Research Article

QPRD: QoS-Aware Peering Routing Protocol for Delay-Sensitive Data in Hospital Body Area Network

Zahoor A. Khan,[1] Shyamala Sivakumar,[2] William Phillips,[3] Bill Robertson,[3] and Nadeem Javaid[4]

[1]*CIS, Higher Colleges of Technology, Fujairah Campus, P.O. Box 4114, Fujairah, UAE*
[2]*Saint Mary's University, Halifax, NS, Canada B3H 3C3*
[3]*Internetworking Program, FE, Dalhousie University, Halifax, NS, Canada B3H 4R2*
[4]*COMSATS Institute of Information Technology, Islamabad 44000, Pakistan*

Correspondence should be addressed to Zahoor A. Khan; zahoor.khan@dal.ca

Consistent performance, energy efficiency, and reliable transfer of data are critical factors for real-time monitoring of a patient's data, especially in a hospital environment. In this paper, a routing protocol is proposed by considering the QoS requirements of the Body Area Network (BAN) data packets. A mechanism for handling delay-sensitive packets is provided by this protocol. Moreover, linear programming based modeling along with graphical analysis is also done. Extensive simulations using the OMNeT++ based simulator Castalia 3.2 illustrate that the proposed algorithm provides better performance than other QoS-aware routing protocols in terms of higher successful transmission rates (throughputs), lower overall network traffic, no packets dropped due to MAC buffer overflow, and fewer numbers of packet time outs in both the mobile and static patient scenarios. The scalability of the protocol is demonstrated by simulating a 24-bed real hospital environment with 49 nodes. It is shown that, even in the larger real hospital scenario requiring the transmission of delay-sensitive data packets with stringent delay requirements, QPRD outperforms comparable protocols.

1. Introduction

A patient's real-time health-related data monitoring is possible with the help of a new emerging field, Body Area Networks (BANs). Body Area Network is a small wireless network which consists of sensors placed inside or outside of the human body. The body implant or wearable sensors transmit the data to a central device called Body Area Network Coordinator (BANC). BANC is computationally more powerful device then the body sensors. BANC is responsible for transferring the sensors' data to the next node or destination reliably.

Some important issues of BAN data transmission are to ensure the high reliability, low latency, compatibility with movable sensors, and low energy consumption. The specific need of BAN communication is not fulfilled by the existing Personal Area Network (PAN) standards [1]. IEEE task group 6 was assigned a job in November 2007 to suggest a BAN

communication standard 802.15.6 by the consideration of short range transmission, reliability and latency requirements of QoS, and less energy consumption [2]. The real-time monitoring of patients requires the transmission of delay-sensitive data such as video imaging, motion sensing, and Electromyography (EMG) using BAN. Some projects like SMART [3], CareNet [4], AID-N [5], and ALARM-NET [6] provide different methods to monitor the patient data. In these methods, the transmission of BAN data from body sensors to the central database is considered and then BAN data is downloaded and monitored from the central database. However, these techniques do not monitor or display in real-time BAN data in hospital environment. The advantages of using a centralized system are to have better control and maintain the data privacy of the patient. However, traffic congestion, server failure, or link failure can cause considerable delays in monitoring the patient data which can badly affect treatment. On the other hand, distributed

data approaches help to reduce the traffic load and can better accommodate patient mobility. The ZK-BAN peering framework proposed in [7] suggests a semicentralized system for reliably monitoring BAN data. The hybrid ZK-BAN uses both centralized and distributed techniques.

The routing protocols EPR, proposed and discussed in [7], resolves the problem of handling ordinary data packets. The QoS-aware peering routing protocol for reliability-sensitive data (QPRR) [8, 9] provides a mechanism of handling the reliability-sensitive packets in addition to the ordinary data packets. The requirement of real-time display for delay-sensitive packets is different from those of ordinary and reliability-sensitive packets. Hence, a new QoS-aware routing protocol is required to handle delay-sensitive packets. A novel routing protocol that addresses the issue of handling delay-sensitive data and displaying in real-time delay-sensitive BAN data is proposed in this paper. The proposed QoS-aware peering routing protocol for delay-sensitive packets (QPRD) is designed for the ZK-BAN peering framework discussed in [7]. QPRD provides an innovative approach to the reliable transmission of ordinary packets (OPs) and delay-sensitive packets (DSPs). The initial results and architecture of QPRD were presented in IEEE conference proceedings [10].

This paper is organized as follows: Section 2 provides the related work; Section 3 discusses the problem formulation and modeling; Section 4 provides the proposed QoS-aware peering routing protocol for delay-sensitive data (QPRD); Section 5 describes the performance evaluation; Section 6 discusses the scalability test of QPRD and Section 7 presents the conclusions.

2. Related Work

A smart monitoring system of BAN data in hospital environment can resolve the challenges related to the management of patients' medical information [11]. The Scalable Medical Alert and Response Technology (SMART) [3] is designed to monitor the patient's data in hospital emergency area. The data from sensors is transferred to the PDA and then the PDA sends it to the next tier by using wireless standard 802.11b. CareNet [4] provides an integrated wireless sensor based solution to monitor the patient's data from remote hospitals. The two-tier wireless communication is used in the projects [3, 4]. A GPS system is used in [5] to monitor the patient's data only in outdoor BAN communication. A wireless sensor network for assisted-living and residential monitoring system with a query based protocol is provided in ALARM-NET [6]. A three-tier communication approach is used in [12] to store the BAN data on the server and then make this data available for the physician to analyze the patient's data. The projects [3–6, 12] used a centralized approach to monitor the patient's data. However, the real-time display of data by considering the delay requirements of delay-sensitive packets is not considered. To access the data from a centralized server may cause delay and even a simple link failure can completely disconnect the healthcare system from the central server.

In [7], an energy-aware peering routing protocol (EPR) was presented which considers the energy level and geographic information of the neighbor nodes for choosing the best next hop. The EPR only considers ordinary packets. It was shown that EPR has an overall lower energy consumption than comparable protocols [11, 13–16] and provides better results in terms of reduced overall network traffic, reduced number of packets forwarded by intermediate nodes, and higher successful data transmission rates. However, EPR does not provide a mechanism for dealing with delay-sensitive packets (DSPs). In this paper, delay-sensitive packets are considered by the proposed QoS-aware peering routing protocol for delay-sensitive data (QPRD) and their performance is investigated by comparing it to the existing DMQoS protocol [13]. In [13], DMQoS categorizes the data packets into four types: ordinary packets (OPs), critical packets (CPs), reliability-driven packets (RPs), and delay-driven packets (DPs). The DMQoS [13] provides better results for delay-driven packets than several previously investigated methods [11, 14–16] in terms of end-to-end path delay. However, DMQoS employs a *hop-by-hop* approach to determine the next hop. DMQoS considers the neighbor device with the lowest delay, and the next hop then determines the best next upstream hop with least delay. The disadvantage of this *hop-by-hop* delay-driven approach employed in DMQoS is that only neighboring nodes delay information is considered by source node. The source node forwards the packet to a particular neighbor node which has lower node delay than the required delay. The neighbor node sends the acknowledgement of the successfully received packet to the source node. Now, the packet receiving neighbor node determines its best upstream node in terms of delay requirement and forwards the packet to the upstream node if the node delay of upstream node is less than the required delay. In case the neighbor node does not find any upstream node with node delay less than required delay, then the packet is dropped. In this case, the packet does not reach the destination, but the source node assumes that the packet has been successfully received by the destination. Furthermore, the *hop-by-hop* approach used in DMQoS causes an increase in overall network traffic, and the required end-to-end latency may not be guaranteed. In this paper, the proposed QPRD addresses these shortcomings by selecting and choosing the next hop device based on the lowest end-to-end path delay from the source node to the destination.

3. Problem Formulation and Modeling

The motivation that BAN consists of nodes connected with each other via wireless links leads us to model it as a directed graph. This section focuses on two points: (i) to maximize throughput, and (ii) to minimize the end-to-end delay. These two problems are modeled via linear programming [17, 18] because requirements to these problems could easily be represented by linear relationships.

3.1. Throughput Maximization. We consider BAN as a directed graph $G = (S, L)$, $|S| = s$, and $|L| = l$; S is the set of nodes and L is the set of directed graphs (links). If the network operations are divided into rounds, each round r is the duration from the network establishment till the death

of all nodes; then linear programming based mathematical formulation for throughput maximization is as follows:

$$\text{Max} \sum_r T(r), \quad \forall r \in R, \tag{1}$$

where

$$T(r) = \sum_i l_{(i,\text{Dst})} \cdot T_{(i,\text{Dst})} \quad \forall i \in S, \tag{2}$$

$$l_{(i,\text{Dst})} = \begin{cases} 1 & \text{if packet delivery is guaranteed} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

such that

$$B \leq B_{\max}, \tag{4a}$$

$$\sum_i E_i \leq E_0, \quad \forall i \in S, \tag{4b}$$

$$\sum_i \text{DL}_{\text{path}(i,\text{Dst})} \leq \sum_i t_{\text{out}(i,\text{Dst})}, \quad \forall i \in S, \tag{4c}$$

$$\sum_i d_{(i,\text{Dst})} \leq \sum_i R_i^{tx}, \quad \forall i \in S, \tag{4d}$$

$$\sum_i f_{(i-1,i)} + \sum_i \lambda_i t \leq \sum_i f_{(i,\text{Dst})}, \quad \forall i \in S. \tag{4e}$$

The objective function in (1) aims to maximize throughput T during each round r such that (2) associates packet delivery from source i to destination Dst with link flag l. Equation (3) provides details about the status of l being raised ($l = 1$) if packet delivery through that link is guaranteed else not ($l = 0$). Constraint in (4a) provides the upper bound for the allocated bandwidth B as $B_{\max}$. Similarly, constraint in (4b) deals with limited energy constraint; that is, each node i is equipped with an energy source E_i such that $\sum_i E_i$ is upper bounded by E_0. Node ceases transmission whenever its battery is drained out, so, energy efficient utilization is very important (routing and MAC layer protocols play a critical role here). Constraint in (4c) comes into consideration if and only if $\exists (m/P) \in P_{\text{QoS}}$; m path(s) out of total P satisfies the given quality of service P_{QoS} where DL_{path} is the end-to-end delay and t_{out} is the timeout period. This means that, as a first priority, QoS needs to be satisfied. Afterwards, if there is more than one QoS path, then as a second priority end-to-end delay is checked. Transmission range R^{tx} constraint in (4d) demonstrates that packet delivery is successful if a source node transmits data to an in-range destination node where $d_{(i,\text{Dst})}$ is the distance between source and destination. For data generation rate λ, (4e) constraint entails flow conservation such that the incoming data flow $f_{(i-1,i)}$ plus the data generated during time t should not exceed the outgoing data flow $f_{(i,\text{Dst})}$. Violation of (4c), (4d), and (4e) leads to packets being dropped which ultimately results in decreased throughput.

