Article

Data-Centric Multiobjective QoS-Aware Routing Protocol for Body Sensor Networks

Md. Abdur Razzaque, Choong Seon Hong [*] **and Sungwon Lee**

Department of Computer Engineering, Kyung Hee University, 1 Seocheon, Giheung, Yongin, Gyeonggi, 446-701, Korea; E-Mails: m_a_razzaque@yahoo.com (M.A.R.); drsungwon@khu.ac.kr (S.L.)

[*] Author to whom correspondence should be addressed; E-Mail: cshong@khu.ac.kr; Tel.: +82-31-201-2532; Fax: +82-31-204-9082.

Abstract: In this paper, we address Quality-of-Service (QoS)-aware routing issue for Body Sensor Networks (BSNs) in delay and reliability domains. We propose a data-centric multiobjective QoS-Aware routing protocol, called DMQoS, which facilitates the system to achieve customized QoS services for each traffic category differentiated according to the generated data types. It uses modular design architecture wherein different units operate in coordination to provide multiple QoS services. Their operation exploits geographic locations and QoS performance of the neighbor nodes and implements a localized hop-by-hop routing. Moreover, the protocol ensures (almost) a homogeneous energy dissipation rate for all routing nodes in the network through a multiobjective Lexicographic Optimization-based geographic forwarding. We have performed extensive simulations of the proposed protocol, and the results show that DMQoS has significant performance improvements over several state-of-the-art approaches.

1. Introduction

Wireless Body Sensor Networks (BSNs) have been receiving more and more attention in academia and industry in recent years, especially under the impending healthcare crisis and due to the availability of much less expensive biomedical sensors (BMSs) with certain computation and communication capabilities. The primary target applications of BSN research, so far, are medical healthcare services, addressing the weaknesses of traditional patient data collection system, such as imprecision (qualitative observation) and undersampling (infrequent assessment) [1,2]. BSNs can offer a paradigm shift from managing illness to proactively managing wellness by focusing on prevention and early detection/treatment of diseases, thereby reducing healthcare costs. They can capture accurate and quantitative data from a variety of sensors (e.g., temperature, blood pressure, heart rate, electrocardiogram (ECG), *etc.*) for longer time periods. BSNs with real-time sensing capability would also help in protecting those exposed to potentially life-threatening environments, including soldiers, first responders, and deep-sea and space explorers [3]. Therefore, on-time and reliable data delivery to the control center is very important for BSN applications.

The Quality-of-Service (QoS) provisioning in BSNs is a challenging task, mainly due to two reasons. First, the dynamic network topology, time-varying wireless channel and scarcity of node energy, computation power and channel bandwidth pose challenges on the design of QoS support schemes in BSNs. Second, there exist wide variations in data generation rate and delay- and loss-tolerances amongst the data packets generated by different types of BMSs [2]. For example, some low data rate BMSs (e.g., heartbeat, blood pressure, electroencephalogram (EEG) sensors) may generate very time-critical data packets, which must be delivered at the destination sink within a guaranteed end-to-end delay deadline; data packets from some of these sensors might also require high reliability. In contrast, some high data rate BMSs (e.g., streaming of ECG signals) may allow a certain percentage of packet losses. Therefore, a scalable solution with data-centric QoS-aware routing that can provide a clear differentiation in route selection between data packets with multiobjective QoS requirements, is greatly required for BSNs.

In the literature, several mechanisms (outlined in Section 2) have been proposed to mitigate the problems of multiobjective QoS provisioning in wireless sensor networks. However, to the best of our knowledge, no effective solution to this problem has yet been proposed so far particularly for BSNs. The key contribution of this paper is the first complete design and evaluation of data-centric multiobjective QoS-aware routing for BSNs that has clear differentiation in route selection between multiple traffic types with respect to their QoS requirements. It also trades off the energy cost and protocol operation overheads while improving the network performance.

The proposed data-centric multiobjective QoS-aware routing protocol, DMQoS, uses modular architecture and it exploits geographic locations to implement localized routing. An important property of the proposed protocol is the end-to-end QoS-aware routing with local decisions at each intermediate node without end-to-end path discovery and maintenance. This property is important for scalability to large-scale sensor networks, self-adaptability to network dynamics, and appropriateness to multiple classes of traffic flows. The routings of delay-critical and reliability-critical packets are handled separately by employing independent modules for each, whereas for the most critical packets having both stringent delay and reliability constraints, the corresponding modules operate in coordination to

guarantee the required service. While the delay control module chooses the next-hop router node offering higher velocity of data packets, the reliability control module injects minimal redundant information by exploiting high reliability links. A Lexicographic Optimization (LO) [4] based approach is used to tune trade-off between the geographic progress and the residual energy levels. Therefore, our model considers not only the QoS requirements, but also the energy cost optimality of the routing path to improve the overall network performance.

We evaluated our routing protocol in terms of end-to-end packet delays, packet delivery ratio, average energy consumption per packet, and routing overhead for variable source traffic loads and wireless link bit error rates. The results show that DMQoS demonstrates a substantial improvement in required data delivery services over several state-of-the-art approaches and we provide insights into the sources of the improvement.

The rest of the paper is organized as follows. In Section 2, we describe the key limitations of some existing QoS-aware routing protocols. Subsequently, we present a body area sensor network model and assumptions we have considered in Section 3. Section 4 presents the proposed DMQoS architecture in detail, followed by the performance evaluations using Network Simulator-2 [5] in Section 5. Finally, we conclude the paper in Section 6.

2. Related Works

Major challenges and open research issues on QoS provisioning in BSNs have been presented in [6] and a cross-layer QoS framework (that spans over three layers) for biomedical sensor networks has been proposed in [7]. In EDDD [8], an energy-efficient differentiated directed diffusion mechanism has been developed that provides with service differentiation between real time and best effort traffic. However, their designs are neither scalable nor adaptive to dynamic environment. In recent years, QoS routing in location-aware wireless sensor networks has received much research interests due to its inherent characteristics of (i) being scalable to large networks, (ii) making routing decisions based on local neighborhood information, and (iii) being very adaptive under dynamic changes and mobility as only a node's neighborhood is affected. In [9], a reinforcement learning-based routing model for BSNs is proposed that selects a QoS route via computing neighborhood node's Q-values and position information, but does not consider energy at all. Directional Geographic Routing (DGR) [10] constructs an application-specific number of multiple disjoint paths for routing real time video communications data in wireless sensor networks. MCMP [11] uses link delay and reliability as routing decision parameters, where data packets are duplicated at source nodes by solving optimization problem. But, this approach considers neither residual energy nor progress speed. Hence, packets may get routed to a node which is highly congested and/or energy critical. MMSPEED [12] also sends duplicate packets (probabilistically) toward multiple paths and multiple reliability- and delay-bound packets are considered for QoS provisioning. However, routing in MMSPEED fully avoids energy consideration, reducing its applicability for BSNs. A hybrid geographic routing (HGR) protocol has been designed in [13] to achieve an efficient tradeoff between energy efficiency and delay performance. A reliable and energy-efficient routing protocol (REER) has been developed in [14] exploiting geographic information and cooperative communications.

The proposals in DARA [15] and LOCALMOR[16] have some similarities with our DMQoS protocol. In DARA [15], a weighted aggregate routing metric consisting of geographic progress, delay and energy is considered for supporting critical and non-critical data packets through defining long- and short-range forwarding zones, respectively. However, the use of same routing function for both the packet types deteriorates the QoS performance. In LOCALMOR [16], the routing functions have been separated for multiple packet types; however, it uses fixed number of sinks (primary and secondary sinks) and all packets are blindly duplicated toward both the sinks, making it unscalable. Also, it increases the overhead of sending too many duplicate packets.

The distinguished features of our DMQoS protocol design from those of the above mentioned works are as follows. Its routing functions are distinct for different packet types based on their QoS requirements. Its architecture is modular and the routing modules cooperate to render enhanced QoS services to different traffic classes. It uses Lexicographic Optimization (LO) for trading-off energy and service.

