}	100 pJ/bit/m^2
Min CW	32		

We have done extensive simulation to evaluate the performance of our proposed scheme in NS2. We have considered a network area of 150 m $\times$ 150 m and deployed 100 nodes in uniform random distribution. There are 4 sinks and the coordinations of the sinks are: [0, 75], [75, 0], [75, 150], and [150, 75]. We use static clustering and routing for the simulation. The clusters are formed in such a way so that the CHs form a connected dominating set. There are 12 clusters and the number of member sensors in each cluster is 6–10. From each cluster there is one path to each sink and the paths from each CH are selected based on the shortest distance from CH to the sink. The transmission range and the sensing range of the nodes are set to 30 m and 15 m, respectively; whereas, the link bandwidth is set to 512 kbps. Sensors generate data at a rate of 6 packets per second (PPS). Table 3 shows the simulation parameters.

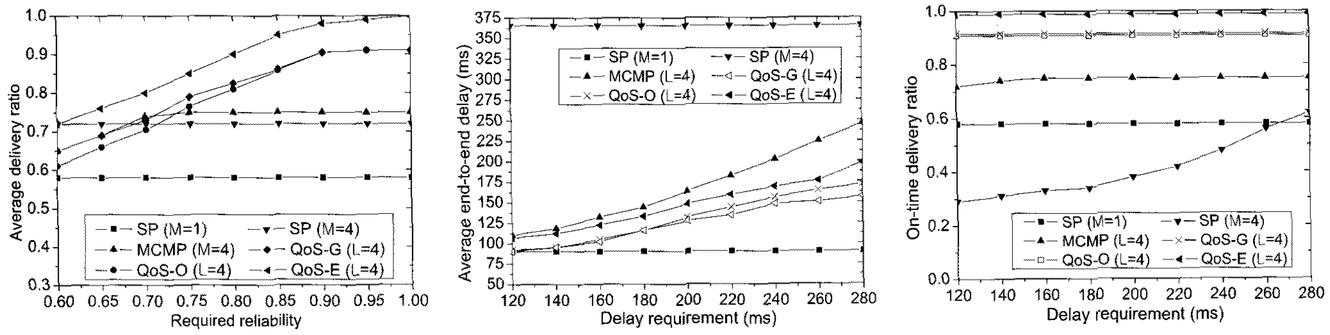
B. Performance Metrics

Performance metrics used for the performance evaluations are: average packet delivery ratio, on-time packet delivery ratio, average packet delay, average energy consumption per packet and network lifetime. Average packet delivery ratio is the number of packets received to the number of packets sent without considering the delay requirement; whereas, on-time packet delivery ratio considers the packets those meets the delay requirement. The average packet delay is the average end-to-end delay of the successfully delivered packets. Average energy consumption measures the average energy consumption for a single packet, if multiple copies are sent then the total energy consumption for a packet is the sum of the energy consumptions for all the copies of a single packet. Network lifetime measures the number of alive sensor nodes over times.

C. Simulation Results

Fig. 4 shows the quality of service (QoS) achieved by the proposed QoS provisioning mechanism with four available paths for multi-path routing. We compare the results with single-path routing with maximum retransmission limit 1 and 4 and MCMP [3]. We show the performance of the proposed mechanism for three different cases: i) QoS provisioning based on greedy path selection, termed as QoS-G, ii) QoS provisioning based on optimization based path selection, termed as QoS-O, and iii) enhanced QoS provisioning (that uses priority queueing and reservation-based retransmission) based on greedy path selection, termed as QoS-E.

In Fig. 4(a), average delivery ratio with different reliability requirements is shown without considering the end-to-end delay. Single-path routing with no retransmission achieves about 58% delivery ratio irrespective of the requirement, whereas,



(a) (b) (c)

Fig. 4. (a) Average delivery ratio for different reliability requirements, (b) average end-to-end delay for different delay requirements, and (c) on-time packet delivery ratio for different delay requirements.