3.2. Delay Minimization. The delay minimization problem, while routing dynamically such that path for each request is selected to prevent routing latency for future demands, is addressed here. The linear programming problem is formulated as follows:

$$\text{Min} \sum_i \text{DL}_{\text{path}(i,\text{Dst})}, \quad \forall i \in S, \tag{5}$$

where

$$\text{DL}_{\text{path}(i,\text{Dst})} = \begin{cases} \text{DL}_{\text{node}(i)} + \text{DL}_{\text{path}(j,\text{Dst})} & j \neq \text{Dst} \\ \text{DL}_{\text{node}(i)} & j = \text{Dst}, \end{cases} \tag{6}$$

$$\text{DL}_{\text{node}(i)} = \text{DL}_{\text{trans}(i)} + \text{DL}_{\text{queue+channel}} + \text{DL}_{\text{proc}}, \\ \forall i \in S \tag{7}$$

such that

$$l_{\text{path}(i,\text{Dst})} = 1, \quad \forall i \in S, \tag{8a}$$

$$R_{\text{bit}} \leq R_{\text{bit}}^{\max}, \tag{8b}$$

$$\sum_{t=1}^{t_{\max}} n \leq n_{\text{cap}}, \tag{8c}$$

$$\sum_i \lambda_i^{\text{arrival}} < \sum_i \lambda_i^{\text{departure}}, \quad \forall i \in S, \tag{8d}$$

$$0 \leq \sum_i \text{node}(i) \leq s, \quad \forall i \in S, \tag{8e}$$

$$N_{\text{bit}}^e \leq N_{\text{bit}}^{\text{proc}}. \tag{8f}$$

The objective function in (5) aims to minimize the end-to-end path delay $\text{DL}_{\text{path}(i,\text{Dst})}$, where (6) depicts the two possible cases: communication via intermediate node and without intermediate node, and (7) defines the node delay $\text{DL}_{\text{node}(i)}$ calculated at the network layer as the addition of packet delays due to transmission $\text{DL}_{\text{trans}(i)}$, queuing DL_{queue}, channel capturing $\text{DL}_{\text{channel}}$, and processing DL_{proc}. Constraint (8a) clearly says that the link through which data is routed must be established where l_{path} is the link flag. Constraint in (8b) provides the upper bound of data rate R_{bit} as $R_{\text{bit}}^{\max}$ such that R_{bit} is inversely proportional to DL_{trans} according to (15) explained in later Section 4.4. Constraint in (8c) says that the number of transmitted packets n in 4 seconds ($t_{\max} = 4 \text{ sec}$) should not exceed the packet handling capacity n_{cap} because, at negligible load, there is constant small delay. However, queuing delays on each node are added as soon as the network load increases and when n exceeds n_{cap} delays increase without bound. Similarly, constraint in (8d) deals with queue stability meaning that the packet arrival rate $\lambda_i^{\text{arrival}}$ should be always less than the packet departure rate $\lambda_i^{\text{departure}}$. Violation of (8d) results in nonavailability of buffer space leading to queue instability which in turn leads to congestion and thus causing increased delay. Constraint in (8e) states that the total number of nodes $\sum_i \text{node}(i)$ in the given network is limited such that i has an inverse relation with $\text{DL}_{\text{channel}}$. In other words, increasing the number of nodes means that there are more chances that one of the noncapturing nodes might have

a lower backoff time as compared to that of the capturing node, thereby increasing the idle time due the backing off of noncapturing nodes. Constraint in (8f) deals with DL_{proc}. It is obvious that the total bit level errors N_{bit}^e should not exceed a certain threshold N_{bit}^{proc}; otherwise increased erroneous bits would increase DL_{proc} and performance of the network would degrade in terms of processing delays at the nodes.

3.3. Graphical Analysis. Consider the simplified path B_3-B_1-NSC where B_3 can directly send data to NSC as well as via B_1 (refer to Figure 3 in Section 4.4). Moreover, NSC can also send data to itself. For typically assumed delay values, path delay is maximum when B_1 is involved in forwarding the data of B_3 intended for the destination NSC (i.e., 50 ms), and path delay is minimum when B_3 directly communicates with NSC (i.e., 20 ms). Let $z = DL_{path(i,Dst)}$, $x = DL_{node(i)}$, and $y = DL_{path(j,Dst)}$. The objective function in (5) can be reformulated as

$$\text{Min}\,(z),\tag{9}$$

where

$$z = x + y\tag{10}$$

such that

$$x + y \geq 20,\tag{11a}$$

$$x + y \leq 50,\tag{11b}$$

$$0 \leq x \leq 20,$$
$$0 \leq y \leq 30.\tag{11c}$$

The objective function in (9) aims to minimize end-to-end delay regarding the selected path, whereas (10) illustrates nature of the objective function, that is, two-dimensional linear programming problem. Constraints in (11a) and (11b) provide lower and upper bounds for the selected path, respectively, whereas constraint in (11c) deals with similar bounds for x as well as y. For simplicity in calculation, we replace the inequalities in (11a), (11b), and (11c) with equalities, respectively. In subject to the given constraints, Figure 1 shows the set of feasible solutions which is obtained by the intersection of lines, L_1, L_2, L_3, and L_4, and is indicated by coloured region such that each point in the feasible region satisfies each constraint. We can find the minimum value of z by testing it at each of the vertices (refer to $P1$, $P2$, $P3$, and $P4$ in Figure 1) as follows:

at $P1(0,0)$: $z = 0$ ms,

at $P2(20,0)$: $z = 20 + 0 = 20$ ms,

at $P3(0,30)$: $z = 0 + 30 = 30$ ms,

at $P4(20,30)$: $z = 20 + 30 = 50$ ms.

The minimum value of z is 0 ms at $x = 0$ and $y = 0$. However, this value indicates self transmission or communication within NSC. The next minimum value is $z = 20$ ms showing the case of direct communication
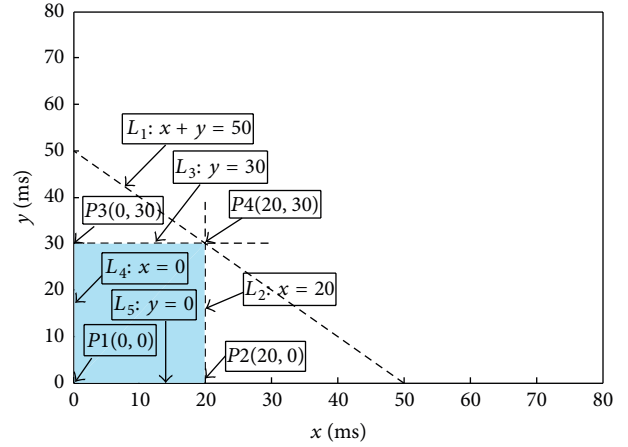

FIGURE 1: Feasible region.

between B_3 and NSC. Similarly, $z = 30$ ms when B_1 directly communicates with NSC. On the other hand, end-to-end path delay is maximum when B_3 communicates with NSC via an intermediate B_1; $z = 50$ ms.

4. QoS-Aware Peering Routing Protocol for Delay-Sensitive Data (QPRD)

Based on the mathematical analysis in Section 3, the proposed QoS-aware routing protocol used in an indoor hospital ZK-BAN peering framework [7] is discussed in this section. The proposed QPRD provides a mechanism to (1) calculate the node delays and path delays of all possible paths from the source node to the destination, (2) determine the best path, and (3) choose the best next hop NH_D based on the delay requirements of the packet. For each destination, the routing table contains information about the next hop device connected to the path with the least end-to-end latency. For any DSP, if the path delay ($DL_{path(i,Dst)}$) is less than or equal to the delay requirement, the source node sends the DSP through that path.

The architecture of proposed QPRD, based on the mathematical formulation of the end-to-end path delay problem, is shown in Figure 2. It consists of seven modules: MAC receiver, delay module (DM), packet classifier (PC), Hello protocol module (HPM), routing services module (RSM), QoS-aware queuing module (QQM), and MAC transmitter. The modules are discussed below.

4.1. MAC Receiver. The MAC receiver receives the data or Hello packets from other nodes (BAN, MDC, or NSC). It checks the MAC address of the packet. It only forwards the broadcast packets or the packets which have the same node's MAC address as destination address to the network layer.