3. Network Model and Assumptions

Network Model   We assume that several biomedical sensors (BMSs) are attached to a human body, and that they acquire sensor data and transmit to a central node, namely body sensor mote (BSM), which is responsible for collecting raw data, processing (coding, aggregation, *etc.*) and forwarding them toward the sink node in multihop fashion, as shown in Figure 1. They might have one or more sensing devices as well. Note here that these central nodes, which are also considered in [16], are different from personal servers (e.g., PDA) as used in other works [17] and references therein. These sensor motes have relatively high energy and computing capability compared to tiny BMSs. We also assume that there may be several sink nodes $\mathcal{S}$ in the network and any sink node $s \in \mathcal{S}$ is primarily responsible for collecting and recording the patient's data and then uploading this data to the medical care server via the Internet. In this architecture, the master BSM node and the slave BMSs together form a cluster of a body sensor network. As in Figure 1(b), a set of such cluster heads $\mathcal{N}$ form the backbone of the network. The key idea used in this network architecture is to move much of the network and protocol complexity away from the power-constrained BMSs and into the much more capable cluster head node. Note that the above network architecture is not only suitable for medical healthcare applications as in [18] but also for sports, battlefield, rescue operations, *etc.*, those require large number of sensor nodes to be deployed. We also assume that each cluster head node has equal initial energy e_{init}, and the residual energy of any node $i \in \mathcal{N}$ is denoted by $e_{res}(i)$.

Geographic Information Consideration   In many applications of BSNs, the knowledge of the location of an event is also desired. Hence, it is important that the location of the network cluster nodes be known. If the nodes know their global or relative coordinates, distributed or stateless routing schemes can be employed, mitigating the need for propagation and updating of routing tables across the network. Distributed routing schemes based on the geographic locations of the nodes have been proposed and widely examined in [12,15] and references therein. Like them, we also assume that the coordinates of the nodes are known locally, *i.e.*, each node knows its own coordinates as well as those of its routing

neighbors through execution of a HELLO protocol. All nodes would also know the coordinates of the sink(s).

Figure 1. Body Sensor Network (BSN).

(a) Cluster of a BASN

(b) Architecture of a Body Area Sensor Network (BASN)
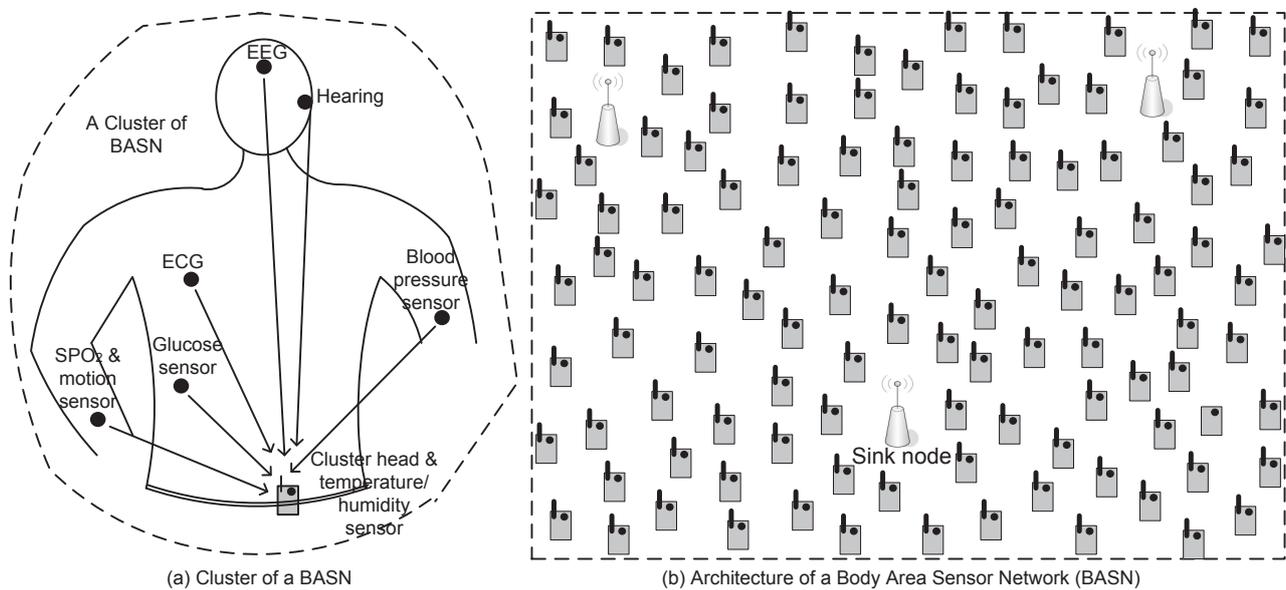
Constraints We assume that every delay-sensitive packet has a lifetime t_{life}, specified by the application layer, which indicates the time limit within what the packet should be delivered to the final recipient; otherwise, the information in the packet is useless. For the reliability-sensitive packets, the application layer also tags the required reliability level R with a packet. Moreover, routing of all data packets should be energy-aware in order to extend the network lifetime.

HELLO Protocol Each cluster head node broadcasts a HELLO packet, periodically or upon observing significant changes in some parameter values, including its *current position, residual energy* and *average queuing delays* of different packet types in the node. This HELLO protocol works in similar way as in [15,19]. Upon reception of a HELLO packet, the neighbor cluster head nodes update their neighbor table entries. A new entry is added into the neighbor table when a new node moves into vicinity, and an existing entry is deleted when a neighboring node moves away or breaks down, which can be determined when a HELLO packet is not received during a predefined period of time (timeout). Thus, HELLO packets make the neighborhood information available to each cluster head node at a cost of additional energy and bandwidth. Time between broadcasts represents a trade-off between overhead communications and outdated cost function parameters. Therefore, update frequency should be carefully chosen to maintain a proper balance between information freshness and cost.

4. The Proposed QoS-Aware Routing

Figure 2 shows the components and their interconnections in our proposed data-centric multiobjective QoS-aware routing protocol, DMQoS. The DMQoS consists of five modules: the dynamic packet

classifier, delay control, reliability control, energy-aware geographic routing and QoS-aware queuing and scheduling modules. The details of these modules are presented in the following subsections.

Figure 2. Data-centric multiobjective QoS-aware routing architecture.
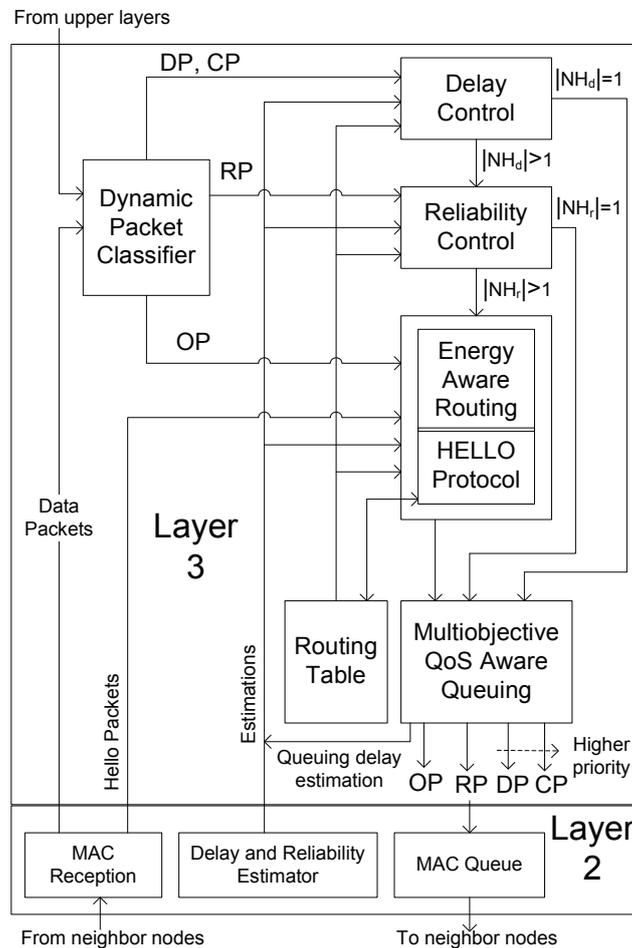
4.1. Dynamic Packet Classifier

The service differentiation paradigm used in this paper is as follows. We define four classes of data packets - ordinary data packets (OP), reliability-driven data packets (RP), delay-driven data packets (DP) and most critical data packets (CP). The CP packets are given the highest priority due to their stringent delay and reliability constraints, e.g., they may carry electroencephalogram (EEG) and electrocardiogram (ECG) monitoring information during a critical situation such as a surgery. The next higher priority is given to DP packets which should be delivered within a predefined deadline, but may tolerate reasonable packet loss, e.g., video streaming. The RP packets should be delivered without loss, but do not need to be immediate or within a hard deadline, such as vital signal monitoring, respiration monitoring and PH-level monitoring. The lowest priority is given to the OP packets that are corresponding to regular measurements of patient physiological parameters, like body temperature, heartbeat, *etc.*, that typically indicate normal values.