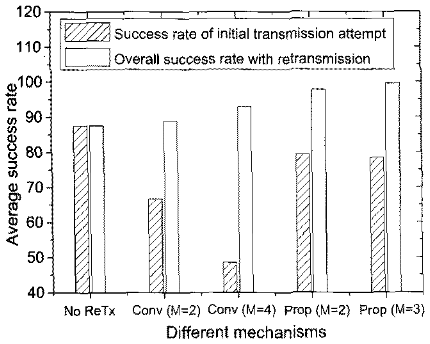


Fig. 5. Average success rate of the proposed retransmission mechanism.

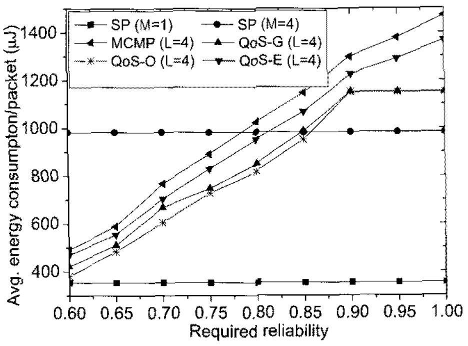


Fig. 6. Average energy consumption of a packet for different mechanisms.

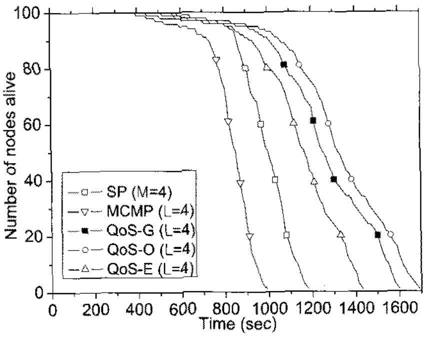


Fig. 7. Network lifetime for different mechanisms.

with maximum retransmission limit equal to 4, it is just above 70%. MCMP uses multi-path routing with single sink and uses optimization technique to select multiple paths and achieves required delivery ratio up to 75%, which is much less than that of the proposed mechanisms. Whereas, the proposed QoS-G mechanism achieves a maximum delivery ratio above 90% due to multi-sink, multi-path space diversity. Also, due to optimum path selection, the achieved delivery ratio is just above the required delivery ratio. The optimization-based QoS mechanism (QoS-O) has almost same delivery ratio as QoS-G, which indicates that the greedy algorithm does not sacrifice the QoS requirements. Finally, QoS-E mechanism achieves a delivery ratio very close to 100%, and outperforms all the other mechanisms in terms of delivery ratio.

Fig. 4(b) shows the average end-to-end delay of a packet with different delay requirements and there is no reliability requirement; every mechanism tries to maximize the delivery ratio. Therefore, we just measure the average delay of the successfully delivered packets. As expected, single-path routing with no retransmission experienced the minimum delay and single-path routing with maximum retransmission limit 4 experienced the maximum delay. The proposed mechanism, QoS-G, experienced a little higher delay than single-path routing with no retransmission and delay increases with the delay requirement. The optimization-based mechanism (QoS-O) has almost same delay as the QoS-G mechanism. The enhanced QoS mechanism (QoS-E) experienced more delay than both QoS-G and QoS-O and the reason is the retransmission of the packets in the intermediate nodes. On the other hand, MCMP experienced more delay than all three proposed mechanisms. However, for all four cases, the average delay is less than the delay requirement.

Fig. 4(c) shows the on-time delivery ratio for different delay requirements, where the reliability requirement is assumed to be 0.95. In single-path routing with no retransmission, the end-to-end delay is less than the delay requirement, and so the on-time delivery ratio is same as shown in Fig. 4(a). But for single-path routing with retransmission ($M = 4$), when the delay requirement is very strict, on-time delivery ratio is very less and even though many packets reach the sink, these packets are discarded due to late delivery. Therefore, the delivery ratio increases, when packets allow more end-to-end delay. However, for all the proposed QoS mechanisms (QoS-G, QoS-O, and QoS-E), delivery ratio does not decrease even if the packets demand lesser delay, since the algorithms select the only paths which meet the delay deadline. Finally, for MCMP on-time delivery ratio is almost as same as shown in Fig. 4(a).