4.2. Delay Module (DM). The delay module monitors the time required to capture the channel ($DL_{channel(i)}$), MAC layer queuing delay ($DL_{MAC\_queue(i)}$), and transmission time
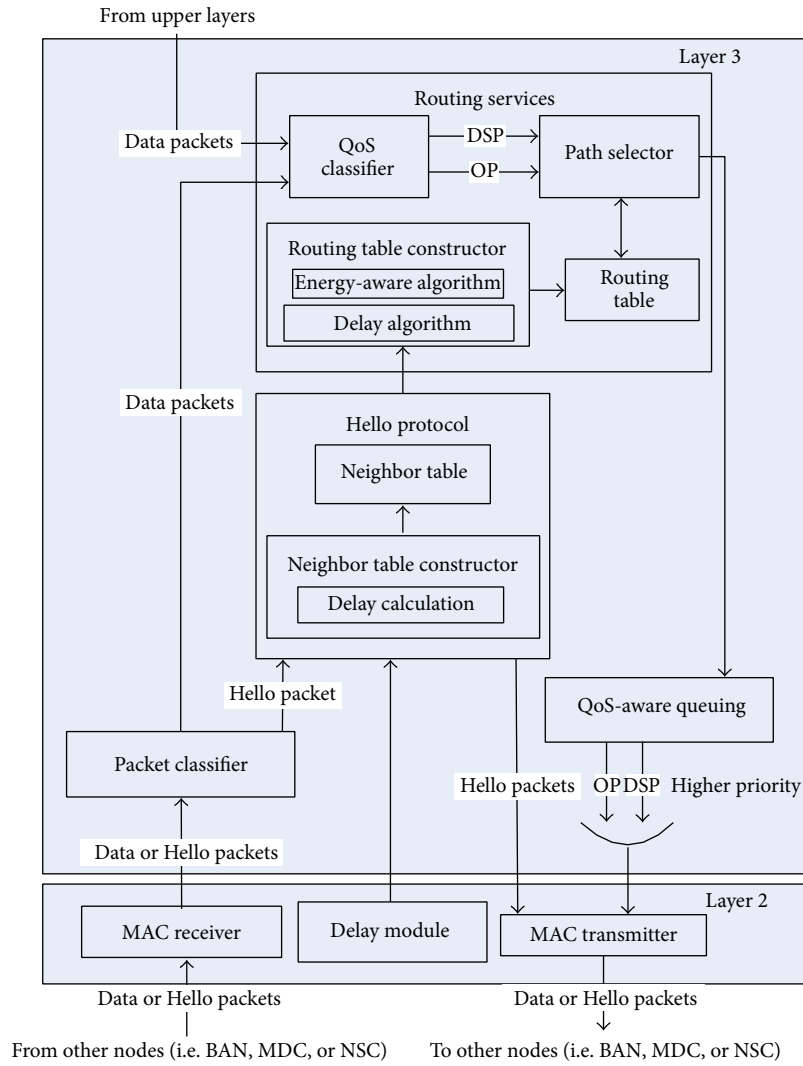
FIGURE 2: QPRD protocol architecture.

($DL_{trans(i)}$) of a packet. The delay module sends this information to the network layer. The network layer uses this information to calculate the node delay ($DL_{node(i)}$).

4.3. Packet Classifier (PC). The packet classifier (PC) receives all the packets from the MAC receiver. The data packets and Hello packets are differentiated by the PC. The PC forwards the data and Hello packets to the routing services module and Hello protocol module, respectively.

4.4. Hello Protocol Module (HPM). The neighbor table constructor and the neighbor table are the two submodules of Hello protocol module. The information received from the delay module of the MAC layer and the Hello packets is used by the neighbor table constructor to construct the neighbor table. Initially, Hello packets are broadcasted by each of type 1 (NSC) and type 2 (MDC) devices. The node i receives the Hello packet. The neighbor table constructor of node i

calculates its own $DL_{path(i,Dst)}$ based on the information in the Hello packets. The Hello packet is updated and forwarded by node i to the other nodes. The Hello packet fields of node j are shown as follows.

Hello Packet Structure. Consider

$$ \boxed{\text{ID}_{\text{Dst}} \mid L_{\text{Dst}} \mid \text{ID}_j \mid L_j \mid D_{(j,\text{Dst})} \mid E_j \mid T_j \mid \text{DL}_{\text{path}(j,\text{Dst})}} \quad (12) $$

The commonly used notations in this paper and their descriptions are summarized in notations section.

The neighbor table contains fields for both hop-by-hop delay ($DL_{node(i)}$) and end-to-end path delay ($DL_{path(i,Dst)}$). The neighbor table constructor updates the neighbor table periodically after receiving every new Hello packet. The neighbor table structure of node i is shown as follows.

The Neighbor Table Structure. Consider

$$ \boxed{\text{ID}_{\text{Dst}} \mid L_{\text{Dst}} \mid \text{ID}_j \mid L_j \mid D_{(j,\text{Dst})} \mid D_{(i,j)} \mid C_j \mid T_j \mid \text{DL}_{\text{node}(i)} \mid \text{DL}_{\text{path}(i,\text{Dst})}} \quad (13) $$

The node delay ($DL_{node(i)}$) can be found by adding the packet delays due to transmission, queuing, processing, and channel capturing:

$$DL_{node(i)} = DL_{trans(i)} + DL_{queues+channel} + DL_{proc}. \qquad (14)$$

The node updates its Hello packets periodically; 4 seconds are used in QPRD for simulation purposes. The time interval 4 seconds is used because the delay module sends the delays of MAC queue and channel capture after every 4 seconds. The average transmission delay (DL_{trans}) before sending the Hello packets is calculated by using

$$DL_{trans} = \frac{1}{R_{bit}} \frac{\sum_{z=1}^{n} N_{bit(z)}}{n}, \qquad (15)$$

where R_{bit} is the data rate and as per BAN requirement 250 kbps is used in the simulations. N_{bit} is the total number of bits in each packet. n is the number of packets transmitted in 4 seconds.

The delay due to the MAC and network layers' queues and capturing the channel can be calculated by using the Exponentially Weighted Moving Average (EWMA) formula and is given in

$$DL_{queues+channel} = (1 - \rho) * DL_{queues+channel} + \rho \\ * DL_{queues+channel}, \qquad (16)$$

where queues are both network and MAC layers' queues.

Initial values of $DL_{queues+channel}$ are the delay of the first packet sent by the node. ρ is the average weighting factor that satisfies $0 < \rho \leq 1$. The selection of ρ value is heuristic and was chosen based on simulations experience. The recommended values are $0.2 \leq \rho \leq 0.3$. The best suited value of ρ found for QPRD simulations is 0.2.

The path delay between node i and destination node Dst ($DL_{path(i,Dst)}$) is calculated by using

$$DL_{path(i,Dst)} = DL_{node(i)} + DL_{path(j,Dst)}, \qquad (17)$$

where initial value of $DL_{path(j,Dst)}$ is zero when j = Dst.

An example of finding the path delay from node i (B_3) to Dst (NSC) is shown in Figure 3. The delay calculation of two paths B_3-B_1-MDC_2-NSC (path 1) and B_3-MDC_3-B_2-MDC_1-NSC (path 2) is given for illustrative purposes. The typical assumed values are chosen for illustrated purposes. The individual node delays used in this example are given below:

$$DL_{node(NSC)} = 20 \, ms, \qquad (18a)$$

$$DL_{node(MDC_2)} = 40 \, ms, \qquad (18b)$$

$$DL_{node(B_1)} = 30 \, ms, \qquad (18c)$$

$$DL_{node(B_3)} = 20 \, ms, \qquad (18d)$$

$$DL_{node(MDC_1)} = 20 \, ms, \qquad (18e)$$

$$DL_{node(B_2)} = 30 \, ms, \qquad (18f)$$

$$DL_{node(MDC_3)} = 10 \, ms. \qquad (18g)$$

The path delay of destination ($DL_{path(Dst,Dst)}$) is approximately zero, because the time required to receive the packet from MAC to network layer is negligible. So, in this example initial path delay is given below:

$$DL_{path(NSC,NSC)} = 0 \, ms. \qquad (19)$$

Each node calculates the path delay from itself to the NSC. First, the calculations of the path delay for path 1 (B_3-B_1-MDC_2-NSC) are considered.

The path delay of MDC_2 ($DL_{path(MDC_2,NSC)}$) is calculated by using (17):

$$DL_{path(MDC_2,NSC)} = DL_{node(MDC2)} + DL_{path(NSC,NSC)}. \qquad (20)$$

Using the values from (18a) and (19) in the above equation, we get

$$DL_{path(MDC_2,NSC)} = 40 + 0 = 40 \, ms. \qquad (21)$$

The path delay of BAN B_1 is calculated below:

$$DL_{path(B_1,NSC)} = DL_{node(B_1)} + DL_{path(MDC_2,NSC)}, \\ DL_{path(B_1,NSC)} = 30 + 40 = 70 \, ms. \qquad (22)$$

The node B_3 determines the path delay by using the values from (18d) and (22):

$$DL_{path(B_3,NSC)} = DL_{node(B_3)} + DL_{path(B_1,NSC)}, \\ DL_{path(B_3,NSC)} = 20 + 70 = 90 \, ms. \qquad (23)$$

In the same manner, the path delay of path 2 (B_3-MDC_3-B_2-MDC_1-NSC) can be calculated as follows:

$$DL_{path(MDC_1,NSC)} = 20 + 0 = 20 \, ms,$$

$$DL_{path(B_2,NSC)} = 30 + 20 = 50 \, ms,$$

$$DL_{path(MDC_3,NSC)} = 10 + 50 = 60 \, ms, \qquad (24)$$

$$DL_{path(B_3,NSC)} = 20 + 60 = 80 \, ms.$$

Equations (23) and (24) show that the path delays of path 1 and path 2 are 90 ms and 80 ms, respectively. It is quite possible that the path with less delay is longer (has more hops) than the other paths. As it is observed from the above example, path 2 includes five devices and path 1 has four devices. However, the path delay of path 2 is lower than the path delay of path 1.

4.5. Routing Services Module (RSM). The routing services module is responsible for constructing the routing table, categorizing the data packets into delay-sensitive packets (DSPs) and ordinary packets (OPs). It also chooses the best path(s) for each category (DSPs or OPs) of traffic. QoS classifier, routing table constructor, path selector, and routing table are the submodules of routing services module. The routing table structure for node i is shown as follows.