As shown in Figure 2, on reception of data packets either from application layer or from neighbor nodes, the dynamic packet classifier (DPC) of a node assigns them to one of the proper aforementioned categories and feeds them into the respective module in first-come-first-serve (FCFS) manner.

4.2. Energy-Aware Geographic Forwarding

Rather than using traditional end-to-end path discovery-based routing, we use localized packet forwarding that implements hop-by-hop routing. This deferred choice gives each packet transmission multiple opportunities to make progress toward the destination.

Our goal is to select a downstream node that has comparatively higher residual energy and gives higher geographic progress toward the destination sink. Note that while the second criterion decreases the number of hops between the source and the destination, the first one attempts to balance the energy consumption among the candidate downstream nodes. Our proposed energy-aware geographic forwarding (EAGF) uses a multiobjective Lexicographic Optimization (LO) approach [4] to manage this trade-off. In LO, the objective functions are arranged according to their absolute importance and the most important objective function is maximized (or minimized) first subject to the original constraints. If this problem has a unique solution, it will solve the whole multiobjective optimization problem. Otherwise, the second most important objective function is maximized (or minimized). Now, in addition to the original constraints, a new constraint is added. This new constraint is imparted to guarantee that the most important objective function preserves its optimal value. If this problem has a unique solution, it solves the original problem; otherwise, the process goes on as above.

In solution to our problem, there are only two objective functions of which the first one, the maximization of the geographic progress, is the most important. Let the objective functions be arranged according to the lexicographic order, with the most important function being, $f_1(j) = \frac{dist(i,s)-dist(j,s)}{dist(i,s)}, \forall (i,j) \in \mathcal{N}, \forall s \in \mathcal{S}$, where $dist(a,b)$ denotes the geometric distance between nodes a and b; and, the least important one being $f_2(j) = \frac{e_{res}(j)}{e_{init}}$. Thus, we write the lexicographic problem for any node $i \in \mathcal{N}$ as follows,

$$\text{lex maximize} \qquad f_1(j), \ f_2(j) \tag{1}$$
$$\text{subject to}$$
$$j \in \mathcal{N}_i, \tag{2}$$

where, $\mathcal{N}_i$ is the list of i's single hop neighbor nodes. The above LO problem can be divided into two separate problems with different constraint sets. The first problem is formulated as

$$\text{maximize} \qquad f_1(j) \tag{3}$$
$$\text{subject to}$$
$$j \in \mathcal{N}_i \tag{4}$$
$$dist(i,\mathcal{S}) > dist(j,\mathcal{S}), \quad \forall j \in \mathcal{N}_i \tag{5}$$
$$dist(i,j) \geq \overline{dist(i,j)}, \quad \forall j \in \mathcal{N}_i \tag{6}$$

and its solution j_1^* and $f_1^* = (j_1^*)$ is obtained; here, $\overline{dist(i,j)}$ is the average distance from node i to all neighbor nodes j. Then the second problem is formulated as

$$\text{maximize} \qquad f_2(j) \tag{7}$$
$$\text{subject to}$$
$$j \in \mathcal{N}_i \tag{8}$$
$$dist(i,\mathcal{S}) > dist(j,\mathcal{S}), \quad \forall j \in \mathcal{N}_i \tag{9}$$
$$dist(i,j) \geq \overline{dist(i,j)}, \quad \forall j \in \mathcal{N}_i \tag{10}$$
$$e_{res}(j) > \overline{e_{res}(j)}, \quad \forall j \in \mathcal{N}_i \tag{11}$$
$$f_1(j) = f_1^* \tag{12}$$

and the solution of this problem is j_2^* and $f_2^* = (j_2^*)$; here, $\overline{e_{res}(j)}$ is the average residual energy level of the neighbor nodes j of i. If j_2^* does not produce a unique solution, we break the tie by selecting the one that produces the most geographic progress.

Note that the nice property of LO is its simplicity, and such a lightweight but effective method is suitable for resource-constrained BSNs. Even though LO has the drawback of neglecting less important objective functions when the most important one produces the unique solution, in our case, such situations should occur less frequently due to the high density of the network nodes. Moreover, LO exploits only local information to make routing decisions. The absence of a global routing scheme reduces the networks setup and updating costs, eliminates the need for storage of network-wide routing information at each node, and alleviates the possibility of incorrect information at the nodes as changes in system topology occur. What requires only is that the nodes must broadcast an upkeep packet including node identification, current location information, average queuing delays, and remaining energy periodically throughout its lifetime.

4.3. Reliability Control

Link quality degradation, congestion, node mobility, link failure(s), node failure(s), *etc.* may cause packet losses, which affect the reliable data delivery to the destination(s). It has been shown in the literature that the probability of successful packet delivery to the destination can be increased by sending duplicate packets over multiple spatially separated routes [11,12,15,16]. However, choosing the most appropriate next-hop nodes (NH_r) of multiple routes toward different sinks to assure end-to-end required reliability as well as determining the optimized number of such next-hop nodes ($NH_{r,opt}$) are challenging problems.

In our reliability control algorithm (see Algorithm 1), we use greedy approach to solve the above problems. At first, each node i, for each sink $s \in \mathcal{S}$, identifies a candidate downstream node j that produces the maximum link reliability $\hat{r}_{i,j}$ and stores it in the variable NH_r (lines 1–3); the estimation method of the parameter $\hat{r}_{i,j}$ will be discussed in Section 4.5. If $|NH_r|$ returns a NULL, the packet is dropped immediately; and, if it returns only one node, the packet is forwarded toward that next-hop node (lines 4–9) provided that its offered reliability is greater than the required reliability R; otherwise, the packet is dropped due to unavailability of feasible path. In the case $|NH_r|$ returns multiple nodes, the nodes from the list NH_r are chosen one after another (in descending order of their $\hat{r}_{i,j}$ values) until

their aggregate reliability becomes greater than or equal to the required reliability, R (lines 11–17). Since the reliability is a multiplicative metric, we calculate the failure probability $F = F \times (1 - \hat{r}_{i,j})$, as presented in line 16. In fact, the above operation produces the optimal number of next-hop nodes $|NH_{r,opt}|$ that determines the degree of packet duplication for the reliability control. Note also that the value of $|NH_{r,opt}|$ ranges between 1 (best case) and the maximum number of sinks in the network (worst case). Therefore, our reliability control algorithm uses only the adequate number of duplicate packets, which in turn minimizes the routing and energy overheads compared to those of other approaches.