Fig. 5 demonstrates the effectiveness of the proposed retransmission method and justifies the higher delivery ratio achieved by the enhanced QoS mechanism, QoS-E. In the figure, we compare the success rates of the proposed retransmission mechanism with the conventional retransmission mechanism of CSMA/CA. For the conventional method, we considered three different values of M, namely 1, 2, and 4; whereas, for the proposed mechanism the values of M are 2 and 3. For each case, we show the average success rate of a packet in the initial transmission attempt and the overall success rate of a packet with retransmission. For the conventional retransmission method, even though the overall success rate increases with retransmission,

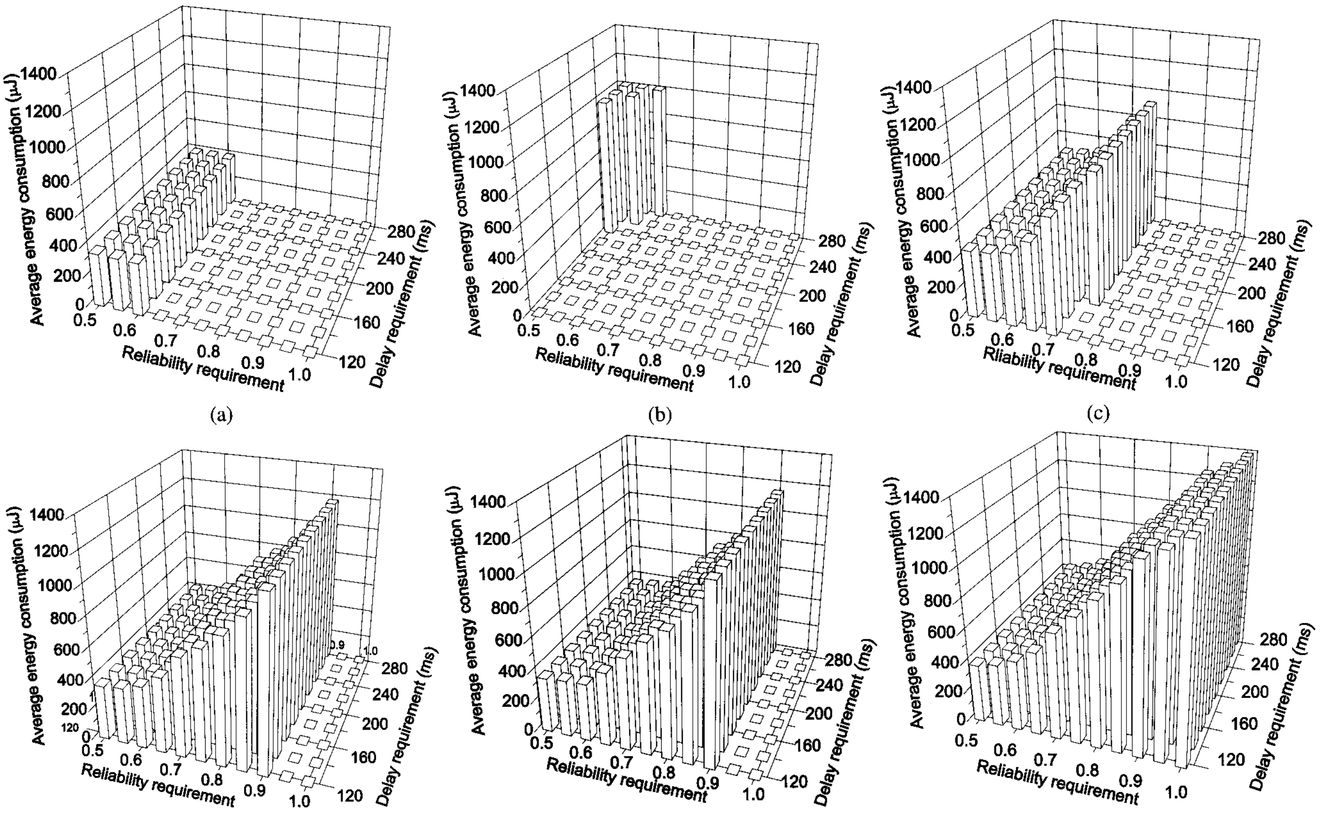


Fig. 8. Average energy consumption per packet for different reliability and delay requirements: (a) Single-path routing without retransmission, (b) single-path routing with maximum retransmission limit 4, (c) MCMP with $L = 4$, (d) proposed mechanism, QoS-G with $L = 4$, (e) proposed mechanism, QoS-O, with $L = 4$, and (f) proposed mechanism with QoS enhancement, QoS-E, with $L = 4$.

but it decreases the success rate of initial transmission attempt. Whereas, the proposed mechanism uses reservation-based retransmission, which not only increases the overall success rate but also does not decrease the success rate of the initial attempts.