The Routing Table Structure for QPRD. Consider

$$\boxed{ID_{Dst} \mid L_{Dst} \mid NH_E \mid NH_D \mid DL_{path(i,Dst)}} \qquad (25)$$

$$\mathrm{DL}_{\mathrm{path}(B_3,\mathrm{NSC})} = \mathrm{DL}_{\mathrm{node}(B_3)} + \mathrm{DL}_{\mathrm{path}(B_1,\mathrm{NSC})}$$
$$= 20\,\mathrm{ms} + 70\,\mathrm{ms} = 90\,\mathrm{ms}$$

$$\mathrm{DL}_{\mathrm{path}(B_3,\mathrm{NSC})} = \mathrm{DL}_{\mathrm{node}(B_3)} + \mathrm{DL}_{\mathrm{path}(\mathrm{MDC}_3,\mathrm{NSC})}$$
$$= 20\,\mathrm{ms} + 60\,\mathrm{ms} = 80\,\mathrm{ms}$$

$$\mathrm{DL}_{\mathrm{path}(B_3,\mathrm{NSC})} = \mathrm{DL}_{\mathrm{node}(\mathrm{MDC}_3)} + \mathrm{DL}_{\mathrm{path}(B_2,\mathrm{NSC})}$$
$$= 10\,\mathrm{ms} + 50\,\mathrm{ms} = 60\,\mathrm{ms}$$

$$\mathrm{DL}_{\mathrm{path}(B_1,\mathrm{NSC})} = \mathrm{DL}_{\mathrm{node}(B_1)} + \mathrm{DL}_{\mathrm{path}(\mathrm{MDC}_2,\mathrm{NSC})}$$
$$= 30\,\mathrm{ms} + 40\,\mathrm{ms} = 70\,\mathrm{ms}$$

$$\mathrm{DL}_{\mathrm{path}(B_2,\mathrm{NSC})} = \mathrm{DL}_{\mathrm{node}(B_2)} + \mathrm{DL}_{\mathrm{path}(\mathrm{MDC}_1,\mathrm{NSC})}$$
$$= 30\,\mathrm{ms} + 20\,\mathrm{ms} = 50\,\mathrm{ms}$$

Path1
$$\mathrm{DL}_{\mathrm{path}(\mathrm{MDC}_2,\mathrm{NSC})} = \mathrm{DL}_{\mathrm{node}(\mathrm{MDC}_2)} + \mathrm{DL}_{\mathrm{path}(\mathrm{NSC},\mathrm{NSC})}$$
$$= 40\,\mathrm{ms} + 0\,\mathrm{ms} = 40\,\mathrm{ms}$$

Path2
$$\mathrm{DL}_{\mathrm{path}(\mathrm{MDC}_1,\mathrm{NSC})} = \mathrm{DL}_{\mathrm{node}(\mathrm{MDC}_1)} + \mathrm{DL}_{\mathrm{path}(\mathrm{NSC},\mathrm{NSC})}$$
$$= 20\,\mathrm{ms} + 0\,\mathrm{ms} = 20\,\mathrm{ms}$$

$$\mathrm{DL}_{\mathrm{path}(\mathrm{NSC},\mathrm{NSC})} = 0$$
$$\mathrm{DL}_{\mathrm{node}(\mathrm{NSC})} = 20\,\mathrm{ms}$$

FIGURE 3: Example of finding the path delay.

The notations and their descriptions are listed in notation section. Two next hop entries NH_E and NH_D are given for each destination Dst in routing table. The routing table constructor contains the energy-aware and delay algorithms. The energy-aware algorithm discussed in [7] is used to find next hop NH_E for OPs. Residual energy and geographic location of the neighbor nodes are considered for choosing NH_E. For DSPs, the new proposed algorithm finds the best possible path to ensure the minimum required path delay. The routing table is constructed by using the neighbor table entries. Neighbor table contains multiple records for each destination. For example, Figure 3 shows that there are many paths from B_3 to NSC. Some of these paths are B_3-B_1-MDC_2-NSC, B_3-MDC_3-B_2-MDC_1-NSC, and so forth. For each destination, the routing table constructor stores the next hop (NH_D) which has the lowest latency.

Algorithm 1 shows that node i identifies the next hop candidates by searching the records which have the same $\mathrm{ID}_{\mathrm{Dst}}$ in the neighbor table. The path delay has been calculated by using the neighbor table constructor and stored in neighbor table for each next hop candidate, using (17). The node stores the neighbor nodes' IDs in the variable NH (line 2). If NH has only one entry, this means there is only one path available. The node stores this entry to NH_D (line 4).

Otherwise, the node sorts the NH entries in ascending order of delay and then stores the first entry which has the lowest path delay in NH_D (lines 6-7). The next hop candidate NH_D is then stored with its path delay value ($\mathrm{DL}_{\mathrm{path}(i,\mathrm{Dst})}$) in the routing table. The data packets from both upper layers and packet classifier are received by QoS classifier. The QoS classifier classifies the packets into DSP and OP data. For

each data packet, the path selector (PS) checks the QoS requirement and chooses the most appropriate next hop(s) by using Algorithm 2.

The path selector compares the delay requirement ($\mathrm{DL}_{\mathrm{req}}$) with the path delay ($\mathrm{DL}_{\mathrm{path}(i,\mathrm{Dst})}$) of NH_D which is stored in the routing table. If the path delay ($\mathrm{DL}_{\mathrm{path}(i,\mathrm{Dst})}$) is lower than required delay ($\mathrm{DL}_{\mathrm{req}}$), the packet is sent to NH_D (lines 3-4). Otherwise, the packet is dropped (line 6).

For ordinary packets, the PS returns the next hop NH_E which is discussed by the EPR (lines 8-9); else the packet is dropped.

4.6. QoS-Aware Queuing Module (QQM).

The routing services module passes the data packets to the QoS-aware queuing module (QQM) after choosing the appropriate next hop(s). The QQM receives the data packets and separates these packets in two classes (DSP and OP). An individual queue is used for each class of packets. QQM functions are the same as discussed in [13]. The priority of the DSP queue is higher than that of the OP queue. By default, the DSP queue with higher priority sends the packets first. The packets from lower priority OP queue will be sent only when the DSP queue is empty. However, for fair treatment of OP data, a timeout is used by all the queues. A queue sends the packets to the MAC layer within the period specified by the timeout for that queue. QQM changes the control from higher priority queue to lower priority queue after the queue timeout occurs or when the higher priority queue is empty whichever is earlier.

4.7. MAC Transmitter.

The MAC transmitter receives the data and Hello packets from the network layer and stores it in

INPUT: Neighbor table, i's neighbor table records $NH_{(i,Dst)}$, $\forall Dst \in \{MDC, NSC, BAN\}$
(1) **for** each destination $Dst \in \{NSC, MDC, BAN\}$ **do**
(2) $NH = \{All\ neighbor\ nodes\ j \in NH_{(i,Dst)}\}$
(3)   **if** ($|NH| == 1$) **then**
(4) $NH_D \leftarrow NH$
(5)   **else if** ($|NH| > 1$) **then**
(6)   Sort NH in ascending order of $DL_{path(i,Dst)}$
(7) $NH_D =$ first neighbor node $j \in NH$;
(8)   **end if**
(9) **end for**

ALGORITHM 1: Routing table construction algorithm for delay-sensitive packets.

INPUT: Routing table, i's routing table records $NH_{(i,Dst)}$, $\forall Dst \in \{MDC, NSC, BAN\}$
(1) **for** each data packet **do**
(2)   **if** data packet is delay-sensitive packet (DSP)
(3)   **if** ($DL_{path(i,Dst)} \leq DL_{req}$) **then**
(4)   send to NH_D
(5)   else
(6)   Drop the packet immediately
(7)   **end if**
(8)   **else if** data packet is Ordinary Packet (OP)
(9)   send to NH_E
(10)   **else**
(11)   drop the packet immediately
(12) **end if**
(13) **end for**

ALGORITHM 2: Path selector algorithm for delay-sensitive packets.

the queue. The queue works in a First-In-First-Out (FIFO) fashion. It transmits the packets after capturing the channel by using CSMA/CA algorithm.

5. Performance Evaluation

Simulations are performed on OMNeT++ based simulator Castalia 3.2 [19]. In this section, the proposed QPRD algorithm is compared with the DMQoS [13] and noRouting protocols. In noRouting, the delay-sensitive data packets are forwarded to random next hop devices instead of algorithm's next hop based on end-to-end path delay routes. The network parameters used in simulations are shown in Table 1.

Three scenarios are considered for simulation. All the nodes used in scenario 1 are static, whereas the source node B_4 is moving in scenario 2. Scenario 3 is used for the scalability test of the protocol. The transmit power used in the simulations is −25 dBm. The performance of the QPRD is measured by calculating the throughput, number of packets forwarded by the intermediate nodes, overall network traffic, packets timeout due to not fulfilling the required delay condition, and packets dropped due to the buffer overflow. The better results provided by QPRD are in accordance with the equations used in Section 4. The higher throughput is due to the use of objective function in QPRD, as described in (1), and the least violations of (4c), (4d), and (4e). The simulation

MDC₁ (2, 5)   B₁ (5, 5)

NSC and MDC₄ (0, 3)   MDC₃ (3, 3)   B₃ (6, 3)   B₄ (9, 3)
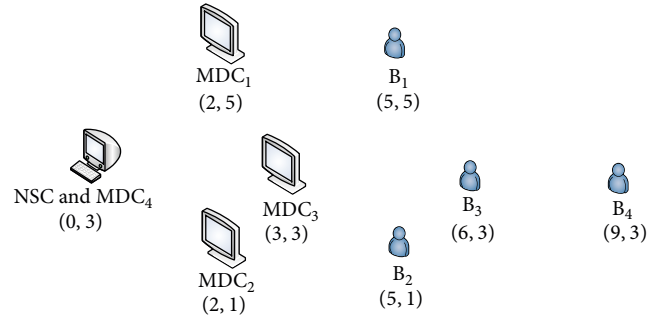
MDC₂ (2, 1)   B₂ (5, 1)

FIGURE 4: Node deployment for scenario 1.