Algorithm 1 Reliability Control Algorithm, *at each source node i.*

INPUT: RP or CP packets, required reliability R and i's single hop neighbor nodes $\mathcal{N}_{i,s}$, $\forall s \in \mathcal{S}$

1. **for** each $s \in \mathcal{S}$ **do**
2. $\quad\quad NH_r = \{j \in \mathcal{N}_{i,s} : \hat{r}_{i,j} = \max\limits_{j \in \mathcal{N}_{i,s}} (\hat{r}_{i,j})\}$
3. **end for**
4. **if** $(|NH_r| ==$ Null$)$ **then**
5. $\quad\quad$ Drop the packet immediately;
6. **else**
7. $\quad\quad$ **if** $(|NH_r| == 1)$ **then**
8. $\quad\quad\quad$ **if** $(\hat{r}_{i,j} \geq R, \quad j \in NH_r)$ **then**
9. $\quad\quad\quad\quad$ $j \in NH_r$ is the desired next hop node and send the packet to the outgoing queue;
10. $\quad\quad\quad$ **else**
11. $\quad\quad\quad\quad$ Drop the packet immediately;
12. $\quad\quad\quad$ **end if**
13. $\quad\quad$ **else**
14. $\quad\quad\quad$ Sort NH_r in descending order of $\hat{r}_{i,j}$
15. $\quad\quad\quad$ $NH_{r,opt} =$ first $j \in NH_r$
16. $\quad\quad\quad$ $F = (1 - \hat{r}_{i,j}),\quad$ first $j \in NH_r$
17. $\quad\quad\quad$ **while** $(1 - F < R)$ **do**
18. $\quad\quad\quad\quad$ Add next $j \in NH_r$ to the set $NH_{r,opt}$;
19. $\quad\quad\quad\quad$ $F = F \times (1 - \hat{r}_{i,j})$
20. $\quad\quad\quad$ **end while**
21. $\quad\quad\quad$ Call EAGF with $NH_{r,opt}$ as input instead of $\mathcal{N}_{i,s}$;
22. $\quad\quad$ **end if**
23. **end if**

4.4. Delay Control

This unit concerns the routing strategy for on-time delivery of time critical emergency packets. The delay-guaranteed service defines the maximum allowable latency, bounded by the lifetime of a packet (t_{life}), required by the application. The total latency is experienced by a packet to traverse the network nodes from the source to the destination. At the network layer, the end-to-end packet latency is the sum of the processing delay, the transmission delay, the queuing delay, and the propagation delay. The queuing delay contributes most significantly to the total latency followed by the transmission delay; the

other delays are negligible. The CP and DP packets should travel through the next-hop nodes that provide with higher speeds so that the delay-guaranteed service is maintained.

Our delay control algorithm is presented in Algorithm 2 that works as follows. Each node i first computes the required velocity of a packet, $v_{req}(s) = \frac{dist(i,s)}{t_{life}[packet.type]}$, toward any sink $s \in \mathcal{S}$ based on its distance from the sink, $dist(i,s)$, and the remaining lifetime of the packet, $t_{life}[packet.type]$. As a packet travels, intermediate nodes update the packet's remaining lifetime as follows, $t_{life} = t_{life} - t_{elapsed}$, where $t_{elapsed}$ is the elapsed time of the packet at an intermediate node. We measure the *elapsed time* at each node i and piggyback it to the packet so that the following node j can determine the remaining time to deadline without using a globally synchronized clock. For this, when a node i receives the last bit of a packet, its MAC layer tags $t_{arrival}$ to the packet. This packet is processed by the network layer and forwarded to the chosen next-hop node j via the MAC layer. Note that the MAC layer of i requires some time to capture the channel using an RTS/CTS handshake and may transmit the packet several times until receiving ACK from j. For i to piggyback the accurate elapsed time, the MAC layer updates the field of elapsed time $t_{elapsed}$ just before it actually transmits the packet to the physical link as follows, $t_{elapsed} = t_{departure} + t_{transDelay} - t_{arrival}$, where $t_{departure}$ is the time at which node i transmits the first bit of the packet to the physical link and $t_{transDelay}$ is the transmission delay of the packet which can be computed using the transmission rate and packet length. Thus, once node j successfully receives the packet, the packet contains the correct measurement of the elapsed time at node i and computes the remaining lifetime of the packet.

After that the node i calculates the velocities offered by candidate next-hop nodes j toward a sink $s \in \mathcal{S}$, $v_j(s)$, by taking into account the packet's waiting time at the queue of node i, $\hat{d}_q^i[packet.type]$, the estimated packet transmission delay of node i, $\hat{d}_{tr}^i$, and the packet's waiting time at the candidate next-hop node j, $\hat{d}_q^j[packet.type]$ (see instruction 4). The consideration of queuing delay at a candidate next-hop node is very important for localized and delay-constrained routing because it indicates the traffic forwarding efficiency of the node. Nodes in the congested (or high traffic) area may have higher queuing and transmission delays, and thus their offered velocities will be lower. Hence, they will not be able to meet the required velocity level of the CP or DP data packets.

For all sinks, after computing the velocities of all candidate nodes, the algorithm then identifies the set of next-hop nodes NH_d that are supposed to meet the required delay deadline (see instruction 6). Note here that there could be several non-overlapping paths from a source to the destination(s) even though they may not be the shortest paths. A non-shortest path is acceptable as long as it can deliver a packet within its end-to-end delay deadline. The estimation methods for queuing and transmission delays will be discussed in Section 4.5.

In the case in which $|NH_d|$ returns a NULL value, *i.e.*, there is no neighboring router node satisfying the required velocity level, the packet is dropped immediately in order to prevent the network nodes from expending unnecessary energy. If $|NH_d|$ returns only one node, the packet is sent to the outgoing queue putting that node as the next-hop. On the other hand, if $|NH_d|$ includes several nodes, either the EAGF or the reliability control algorithm is called depending on the packet type in order to select the most appropriate router(s) from this set.

Algorithm 2 Delay Control Algorithm, *at each node i.*

INPUT: DP or CP packets, i's single hop neighbor nodes $\mathcal{N}_{i,s}$, , $\quad \forall s \in \mathcal{S}$

  1. **for** each $s \in \mathcal{S}$ **do**

  2.   Required velocity, $v_{req}(s) = \frac{dist(i,s)}{t_{life}[packet.type]}$

  3.   **for** each $j \in \mathcal{N}_{i,s}$ **do**

  4.   Offered velocity, $v_j(s) = \frac{dist(i,s)-dist(i,j)}{\hat{d}_q^i[packet.type]+\hat{d}_{tr}^i+\hat{d}_q^j[packet.type]}$

  5.   **end for**

  6. $NH_d = \{j \in \mathcal{N}_{i,s} : v_j(s) \geq v_{req}(s)\}$

  7. **end for**

  8. **if** ($|NH_d|$ == Null) **then**

  9.   Drop the packet immediately;

10. **else**

11.   **if** ($|NH_d|$ == 1) **then**

12. $j \in NH_d$ is the desired next hop node and send the packet to the outgoing queue;

13.   **else**

14.   **if** ($packet.type == DP$) **then**

15.   Call EAGF with NH_d as input instead of $\mathcal{N}_{i,s}$;

16.   **else**

17.   **if** ($packet.type == CP$) **then**

18.   Call reliability control algorithm with NH_d as input;

19.   **end if**

20.   **end if**

21.   **end if**

22. **end if**

4.5. Parameter Estimations

Estimation of Queuing Delay As shown in Figure 2, several queues are used for storing the different classes of data packets. The transmission scheduling of data packets from different queues is also different, to be discussed soon, and therefore, each class of data packets from a single node i may experience separate estimation of queuing delay d_q^i. We calculate this delay as the time difference between the insertion time of the packet at the queuing system and the time at which it enters into the position of transmission. Note here that we omitted the processing time of a packet at the node because it is trivial and is assumed to be almost the same for all packet types.

An Exponentially-Weighted Moving Average (EWMA) formula may suffice to estimate the running average queuing delays for each packet type $\hat{d}_q^i[packet.type]$. Algorithm 3 shows the steps for estimating queuing delays, wherein, the weighting for each older data point decreases exponentially, giving much more importance to recent observations $d_q^i[packet.type]$. The degree of weighing decrease is expressed as a constant smoothing factor α, a number usually between 0 and 0.4. In our simulation, we use $\alpha = 0.2$.

Algorithm 3 Queuing Delay Estimator, *at each node i.*

---

1. **Initialization:** $\hat{d}_q^i[packet.type] = 0$
2. **for** each packet transmission **do**
3.   **if** $\hat{d}_q^i[packet.type] = 0$ **then**
4. $\hat{d}_q^i[packet.type] = d_q^i[packet.type]$
5.   **else**
6. $\hat{d}_q^i[packet.type] = (1 - \alpha)\hat{d}_q^i[packet.type] + \alpha d_q^i[packet.type]$
7.   **end if**
8. **end for**

Estimation of Transmission Delay Each node $i \in \mathcal{N}$ in the MAC layer measures the average packet transmission delay $\hat{d}_{tr}^i$ from itself using a Weighted Average Transmission Delay (WATD) method. WATD is very similar to the WALI method [20], which measures packet loss intervals for TCP congestion control. The WATD that works as follows. It measures the instantaneous delay for each packet transmission, $d_{tr}^i(n)$, which is the delay for the nth packet transmission measured as the time duration from the time at which the packet is ready for transmission (becoming the head of the transmission queue) to the time of successful transfer of its last bit and calculates the average value for the last P packets using (13),

$$\hat{d}_{tr}^i = \frac{\sum_{n=1}^{P} d_{tr}^i(n) \times w_n}{\sum_{n=1}^{P} w_n} \tag{13}$$

where,

$$w_n = 1 - \frac{n - P/2}{P/2 + 1}, \quad P/2 < n \leq P$$

For $P = 8$, this gives weights of 1, 1, 1, 1, 0.8, 0.6, 0.4 and 0.2 for w_1 through w_8, respectively.