Fig. 6 shows the average energy consumption of a packet with different reliability requirements. As expected, single-path routing with $M = 1$ consumes minimum energy. For low reliability requirement, the proposed mechanisms (QoS-O, QoS-G, and QoS-E) consume less energy than the single-path routing with retransmission $(M = 4)$. However, when the reliability requirement is high the proposed mechanisms consume more energy than single-path routing with retransmission and the extra energy consumption is due to the higher achieved reliability. The energy consumption of a packet in the enhanced QoS provisioning mechanism (i.e., QoS-E) is little higher than QoS-G due to retransmission of the packets. The optimization based mechanism (QoS-O) consumes less energy than QoS-G, since it selects the paths based on optimization. Thus, we admit that the greedy algorithm does not select the optimum energy consuming paths, though the difference in energy consumption is very little. MCMP consumes almost same energy as the proposed mechanisms for lower reliability requirement, but as the reliability requirement increases MCMP consumes more energy than both QoS-O and QoS-G, even though extra energy does not increase the reliability.

Fig. 7 shows the number of node alive with time, which actually shows the network lifetime. As mentioned earlier, multi-sink multi-path routing increases the lifetime of the network due to the spatially balanced energy consumption. As shown in the figure, the proposed mechanisms outperform others. The network lifetime is shortest for MCMP due to multi-path routing toward a single sink. And single-path routing with retransmission $(M = 4)$ performs better than MCMP. On the other hand, for QoS-E lifetime is little shorter due to the retransmission of packet and QoS-O has the highest lifetime among the proposed mechanisms.

Fig. 8 shows the average energy consumption of a packet for different mechanisms. Note that the figures only show the cases, where both the QoS requirements are satisfied. One important property is that energy consumption increases with the increased demand of reliability, and for a particular reliability requirement, energy consumption is almost independent of the delay requirement. A minimum value of the delay constraint determines whether a particular reliability can be satisfied or not. All the delay constraint values above the minimum value (for which the reliability is barely met), the energy consumption does not change much. Figs. 8(a) and 8(b) show that the energy consumption is independent of the QoS requirements for single-path routing with $M = 1$ and $M = 4$, respectively. Retransmission increases the energy consumption, but does not increase the reliability. MCMP increases the reliability due to multi-path routing but energy consumption increases at a high rate with increasing reliability requirement (Fig. 8(c)). Also, for MCMP the upper limit of reliability is only 0.75%. For a particular QoS requirements, all the proposed mechanisms consume less energy than both MCMP and single-path routing. However, among the proposed mechanisms, QoS-O (Fig. 8(e)) requires minimum energy and QoS-E (Fig. 8(f)) requires maximum energy for a packet. The reason is that QoS-O selects optimum paths based on expected energy consumption; whereas QoS-E requires maximum energy due to the retransmission of the packet. Though, the higher cost of QoS-E provides higher reliability than QoS-G (Fig. 8(d)) or QoS-O (Fig. 8(e)).

VI. RELATED WORKS

K. Akkaya and M. Yonis proposed an energy and QoS aware routing protocol in [24], which minimizes the delay of the real-time traffic but maximizes the throughput of non-real-time traffic. In [25], S. Tang and W. Li proposed a cluster based localized QoS controlling scheme for delay guarantee and optimal energy allocation. Dynamic traffic conditions are considered and data generation rates at individual clusters are dynamically controlled. However, reliability is not considered.