results show that the end-to-end path delay mechanism, as discussed in Sections 3.2 and 4.4, used in QPRD helps to reduce the packets forwarded by intermediate nodes and the packets dropped due to the buffer overflow, which results in higher throughput and lower overall network traffic. To achieve a 97% confidence interval for the illustrative results, the average of three runs is simulated in every experiment which may introduce a maximum error of 3×10^{-3}, based on the error calculation done by Castalia 3.2 simulator [20]. The results obtained for first two scenarios are discussed below.
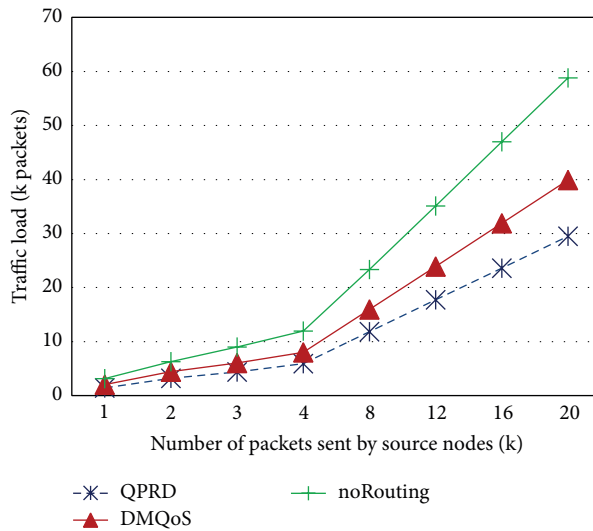
5.1. Scenario 1: Static Nodes. Figure 4 shows the deployment of the experimental network for scenario 1. All the nodes are
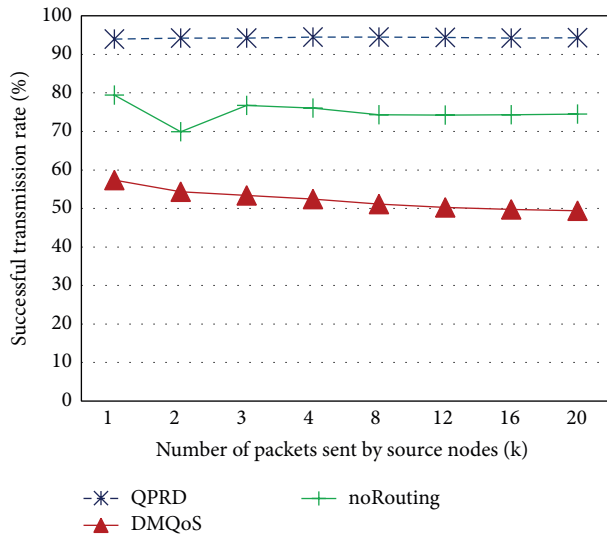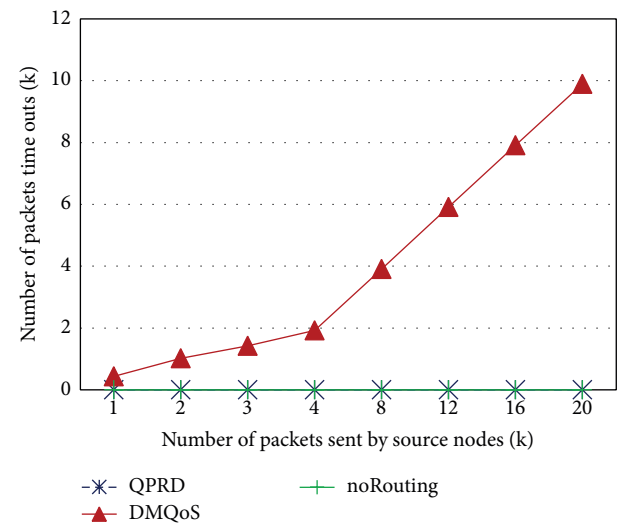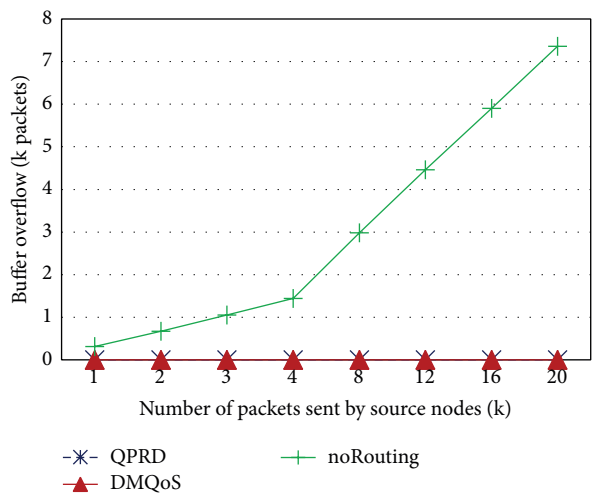
(a) Throughput
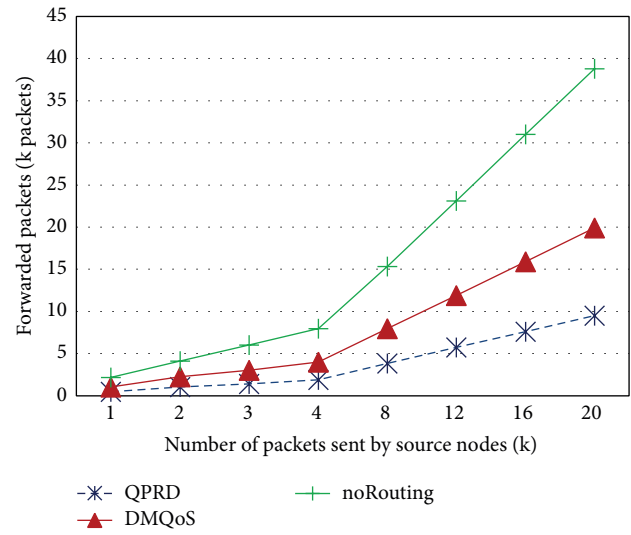
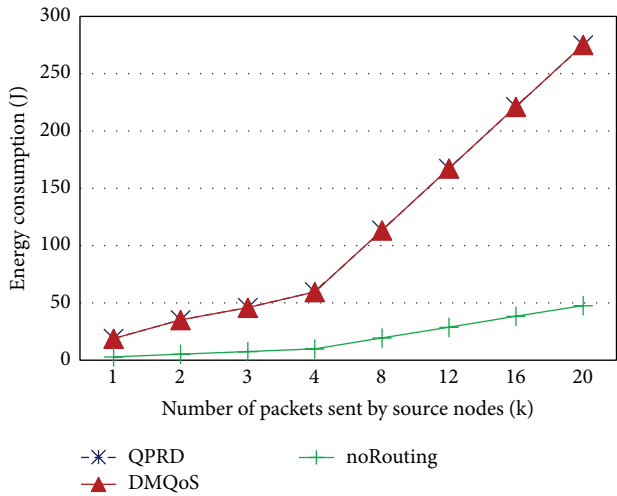(b) Packets forwarded by intermediate nodes

(c) Overall network traffic

(d) Packets timeout

(e) Packets dropped due to MAC buffer overflow

(f) Overall energy consumption

FIGURE 5: Performance comparison for different parameters when source nodes are static.

TABLE 1: Parameters information.

Deployment	Area	Scenarios 1 and 2: 9 m by 9 m
		Scenario 3: 21 m × 16 m
	Deployment type	Scenario 1: all nodes are static
		Scenario 2: movable source node B_4
		Scenario 3: hospital environment
	Number of nodes	Scenarios 1 and 2: 8 nodes (4 BANs, 3 MDCs, 1 NSC)
		Scenario 3: 49 nodes (24 BANs, 24 MDCs, 1 NSC)
	Initial nodes locations	Scenarios 1 and 2: NSC(0,3),
		$MDC_1(2,5)$, $MDC_2(2,1)$, $MDC_3(3,3)$
		$B_1(5,5)$, $B_2(5,1)$, $B_3(6,3)$, $B_4(9,3)$
		Scenario 3: shown in Figure 8
	Initial node energy	18720 J (= 2 AA batteries)
	Buffer size	32 packets
	Link layer trans. rate	250 Kbps
	Transmit power	−25 dBm
Task	Application type	Event-driven
	Max. packet size	32 bytes
	Traffic type	CBR (Constant Bit Rate)
MAC	IEEE 802.15.4	Default values
Simulation	Time	2003 seconds
		(3 seconds are setup time. Simulation results are the average of three rotations.)

static in this scenario. The type 1 devices (BANCs: B_1, B_2, B_3, and B_4) are considered as source nodes, and type 2 devices (NSC and MDCs) are the destination nodes. B_1 sends packets to MDC_1, B_2 sends packets to MDC_2, B_3 sends packets to MDC_3, and B_4 sends packets to NSC. The data of B_4 has to go through the other devices to reach NSC. The source nodes send a total of 20 k delay-sensitive packets. The throughput, packets forwarded by intermediate nodes, overall network traffic, number of packets timeout, packets dropped due to MAC buffer overflow, and overall energy consumption are calculated after every 1000 packets until 4 k and then every 4000 packets sent by all BANCs.

From Figure 5(a), it is seen that QPRD consistently provides throughput of 94% or more. In comparison, noRouting provides an average of 74% transmission rate, whereas DMQoS has a throughput ranging from 49% to 57%. For low offered data loads of 1 k, DMQoS has a throughput of 57% that continues to decrease especially for high offered data loads of 20 k, when the throughput is 49%. The low throughput in DMQoS may be explained by the way it selects the next hop using the energy-aware geographic forwarding scheme. Because the best next hop does not guarantee that it has the smallest latency connection to the destination, the packet may timeout when it is sent using the "best" next hop. Moreover, the energy-aware geographic forwarding scheme used in DMQoS prefers the nearest next hop candidate in terms of hop count and ignores next hop nodes having a lower delay. As a result, the network traffic is increased and the packets are dropped due to timeout before reaching the destination. QPRD resolves these issues by using the end-to-end path delay.

B_2 is the closest node to the destination nodes (i.e., NSC or MDCs) as shown in Figure 4. In DMQoS [13], B_2 is responsible for forwarding the data packets from other nodes to NSC or MDCs. This results in more energy consumption for B_2 and increased traffic congestion experienced by B_2. EPR resolves these problems by choosing the most appropriate next hop. In the proposed QPRD scheme, the BAN coordinator does not send data to another BAN coordinator unless it is absolutely necessary. Figure 5(b) shows the number of packets forwarded by the intermediate nodes. It is seen from Figure 5(b) that number of data packets forwarded by intermediate nodes before reaching the destinations in QPRD are on average 0.5 times and 3 times lower than DMQoS and noRouting, respectively.

The lower number of forwarded packets by intermediate nodes helps to reduce the overall network traffic. Figure 5(c) shows the total network traffic generated by QPRD, DMQoS, and noRouting as a function of the offered traffic load. From this Figure, it is seen that QPRD generates about an average of 26% and 99% less traffic in the network compared to DMQoS and noRouting, respectively. The path calculation in QPRD considers the delay of all the nodes and uses the best path delay information to select the next hop to send the data from source to destination.