Note that d_{tr}^i includes all delays due to media contention, such as channel sensing, RTS/CTS if any (depending on the chosen MAC protocol), backoff time slots, retransmissions, *etc*. Also note that the sensitivity of the average value $\hat{d}_{tr}^i$ depends on the value of P. In practice, a value of $P = 8$, with the most recent four samples equally weighted, appears to be a lower bound that still achieves a reasonable balance between resilience to link variations and fast response to real changes in the network conditions.

Estimation of Link Reliability Each node $i \in \mathcal{N}$ in the MAC layer also measures the average link reliability $\hat{r}_{i,j}$ separately for all of the downstream nodes $j \in \mathcal{D}_i$ using Windowed Mean EWMA (WMEWMA), which is very similar to EWMA but updates the estimated parameter in regular time intervals instead of doing it for every packet. This method is more appropriate for measuring link reliability. WMEWMA counts the total number of transmission attempts $TxCounter$ required (including the retransmissions) to send all of the packets of the current window W and the number of packets successfully transmitted $SucCounter$ of the same window. Therefore, the ratio $\frac{SucCounter}{TxCounter}$ represents the success probability of the link for the window, which we regard as the link reliability

in this paper. This per window link reliability is then averaged with the previous measurements using EWMA as follows,

$$\hat{r}_{i,j} = (1 - \beta) \times \hat{r}_{i,j} + \beta \times \frac{SucCounter}{TxCounter} \tag{14}$$

where β is the moving average smoothing factor, and its value is set equal to 0.4 in our simulation; a higher value is chosen because the current measurement is performed for all of the packets in a window $W = 8$ instead of for a single packet.

4.6. QoS-Aware Queuing and Scheduling

Queuing and scheduling have a direct impact on QoS characteristics. The desired QoS-aware routing for multiple classes of traffic, ranging from most critical data packets having constraints on both the delay and reliability to ordinary packets having no such delay or reliability constraints, is assumed to be met by implementing priority queues. In this work, four separate queues in a sensor node are considered; the highest priority queue for the critical packets (CP), the second higher priority queue for the delay-constrained packets (DP), the next lower priority queue for the reliability-constrained packets (RP) and the least priority queue for the ordinary packets (OP), as illustrated in Figure 2. Here, the scheduler uses strict priority logic, *i.e.*, it always serves the highest priority queue first. If there is no packet waiting in the higher priority queues, it will serve the lower priority queues.

A key problem of the above multi-queuing system is that the lower priority traffic may be indefinitely blocked by higher priority traffic, which is commonly known as the starvation problem. A solution to this starvation problem of the lower priority traffic is *aging*, which is a technique of gradually increasing the priorities of packets waiting in the system for a longer period of time. In this work, we use a timeout-based policy that moves a packet to an upper priority queue on expiration of the timeout. Note that the above multi-queuing system implements the in-node packet contention based on priority levels, *i.e.*, among the packets of the same node. It is also possible to define inter-node priorities for all traffic classes of the neighboring nodes by modifying the MAC protocol slots and backoff times [15]. However, the implementation of such inter-node prioritization requires important modifications in the MAC layer, which is beyond the scope of this work.

5. Performance Evaluation

5.1. Simulation Model and Method

The performance of the proposed DMQoS routing framework is studied and compared with three other localized and QoS-aware routing protocols DARA [15], LOCALMOR [16] and MMSPEED [12] using simulations based on ns-2 [5], which supports the simulation of multihop wireless networks complete with physical, data link, and MAC layer models. The configuration of network parameters is shown in Table 1. The delay and reliability constraints for each packet class are listed in Table 2. Note here that the constraints for a candidate application may vary depending upon the values generated by the sensors. For instance, BP or body temperature readings may produce CP traffic flows if the values cross certain thresholds.

Table 1. Configuration of Parameters.

Deployment	Area	2,000 m × 2,000 m
	Deployment type	Random
	Number of nodes	1,000 BSMs
		6,000 BMSs
	Sink locations (3 sinks)	(1000, 300)
		(200, 1700)
		(1700, 1800)
	Initial node energy	100 Joules
	Buffer size	60
	Radio range	100 m
	Link layer trans. rate	1 Mbps
	Transmit power	$7.214e^{-3}$ Watt
	Rcv. signal threshold	$3.65209e^{-10}$ Watt
	Bit error rate	10^{-4}
Task	Application type	Event-driven
	Packet size	$<= 32$ Bytes
	Traffic type	CBR
MAC	IEEE 802.15.4	Default values
Simulation	Time	1,000 seconds

Table 2. QoS requirements for different applications.

Packet Class	Delay constraint	Reliability constraint	Candidate applications
CP	0.25 s	0.90	ECG, EEG
DP	0.30 s	-	Video imaging, Motion sensing, EMG
RP	-	0.95	BP, PH and respiration monitoring
OP	Only energy-aware		Glucose, SPO2, Body temperature

For performance studies, we used five different sets of source traffic loads S_1 through S_5, listed in Table 3, each consisting of a certain number of traffic flows for each packet classes. For instance, traffic input set S_1 consists of 5 CP traffic flows, 10 DP flows, 20 RP flows and 30 OP flows.

To prevent buffering packets indefinitely, packets are dropped if they wait in the send buffer for more than their remaining lifetimes. All packets sent via the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has a maximum size of 60 packets and is maintained as a priority queue with four priorities, each served in FIFO order. Routing packets get higher priority than data packets.

The average result of 10 simulation runs is calculated for each graph point. For each simulation run, the active nodes are randomly selected, and the source node for each of the flows is also randomly selected. Thus, the variations in the obtained results mainly occur due to the randomness of the topology.

The error bars in the graphs parallel to the y-axis indicate the variations in the obtained results from the presented average values and, thus, show the minimum and maximum values obtained from the runs.

Table 3. Five different sets of source traffic loads.

Source sets / Packet Class	S_1	S_2	S_3	S_4	S_5
CP	5	10	20	30	40
DP	10	20	40	60	80
RP	20	40	60	80	100
OP	30	60	80	100	120

5.2. Simulation Results

We first evaluate the impacts of various traffic loads and bit error rates (BER) on the average end-to-end packet delay and the on-time packet delivery ratio. While the first parameter is measured as the average delay experienced by all classes of delivered packets to the sinks, the latter one is the ratio of the total number of packets received by the sinks to the number of packets generated by the sources.

Impact of Traffic Loads The modular and distributed architecture of the proposed DMQoS framework helps it to achieve significant performance improvements over the state-of-the-art QoS-aware routing protocols, as shown in Figure 3. Recall (from Algorithms 1 and 2) that DMQoS optimizes the number of packet duplications based on measured link reliability ($r_{i,j}$) values and chooses the next-hop router that has the highest velocity toward the destination sink. The velocity prediction of DMQoS is more accurate than those of other methods since it takes into account both transmission and queuing delays at neighbor nodes. Furthermore, the routings of different classes of data packets are handled distinctly, giving highest priority to the critical traffic flows, thus ensuring more reliable and faster delivery of important information to the destination.

As source traffic load increases, the media contention increases as well, which in turn raises packet delays at each hop toward the destination. Figure 3(a) shows that, in all approaches, the end-to-end packet delay rises to higher values with increasing traffic load. The DMQoS offers the lowest end-to-end delay, and the gaps amongst the graphs are widened as traffic volume increases, indicating that DMQoS is more capable of handling high traffic volume compared to other approaches. On the other hand, MMSPEED suffers from the highest delay followed by LOCALMOR and DARA. Through careful examination of the details of our simulation, we observe that mainly exponential increase in packet duplication in MMSPEED, blind duplication of data packets toward primary and secondary sinks in LOCALMOR and the use of the same aggregate routing function both for critical and non-critical traffic in DARA cause more contention in the wireless medium, resulting in increased end-to-end packet delivery delays.