A packet delivery mechanism MMSPEED is proposed in [7], which provides service differentiation and probabilistic QoS guarantees in timeliness and reliability domains. For the timeliness domain, multiple network-wide speed options are adopted; while for reliability domain, probabilistic multi-path forwarding is used. A reinforcement learning technique is used in [26] to find out the optimal cost routing path and henceforth improves the overall performance of the network. They have used the delay and packet loss as the metrics of QoS. DEED [12] proposes a dynamic delay-constrained distributed minimum-energy dissemination scheme. A dissemination tree is updated in a distributed way without regenerating the tree from scratch; the energy consumption of the tree is minimized while satisfying end-to-end delay requirement.

An energy-aware dual-path routing scheme for real-time traffic is proposed in [27], that balances node energy utilization to increase the network lifetime. It considers the routing delay across the network and increases the reliability of the packets by introducing minimal data redundancy. To reap the full benefit of their routing scheme, they have given a prioritized MAC architecture. A multi-constrained, multi-path QoS provisioning mechanism is proposed in [3], which provides soft-QoS and uses approximation techniques.

VII. CONCLUSION

In this paper, we proposed a multi-constrained, multi-sink, and multi-path QoS provisioning mechanism for wireless sensor networks. We used probabilistic models to analyze the reliability, delay and energy consumption for both single-path and multi-path routing. Based on the analysis, we formulated an optimization problem, to provide both the reliability and delay guarantee with minimum energy consumption. A greedy algorithm is proposed to solve the optimization problem. To further enhance the reliability and reduce the end-to-end delay, we proposed a priority queueing mechanism and a simple but very efficient retransmission mechanism. Simulation results show that the proposed QoS provisioning scheme can provide reliability very close to 100% while keeping the delay within delay bound.

ACKNOWLEDGMENT

The authors like to thank the anonymous referees for their valuable and constructive comments, which helped greatly in improving the quality and presentation of this paper.

REFERENCES

[1] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: A survey," in *Proc. ICWN*, Las Vegas, Nevada, USA, 2004.

[2] P. V. Mieghem and F. A. Kuipers, "On the complexity of QoS routing," *Comput. Commun.*, vol. 26, no. 4, pp. 376–387, 2003.

[3] X. Huang and Y. Fang, "Multiconstrained QoS multi-path routing in wireless sensor networks," *Wireless Netw.*, vol. 14, no. 4, pp. 465–478, 2008.

[4] Özgür B. Akan, and I. F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 1003–1016, 2005.

[5] H. Zhang, A. Arora, Y. ri Choi, and M. G. Gouda, "Reliable bursty convergecast in wireless sensor networks," in *Proc. MobiHoc*, 2005, pp. 266–276.

[6] S. Chen and K. Nahrstedt, "An overview of quality of service routing for next-generation high-speed networks: Problems and solutions," *IEEE Netw.*, vol. 12, no. 6, pp. 64–79, Nov./Dec. 1998.

[7] E. Felemban, C. G. Lee, and E. Ekici, "MMSPEED: Multi-path multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 738–754, 2006.

[8] Y. Wang, W. Wang, and X.-Y. Li, "Efficient distributed low-cost backbone formation for wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 7, pp. 681–693, 2006.

[9] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *J. Clust. Comput.*, vol. 5, no. 2, pp. 193–204, 2002.

[10] L. Bao and J. J. Garcia-Luna-Aceves, "Topology management in ad hoc networks," in *Proc. MobiHoc*, 2003, pp. 129–140.

[11] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Overload traffic management for sensor networks," *ACM Trans. Sen. Netw.*, vol. 3, no. 4, p. 18, 2007.

[12] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Dynamic delay-constrained minimum-energy dissemination in wireless sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 4, no. 3, pp. 679–706, 2005.

[13] S. Chen and Z. Zhang, "Localized algorithm for aggregate fairness in wireless sensor networks," in *Proc. MobiCom*, 2006, pp. 274–285.

[14] *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*, IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), June 12, 2007.

[15] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[16] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*, 2nd ed., Athena Scientific, 2002.

[17] S. Ghahramani, *Fundamentals of Probability with Stochastic Processes*, 3rd ed., Prentice Hall, 2004.

[18] V. Rai and R. Mahapatra, "Lifetime modeling of a sensor network," in *Proc. DATE*, Mar. 2005.