In contrast to the method used in DMQoS which decides on the immediate next hop based merely on next hop delay instead of overall path delay, each upstream hop in DMQoS sends the packet to its next hop and resultant path in DMQoS may not be the most optimal.

From Figure 5(d) it is observed that QPRD and noRouting have no packets that were timed out for all offered
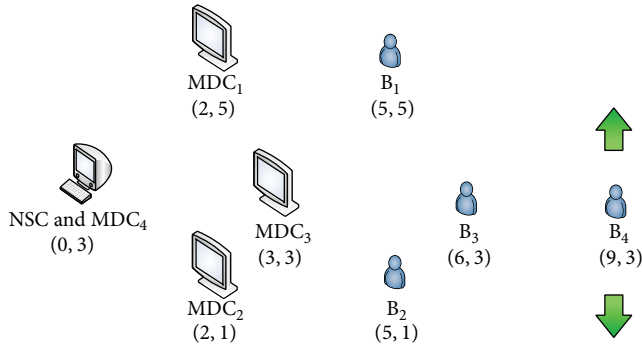
FIGURE 6: Node deployment for scenario 2.

traffic loads (number of data packets sent by source node range from 0 k to 20 k). QPRD has better performance in terms of reduced overall network traffic and fewer numbers of dropped packets due to timeout, because the clear end-to-end path delay information helps the packet to reach the destination within the requested delay requirement. Moreover, the path calculation in QPRD considers the delay of all the nodes in the network and chooses only those paths which can guarantee delivering the packet to the destination before it times out.

Figure 5(e) shows that there is no packet dropped due to the MAC buffer overflow in QPRD protocol. This is due to no violation of the model constraints as explained in Sections 4.1 and 4.2. The source node chooses the path which provides the maximum throughput and minimum end-to-end path delay as described in [16, 18]. Only few packets are dropped in DMQoS, whereas 7.5 k packets are dropped in noRouting.

It is seen from Figure 5(f) that the end-to-end path delay mechanism used in QPRD does not affect the overall energy consumption when compared with DMQoS. QPRD and DMQoS consume the same 18 Joules to 275 Joules of energy when 1 k to 20 k packets are sent by source nodes. On the other hand, the energy consumption of noRouting protocol is 2.6 Joules to 47.7 Joules when 1 k to 20 k packets are sent by source nodes. The data packets in noRouting are randomly forwarded to three neighbor nodes without considering the delay requirements. The additional computations for delay in QPRD consume on average 6 times more energy than noRouting. However, it must be noted that noRouting results in on average a 99% higher overall network traffic. This may be attributed to the 3 times more packets forwarded by intermediate nodes in noRouting resulting in a 20% lower throughput as compared to QPRD.

In summary, QPRD outperforms DMQoS and noRouting when the source node is static.

5.2. Scenario 2: Mobile Source Node. In the second scenario, the source node B_4 is moving at the speed of 1 meter per second vertically as shown in Figure 6. It is assumed that the speed of a fast walking patient is 1 meter per second.

Once again, it is observed that QPRD provides better results than DMQoS and noRouting in case of mobile source node scenario. Figure 7(a) shows that the throughput is in

excess of 80% in QPRD for offered data packet rates less than 8 k. The throughput reduces slightly at higher offered data packet rates of 8 k and more and reduces to 71% when total offered packets sent by the source are 20 k. In contrast with DMQoS, it is observed that when the offered data packet load is increased, DMQoS suffers from a much lower successful data transmission rate that reduces from 50% to 32% with resultant low throughput. Due to node mobility, the source node moves away from its neighbor nodes resulting in a connection loss which results in more packets being lost. QPRD handles this situation much more gracefully than DMQoS. In QPRD, the mobile nodes resume the connection more rapidly once the nodes come back into the range of neighbor node. The overall lower throughput in this scenario is due to the packet lost when the mobile node is out of range. Equation (4d) in Section 3.1 also supports this behavior. According to (4d), the packet delivery is successful only if a source node transmits data to an in-range destination node. The packets are dropped when the movable nodes go out of range. The noRouting provides the lower throughput with an average of 64%.

Figure 7(b) shows that the number of packets forwarded by the intermediate nodes in QPRD is on average 0.75 times and 9 times lower when compared to the number of packets forwarded by intermediate nodes in DMQoS and noRouting protocols, respectively. The routing mechanism used in the QPRD protocol helps to send the data directly to the destination without transferring the packets to the intermediate nodes in case the destination is in range. It can be seen in Section 3.3 that use of intermediate node results in larger delay and in Section 3.2 that the backoff of other noncapturing nodes also contributes to exacerbating the problem. The performance of noRouting for this parameter is worst as it forwards up to 26 k packets which increases the overall network traffic.

It is observed from Figure 7(c) that the overall network traffic in QPRD is about 25% and 50% less than DMQoS and noRouting protocols, respectively, for all offered network data loads considered. This is due to the end-to-end path calculation mechanism used in QPRD. The delay of all the nodes is considered and QPRD algorithm selects the best next hop, on the basis of end-to-end path delay information, to send the data from source to destination.

From Figure 7(d), it is seen that QPRD has no packets that were timed out for data packet transmissions at 8 k or less. The selection of minimum end-to-end path delay, given in [18], helps QPRD to send the data through a path where lower packets time out occurs. For high data packets (above 8 k), the source node moves out of the neighbors' radio range which causes more packets to time out. On the other hand, DMQoS has more timed out packets than QPRD. Initially for low offered data packet rates below 4 k, about 40% of data packets were timed out, and for higher offered data packets (above 4 k) the 40% of data packet time outs increase to 50% (approximately). This is because the packets travel through different nodes by using hop-by-hop delay calculation as discussed in detail in scenario 1. Equation (9) in Section 4.2 shows that the delay on each node is the summation of four different delays (i.e., transmission (DL_{trans}), MAC and network queues
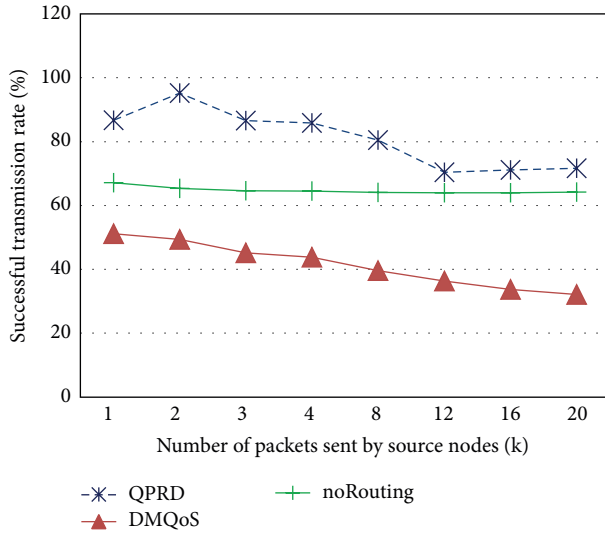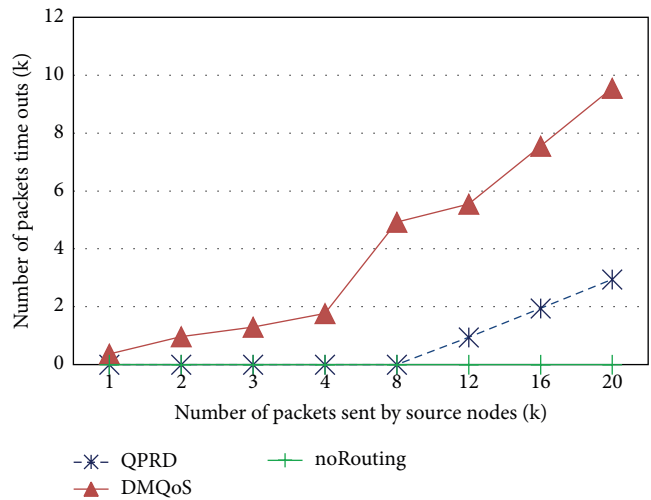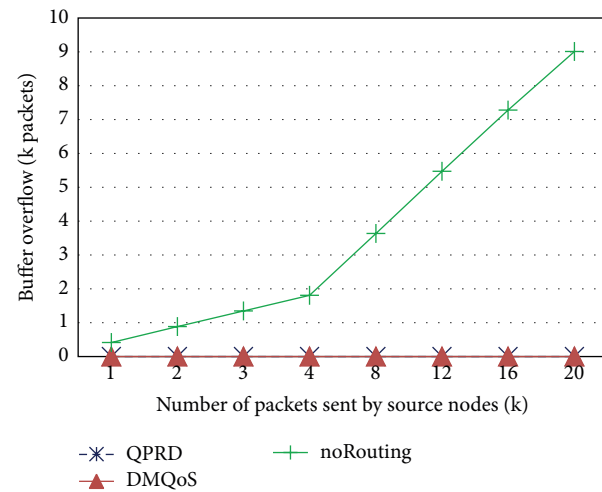
(a) Throughput
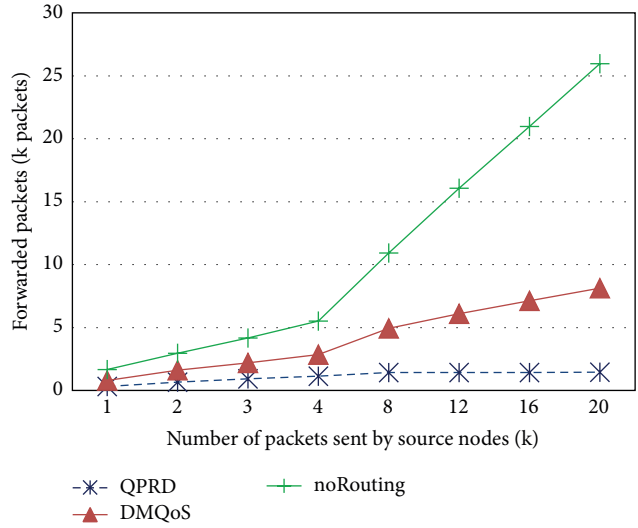
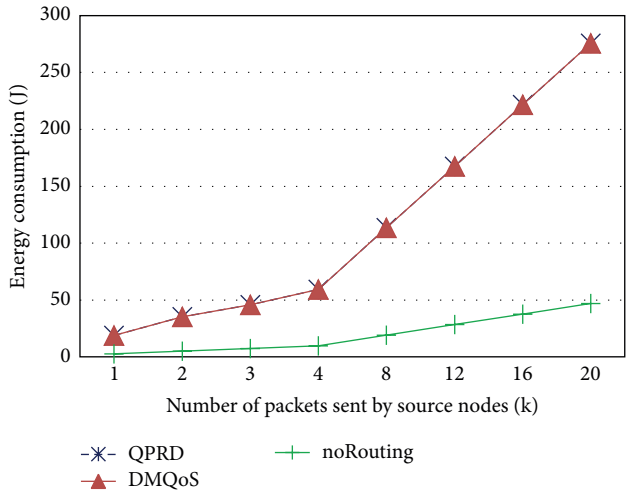(b) Packets forwarded by intermediate nodes
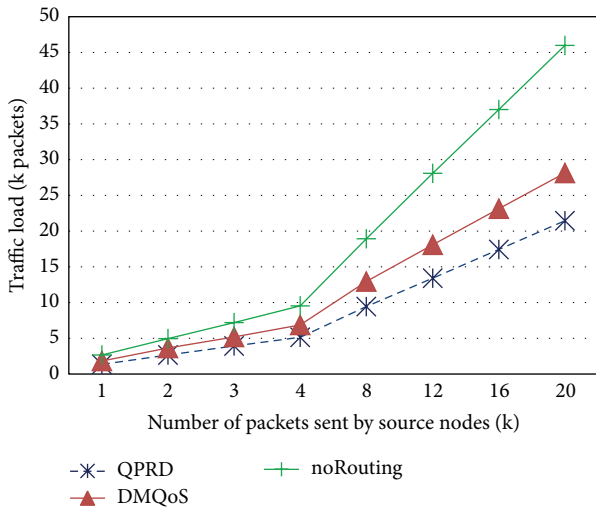
(c) Overall network traffic