Figure 3(b) shows that DMQoS demonstrates superior performance in terms of on-time packet delivery ratio, indicating that packet drops can greatly be reduced when source data packets are dispersed toward multiple sinks in a way that implements load-balancing among the paths by judiciously choosing

the more reliable links and lightly loaded next-hop router nodes along the path. The simulation results indicate that the poor performances of LOCALMOR and MMSPEED protocols are primarily due to the excessive media contentions caused by huge numbers of duplicate packets in the networks. We also plotted the average end-to-end packet delays and delivery ratio experienced by CP traffic only in Figure 3(c) and Figure 3(d), respectively, and found that our DMQoS outperforms the existing approaches, evidencing the advantage of using the data-centric and scalable modular QoS-aware routing approach.

Figure 3. Performance comparisons for varying traffic loads- **(a)** average end-to-end delay of all data packets, **(b)** on-time packet delivery ratio *i.e.*, the achieved reliability, **(c)** average delay of CP traffic and **(d)** reliability of CP traffic.

(a)

(b)

(c)

(d)
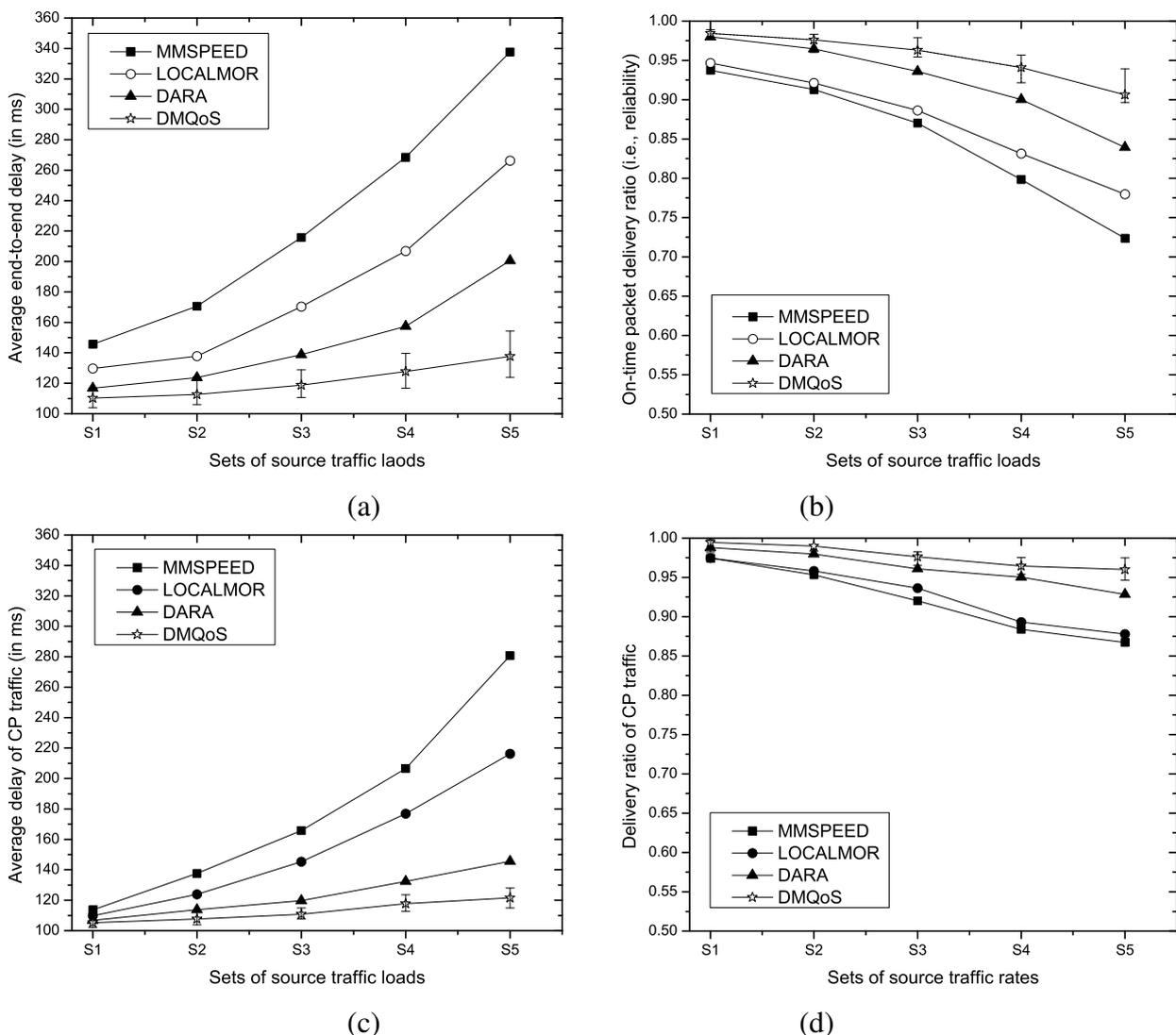
Impact of Bit Error Rate   In what follows, we investigate the impacts of wireless link bit error rates (BER) on the end-to-end packet delay and the reliability. We vary the BER value ranging from 10^{-6} to 10^{-2}, keeping the data traffic load fixed at S_3.

With the increase of bit error rates packet delivery delay increases, and the probability of successful packet delivery decreases. More explicitly, higher value of BER raises the number of retransmissions

required for each packet (at each hop) prolonging the per-hop packet delay and thereby the end-to-end delay. A packet is dropped by an intermediate node if the number of retransmissions exceeds its limit, diminishing the packet delivery ratio. However, by comparing the graphs in Figures 3 and 4, we find that the effect of bit error rate is much higher on the end-to-end delay than that on the packet delivery ratio. The reason is that, at moderate traffic loads, the contention in the wireless medium does not grow up that much, and thus, even though the packet delivery delay increases due to packet retransmissions, it can prevent the effect of bit error rates keeping the packet delivery ratio at an acceptable level.

Figure 4. Performance comparisons for varying bit error rates- **(a)** average end-to-end delay of all data packets, **(b)** on-time packet delivery ratio, **(c)** average delay of CP traffic and **(d)** reliability of CP traffic.

(a)

(b)

(c)