[19] M. V. Ramesh and P. Ushakumari, "Threshold based data aggregation algorithm to detect rainfall induced landslides," in *Proc. ICWN*, 2008.

[20] L. Kleinrock, *Theory, Volume 1, Queueing Systems*, Wiley-Interscience, 1975.

[21] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 8th ed., McGraw-Hill Science/Engineering/Math, 2005.

[22] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. SIGCOMM*, 2000, pp. 43–56.

[23] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *Proc. SenSys*, 2004, pp. 148–161.

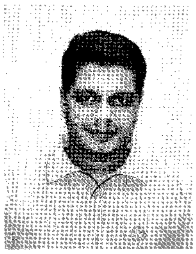
[24] K. Akkaya and M. Younis, "Energy and QoS aware routing in wireless sensor networks," *Clust. Comput.*, vol. 8, no. 2–3, pp. 179–188, 2005.

[25] S. Tang and W. Li, "QoS supporting and optimal energy allocation for a cluster based wireless sensor networks," *Comput. Commun.*, vol. 29, no. 13, pp. 2569–2577, 2006.

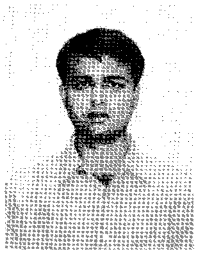
[26] N. Ouferhat and A. Mellouck, "QoS dynamic routing for wireless sensor networks," in *Proc. Q2SWinet*, 2006, pp. 45–50.

[27] A. Mahapatra, K. Anand, and D. P. Agrawal, "QoS and energy aware routing for real-time traffic in wireless sensor networks," *Comput. Commun.*, vol. 29, no. 4, pp. 437–445, 2006.

Muhammad Mahbub Alam received his B.S. degree in Applied Physics and Electronics and M.S. degree in Computer Science from the University of Dhaka, Bangladesh in 1998 and 2000, respectively. He received his Ph.D. from Kyung-Hee University, South Korea in February 2009. Since 2002, he has been working as a faculty member in the Computer Science and Information Technology (CIT) department in Islamic University of Technology (IUT), Dhaka, Bangladesh. Currently, he is working as a Research Faculty in Kyung-Hee University. His research interests include wireless networking and performance modeling and analysis of networking systems.



Md. Abdur Razzaque received the B.S. degree in Applied Physics and Electronics and M.S. degree in Computer Science from the University of Dhaka, Bangladesh in 1997 and 1999, respectively. He worked as a faculty in the Department of Computer Science and Engineering, University of Dhaka. His research interest is in the area of modeling, analysis and optimization of protocols and architectures for wireless sensor networks, distributed systems, etc. He is now Ph.D. student in the Department of Computer Engineering, Kyung-Hee University, Korea.



Md. Mamun-Or-Rashid received the B.S. and M.S. degree in Computer Science from the University of Dhaka, Bangladesh, in the year 2000 and 2002, respectively. He has received Ph.D. in February 2008 from Kyung-Hee University, Korea. During 2003 and 2004, he worked as a faculty member in Department of Computer Science and Engineering, University of Dhaka, Bangladesh. His research interest is in the area of modeling, analysis and optimization of protocols and architectures for wireless sensor networks, distributed systems, etc. He is now working as an Assistant Professor in the computer science and engineering department of Dhaka University, Bangladesh.

Choong Seon Hong received his B.S. and M.S. degrees in electronic engineering from Kyung-Hee University, Seoul, Korea, in 1983, 1985, respectively. In 1988 he joined KT, where he worked on Broadband Networks as a member of the technical staff. From September 1993, he joined Keio University, Japan. He received the Ph.D. degree at Keio University in March 1997. He had worked for the Telecommunications Network Lab, KT as a Senior Member of technical staff and as a Director of the networking research team until August 1999. Since September 1999, he has been working as a Professor of the School of Electronics and Information, Kyung-Hee University. He has served as a Program Committee Member and an Organizing Committee Member for International conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, and TINA. His research interests include ad hoc networks, network security and network management. He is a member of IEEE, IPSJ, KIPS, KICS, and KIISE.