(d) Packets timeout

(e) Packets dropped due to MAC buffer overflow

(f) Overall energy consumption

FIGURE 7: Performance comparison for different parameters when source node is mobile.
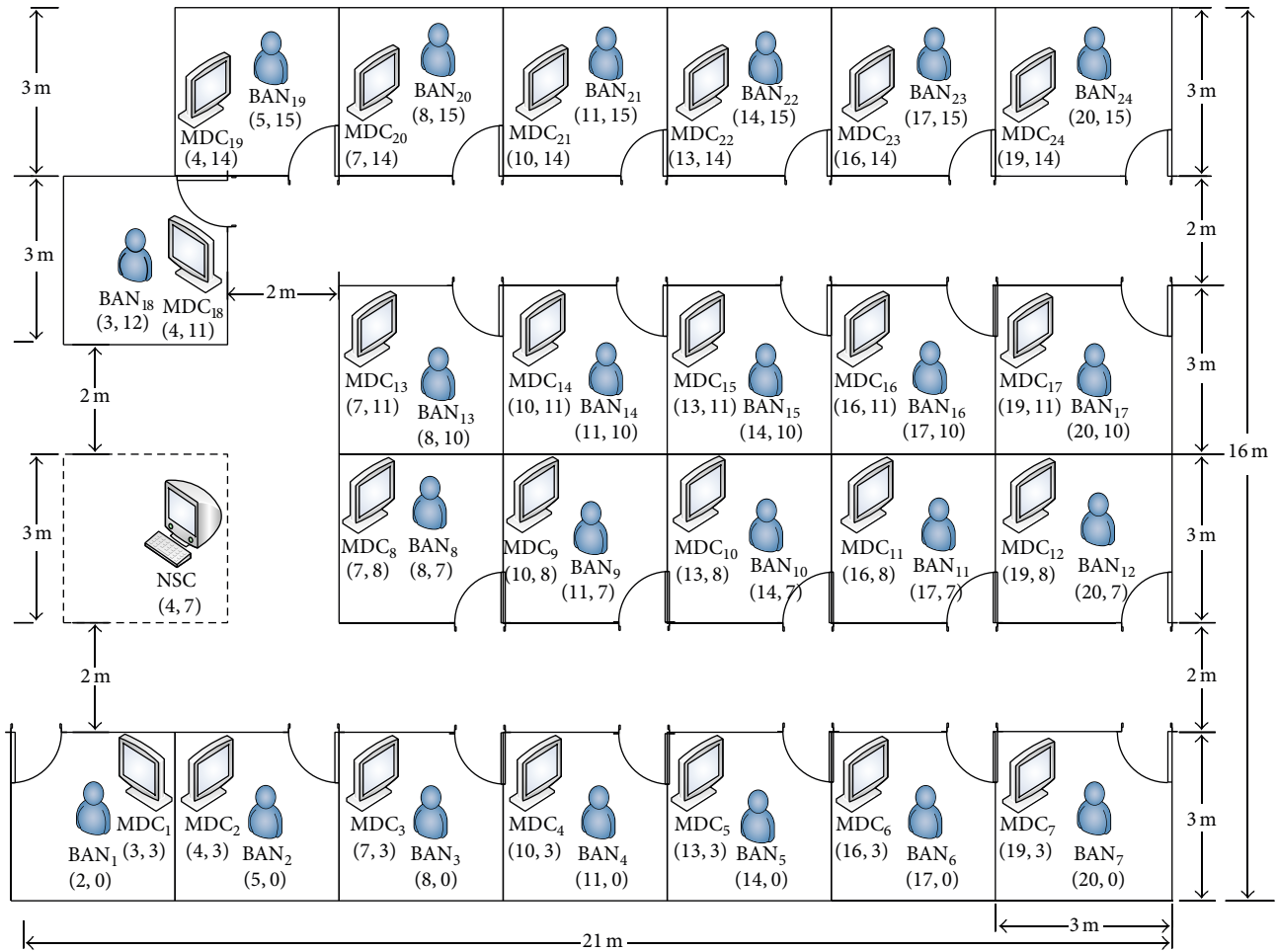
FIGURE 8: Scenario 3. Node deployment for 24 patient beds in hospital environment.

(DL_{queue}), channel ($DL_{channel}$), and processing (DL_{proc}). The calculations done by DMQoS on each node increase the processing delay which causes the increase of overall node delay. The higher node delay results in packet time out. The source node mobility makes the packet time out worse than scenario 1 of Figure 5(d).

Figure 7(e) shows that there are no packet drops due to MAC buffer overflow in QPRD and DMQoS protocols, whereas 9 k packets are dropped in noRouting. The performance of DMQoS is similar to QPRD in terms of MAC buffer overflow; however, DMQoS has on average 39% lower throughput and an average of 25% higher overall network traffic.

From Figure 7(f), it is observed that the overall energy consumptions of QPRD and DMQoS are 18.9 Joules to 275.7 Joules when 1 k to 20 k packets are sent by source nodes. The noRouting consumes 2.6 Joules to 47 Joules when 1 k to 20 k packets are sent by source nodes. The computations for delay in QPRD are almost similar to the DMQoS but QPRD provides on average 25% lower overall network traffic, 73% fewer packets forwarded by intermediate nodes, and, more importantly, a 40% higher successful data transmission rate (throughput) as compared to DMQoS.

In summary, the overall performance of QPRD is better than DMQoS and noRouting when the source node is mobile.

6. Scalability Test: Real Hospital Environment with 24 Beds (49 Nodes)

A real 24-patient-bed hospital is considered for the scalability test of QPRD routing protocol, as shown in Figure 8. The approximate area covered is 16 m by 21 m which is similar to the Hematology-Oncology Unit of the Children Hospital named IWK Health Centre Halifax, Canada. The distance between two beds is 3 meters which is in accordance with the recommended transmission range for BAN communication in hospital environment. The total nodes used in the deployment area are 49 (24 BANs, 24 MDCs, and 1 NSC). Each BAN transmits the data to its peer MDC. All the BANs and MDCs are sending or receiving Hello protocols to/from other nodes and the NSC.

Both MDCs and BANs are movable. Generally, BANs can move freely anywhere and the movement of a MDC is only within the room where it is placed. It is assumed that the MDC of one room has a connection with the MDC of the next room.