(d)
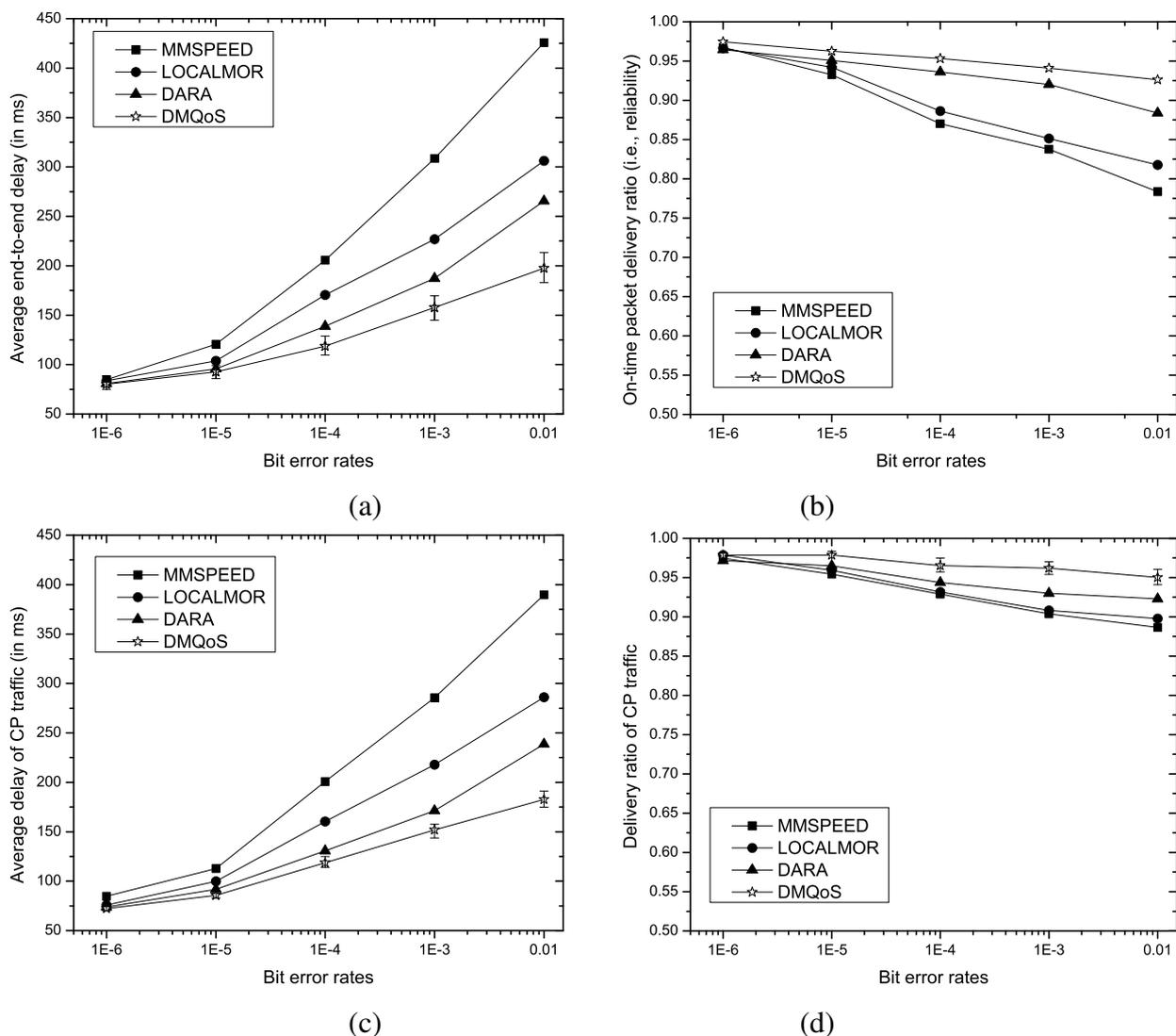
From the results of Figure 4, we also observe that our DMQoS is more robust to the wireless link bit error rates compared to other approaches. This result is characterized by several facts: (i) DARA's aggregate routing metric (*i.e.*, weighted-linear combination of three sub-metrics) fails to select the most appropriate next-hop node for the traffic classes, *i.e.*, it uses the same metric for CP, DP or RP traffic flows, (ii) excessive duplicate packets toward fixed number of sinks generated by LOCALMOR

and MMSPEED cause high media contention and packet drops, and (iii) MMSPEED suffers most since its exponentially generated duplicated packets (along with the original packets) converge together somewhere near the single sink causing many packet drops due to collisions and buffer overflows. However, our DMQoS optimizes the number of duplicate packets, selects the most appropriate next-hop node using separate metrics for different packet classes and disperses the source traffic toward sinks offering lightly-loaded paths.

Impacts of Traffic Load and BER on Energy Consumption   In this experiment, we calculate the total amount of energy consumed for transmission and reception of a packet by source and forwarder nodes until it is received by a sink and average the values for all packets. Figures 5(a) and 5(b) show the average energy consumption per packet for varying source traffic loads and bit error rates in DMQoS and DARA, respectively.

Figure 5. Average energy consumption per packet in **(a)** DMQoS and **(b)** DARA for varying traffic loads and bit error rates.

(a)   (b)
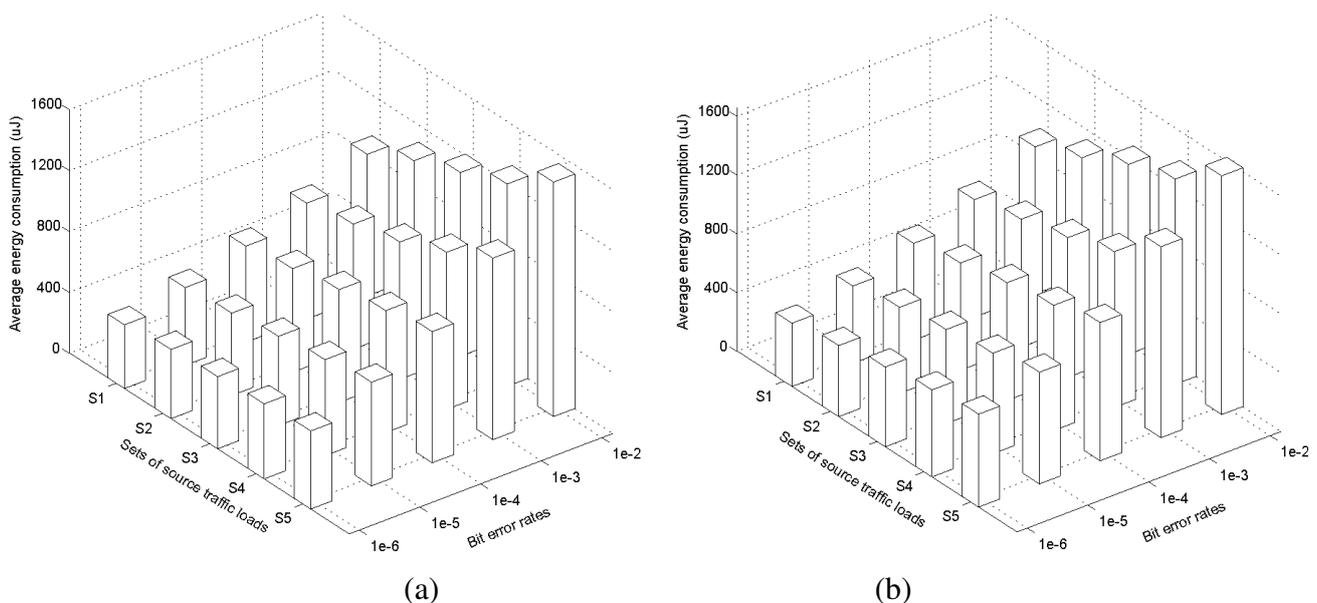
We observe that, in both approaches, the energy consumption rises up linearly for increasing source traffic loads and exponentially for increasing bit error rates. However, the rate of increase in DMQoS is slightly less than that in DARA. Our in-depth look into the simulation results reveals that this is mainly due to the reduced amount of packet collisions and retransmissions in DMQoS compared to those in DARA. As the source traffic load increases, the media contention increases as well and the data packets have to travel longer paths (for load-balancing) to reach their destinations and thus the per packet energy consumption is also increased. BER has more impacts on energy consumption since it forces packets to be retransmitted many times at each hop.

Protocol Operation Overhead   One further issue to consider is the amount of energy overheads due to transmission and reception of routing control packets in different approaches for varying traffic loads and

bit error rates. Every QoS provisioning scheme has to exchange additional control packets (in addition to those for the basic routing mechanism) in order to update the nodes with the current neighborhood information necessary to provide better QoS services, incurring extra overhead.

As expected theoretically, in all approaches, the amount of overhead quickly rises to higher values with increasing numbers of traffic sources. Because each source node needs to exchange routing control packets, the number of such control packets increases with the traffic load of the network as the routing parameters quickly vary. Figure 6(a) shows that MMSPEED incurs the highest overhead since it uses reliability and delay backpressure packets in addition to the neighborhood route information update messages; however, the overheads in the other three approaches are very close to each other as they all use only HELLO (or BEACON) packets to update single-hop neighborhood information. For the same reason, in Figure 6(b), we observe that the overheads in all approaches increase almost linearly with increasing bit error rates.

Figure 6. Protocol operation energy overhead due to routing control packets for **(a)** varying traffic loads and **(b)** bit error rates.

(a)   (b)
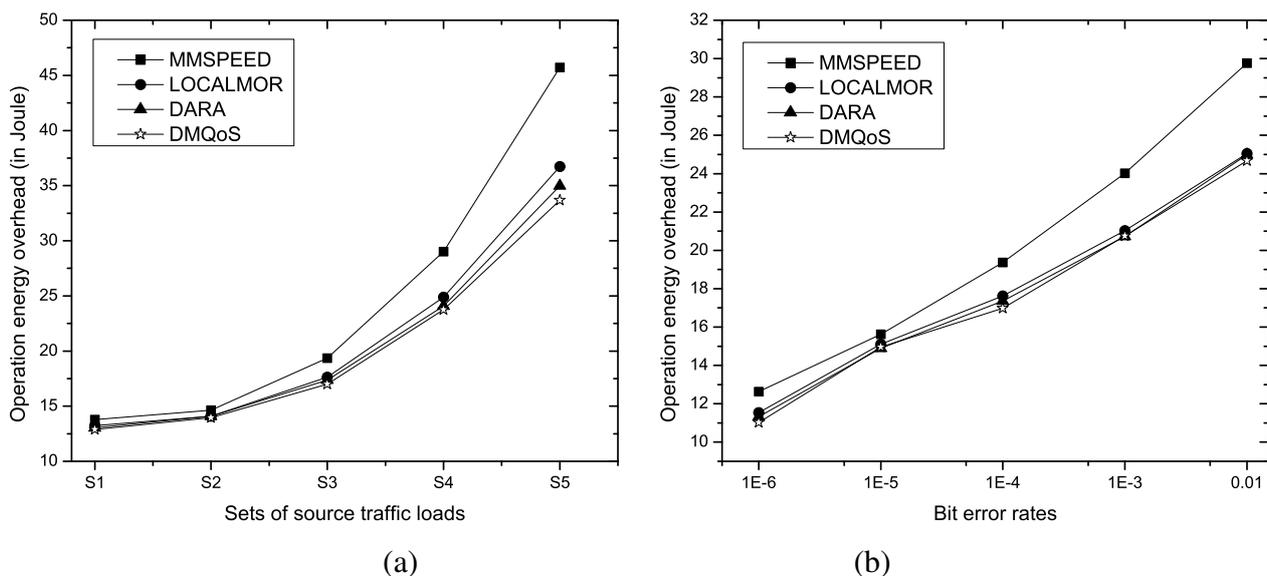
5.3. Discussions

The assembly efficiency of multiple paths is a great boon to unreliable sensor networks. Obviously, there may exist many feasible combinations. To save the energy cost, the set with minimum number of paths is chosen as the forwarding set (see Sections 4.2 and 4.3). We argue that sending a packet on more paths induces more energy cost, because more data packets have to be transmitted. Also, using more paths introduces more contentions which degrades energy efficiency. Even some paths in the set may have more hops, it is still more energy efficient to confine packets to a few paths. For the energy efficiency, as long as the delay and reliability constraints are satisfied we forward the data traffic over energy-rich nodes in order to implement an almost homogeneous energy dissipation rates for all sensor nodes in the network.

The main limitation of this paper is related to a lack of sufficient understanding about the dynamics of several estimation tuning parameters. For example, α, β and w_n values were determined through

numerous simulation experiments. If we could build an analytical model for them, we would be able to dynamically select the optimal values to adapt to different situations. We left it for future work.

6. Conclusions

Due to the overheads caused by implementing QoS in BSNs, QoS and energy must be considered together. In this paper, we have proposed a distributed flexible mechanism to optimize QoS and energy in multihop BSNs based on modular design architecture, following several traffic classes. The routing decision is localized and independent of traffic classes, which distinguishes DMQoS from other approaches. Energy is optimized among the set of candidate nodes offering data-centric QoS services (delay and/or reliability). The results signify that our data-centric multiobjective QoS-aware routing protocol provides a dynamic and modular approach for rendering quality data delivery services and is well suited to resource-constrained BSNs with comparable overhead to those of other methods.

Acknowledgements

References

1. Jovanov, E.; Poon, C.C.Y.; Yang, G.Z.; Zhang, Y.T. Body sensor networks: From theory to emerging applications. *Trans. Info. Tech. Biomed.* **2009**, *13*, 859–863.
2. Patel, M.; Wang, J. Applications, challenges, and prospective in emerging body area networking technologies. *IEEE Wirel. Commun.* **2010**, *17*, 80–88.
3. Hanson, M.; Powell, H.; Barth, A.; Ringgenberg, K.; Calhoun, B.; Aylor, J.; Lach, J. Body Area Sensor Networks: Challenges and Opportunities. *IEEE Trans. Comput.* **2009**, *42*, 58–65.
4. Zykina, A.V. A lexicographic optimization algorithm. *Autom. Remote Control* **2004**, *65*, 363–368.
5. Information Sciences Institute. NS-2 Network Simulator. Software Package, 2003.
6. Ameen, M.A.; Nessa, A.; Kwak, K.S. QoS Issues with Focus on Wireless Body Area Networks. In *Proceedings of ICCIT08: The 2008 Third Intl Conference on Convergence and Hybrid Information Technology*, Washington, DC, USA, 11–13 November 2008; pp. 801–807.
7. Liang, X.; Balasingham, I. A QoS-aware Routing Service Framework for Biomedical Sensor Networks. In *Proceedings of the IEEE Intl Symposium on Wireless Communication Systems (ISWCS)*, Trondheim, Norway, 17–19 October 20; pp. 342–345.
8. Chen, M.; Kwon, T.; Choi, Y. Energy-efficient differentiated directed diffusion (EDDD) in wireless sensor networks. *Comput. Commun.* **2006**, *29*, 231–245.
9. Liang, X.; Balasingham, I.; Byun, S.S. A reinforcement learning based routing protocol with QoS support for biomedical sensor networks. In *Proceedings of ISABEL 08. First International Symposium on Applied Sciences on Biomedical and Communication Technologies, 2008*, Aalborg, Dennmark, 25–28 October 2008; pp. 1–5.

10. Chen, M.; Leung, V.C.M.; Mao, S.; Yuan, Y. Directional geographical routing for real-time video communications in wireless sensor networks. *Comput. Commun.* **2007**, *30*, 3368–3383.

11. Huang, X.; Fang, Y. Multiconstrained QoS multipath routing in wireless sensor networks. *Wirel. Netw.* **2008**, *14*, 465–478.

12. Felemban, E.; Lee, C.G.; Ekici, E. MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* **2006**, *5*, 738–754.

13. Chen, M.; Leung, V.; Mao, S.; Xiao, Y.; Chlamtac, I. Hybrid Geographic Routing for Flexible Energy Delay Tradeoff. *IEEE Trans. Veh. Technol.* **2009**, *58*, 4976–4988.

14. Chen, M.; Kwon, T.; Mao, S.; Yuan, Y.; Leung, V.C.M. Reliable and energy-efficient routing protocol in dense wireless sensor networks. *Int. J. Sen. Netw.* **2008**, *4*, 104–117.

15. Razzaque, M.A.; Alam, M.M.; Rashid, M.M.; Hong, C.S. Multi-Constrained QoS Geographic Routing for Heterogeneous Traffic in Sensor Networks. *IEICE Trans. Commun.* **2008**, *E91-B*, 2589–2601.

16. Djenouri, D.; Balasingham, I. New QoS and geographical routing in wireless biomedical sensor networks. In *Proceedings of Sixth International Conference on Broadband Communications, Networks, and Systems, 2009. BROADNETS 2009*, Madrid, Spain, 14–16 September 2009; pp. 1–8.

17. Chen, M.; Gonzalez, S.; Vasilakos, A.; Cao, H.; Leung, V.C.M. Body Area Networks: A Survey. *ACM/Springer Mob. Netw. Appl.* **2011**, doi: 10.1007/s11036-010-0260-8.

18. Chen, M.; Gonzalez, S.; Leung, V.; Zhang, Q.; Li, M. A 2G-RFID-based e-healthcare system. *IEEE Wirel. Commun.* **2010**, *17*, 37–43.

19. Lee, S.; Bhattacharjee, B.; Banerjee, S. Efficient geographic routing in multihop wireless networks. In *Proceedings of MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, Urbana-Champaign, IL, USA, 25–28 May 2005; pp. 230–241.

20. Floyd, S.; Handley, M.; Padhye, J.; Widmer, J. Equation-based congestion control for unicast applications. In *Proceedings of SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Stockholm, Sweden, 28 August–1 September 2000; pp. 43–56.