(a) Throughput versus offered load

(b) Overall network traffic

(c) Packets dropped due to MAC buffer overflow
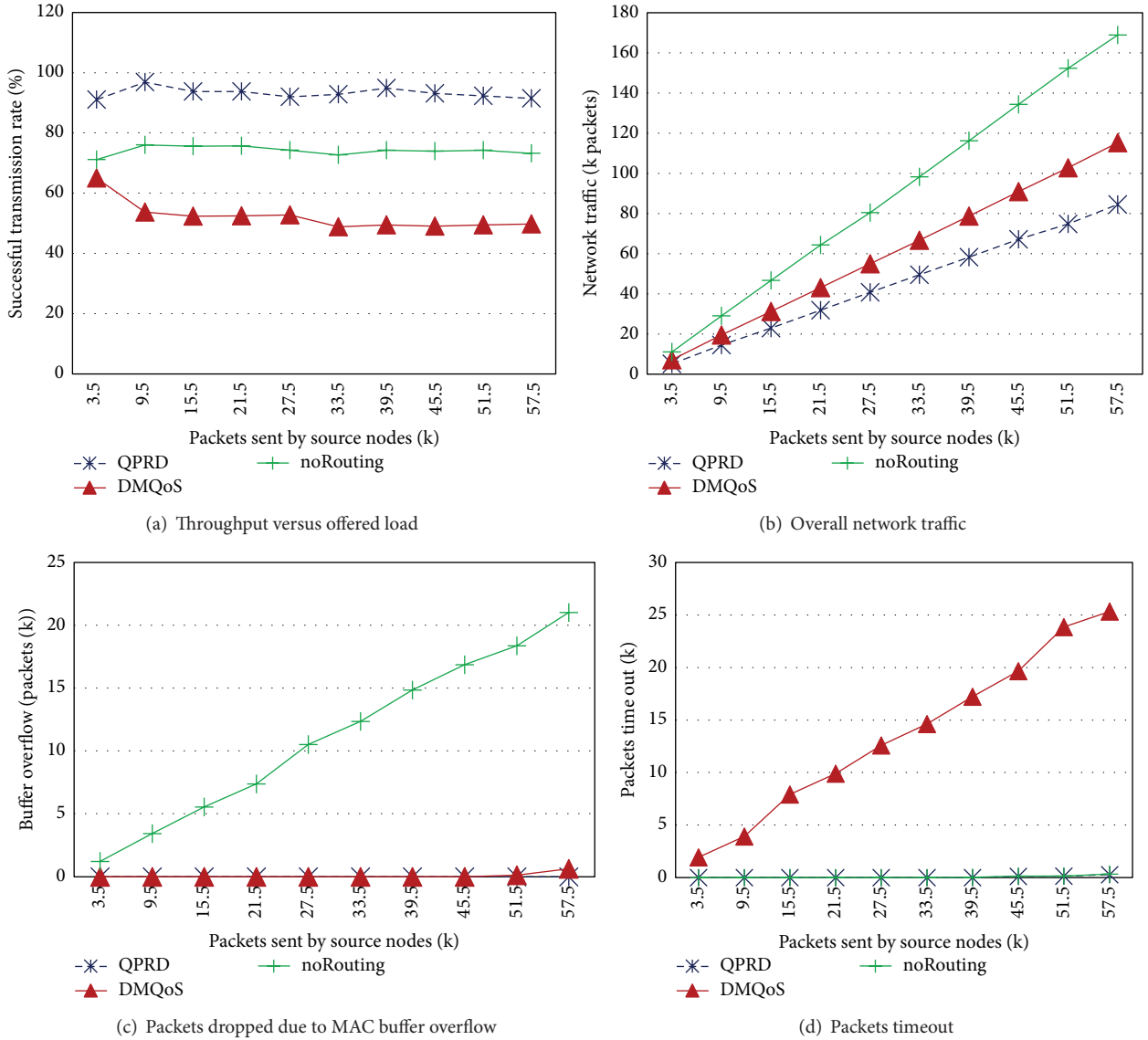

(d) Packets timeout

FIGURE 9: Performance comparison for different parameters of scenario 3.

The simulation results show that QPRD performs better than DMQoS and noRouting even when the number of nodes is increased to 49. From Figure 9(a) it is seen that the throughput provided by QPRD is in excess of 91%, whereas the throughput of noRouting and DMQoS protocols is on average 74% and 52%, respectively. From Figure 9(b), it is observed that the overall network traffic of QPRD is 50% and 25% less than noRouting and DMQoS protocols, respectively. Figure 9(c) shows that the packet drops due to MAC buffer overflow in QPRD and DMQoS protocols are negligible, whereas 9 k packets are dropped in noRouting. Figure 9(d) shows that there are no packets timeouts due to not fulfilling the delay requirements in QPRD and noRouting. On the other hand 25 k packets are timed out in DMQoS. From these results it is shown that QPRD is equally effective when the deployment area is larger, and number of nodes has been increased to simulate a real hospital scenario with 24 patient beds.

7. Conclusion

The paper models the wireless BAN as a directed graph and derives conditions for throughput maximization and end-to-end delay minimization. It is shown that efficient energy utilization is critical to the proper design of the routing and MAC layer protocols. Similarly, delay is minimized by formulating the BAN end-to-end path delay as a linear programming problem with multiple constraints to be satisfied simultaneously.

Based on the mathematical analysis, a novel modular QoS-aware routing protocol for hospital BAN communication is proposed in this paper. The architecture of the new protocol consists of seven modules: the MAC receiver, the delay module (DM), the packet classifier (PC), the Hello protocol module (HPM), the routing services module (RSM), the QoS-aware queuing module (QQM), and the MAC transmitter. The proposed routing protocol provides

a mechanism for the end-to-end path delay calculation of all possible paths from a source to destination and then decides the best possible path by considering the path delay requirements of the delay-sensitive packets.

OMNeT++ based simulator Castalia 3.2 is used to test the performance of the proposed protocol (QPRD) and compare it with DMQoS and noRouting. The simulations are performed for both the movable source and stationary scenarios. A scalability test is done with larger deployment area and by using higher number of nodes. The results show that the QPRD offers over 94% successful data transmission rates for delay-sensitive packets in a stationary patient scenario. QPRD provides about 35% better results in terms of successful transmission rate than DMQoS in the movable patient scenario. The simulation results show that the QPRD improves the reliability of Body Area Networks by 40% on average for each scenario by decreasing the number of packet time outs with zero and averaging 729 packets for the static and mobile patient scenarios, respectively. In addition, QPRD results in an average of 25% lower overall network traffic for each mobile and static patient scenarios as compared to similar protocols. The scalability test results prove that QPRD outperforms DMQoS and noRouting even when a higher number of nodes are employed in the BAN. QPRD provides on average 93% throughput without any packet being timed out and any packet being dropped due to MAC buffer overflow.

Notations for the Proposed Algorithm

Node i: Source node
Node j: Neighbor node of source node
Node Dst: Destination node (i.e., NSC, MDC, BAN)
ID_{Dst}:   Destination ID
L_{Dst}:   Destination location
ID_j: Neighbor node j ID
L_j: Neighbor node j location
$D_{(j,Dst)}$:   Distance between neighbor node j and destination Dst
E_j: Residual energy of node j
T_j: Device type of node j
$D_{(i,j)}$:   Distance between node i to neighbor node j
$NH_{(i,Dst)}$:   Next hop between node i and destination Dst
NH_E: Energy-aware next hop
NH_D: Next hop for delay-sensitive packets
$DL_{path(i,Dst)}$:   Path delay from node i to destination Dst
$DL_{node(i)}$:   Time delay within the node i
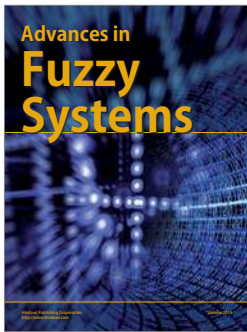DL_{req}:   Required path delay for delay-sensitive packets.
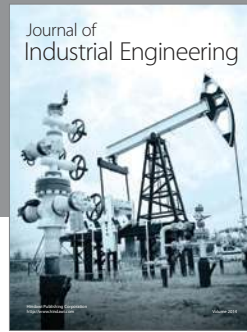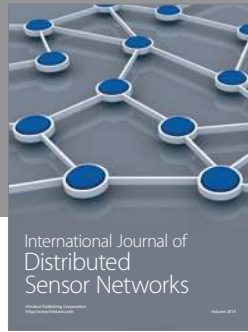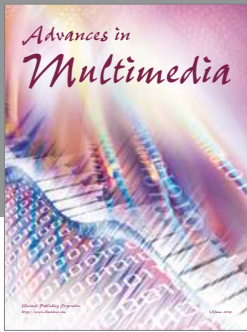
Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] B. Zhen, M. Patel, S. Lee, E. T. Won, and A. Astrin, "15-08-0644-09-0006-tg6-technical-requirements," IEEE Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), 2011, https://mentor.ieee.org/802.15/dcn/11/15-11-0307-00-0006-tg6-closing-report-march-2011.ppt.

[2] IEEE, "IEEE 802.15 WPAN task group 6 (TG6) body area networks," IEEE 802.15, 2007, http://ieee802.org/15/pub/TG6.html.

[3] D. Curtis, E. Shih, J. Waterman et al., "Physiological signal monitoring in the waiting areas of an emergency room," in Proceedings of the ICST 3rd International Conference on Body Area Networks (BodyNets '08), Tempe, Ariz, USA, 2008.

[4] S. Jiang, Y. Cao, S. Iyengar et al., "CareNet: an integrated wireless sensor networking environment for remote healthcare," in Proceedings of the 3rd International ICST Conference on Body Area Networks (BodyNets '08), Tempe, Ariz, USA, March 2008.

[5] T. Gao, T. Massey, L. Selavo et al., "The advanced health and disaster aid network: a light-weight wireless medical system for triage," IEEE Transactions on Biomedical Circuits and Systems, vol. 1, no. 3, pp. 203–216, 2007.

[6] A. Wood, G. Virone, T. Doan et al., "ALARM-NET: wireless sensor networks for assisted-living and residential monitoring," Tech. Rep. CS-2006-11, Department of Computer Science, University of Virginia, Charlottesville, Va, USA, 2006.

[7] Z. Khan, S. Sivakumar, W. Phillips, and N. Aslam, "A new patient monitoring framework and Energy-aware Peering Routing Protocol (EPR) for Body Area Network communication," Journal of Ambient Intelligence and Humanized Computing, vol. 5, no. 3, pp. 409–423, 2014.

[8] Z. Khan, S. Sivakumar, W. Phillips, and B. Robertson, "A QoS-aware routing protocol for reliability sensitive data in hospital body area networks," Procedia Computer Science, vol. 19, pp. 171–179, 2013.

[9] Z. Khan, S. Sivakumar, W. Phillips, and B. Robertson, "QPRR: QoS-aware peering routing protocol for reliability sensitive data in body area network communication," The Computer Journal, 2014.

[10] Z. Khan, S. Sivakumar, W. Phillips, and B. Robertson, "QPRD: QoS-aware peering routing protocol for delay sensitive data in hospital body area network communication," in Proceedings of the 7th International Conference on Broadband, Wireless Computing, Communication and Applications (IEEE BWCCA '12), pp. 178–185, University of Victoria, Victoria, Canada, November 2012.

[11] X. Huang and Y. Fang, "Multiconstrained QoS multipath routing in wireless sensor networks," Wireless Networks, vol. 14, no. 4, pp. 465–478, 2008.

[12] S. Agarwal, Divya, and G. N. Pandey, "SVM based context awareness using body area sensor network for pervasive healthcare monitoring," in Proceedings of the 1st International Conference on Intelligent Interactive Technologies and Multimedia (IITM '10), pp. 271–278, Allahabad, India, December 2010.

[13] A. Razzaque, C. S. Hong, and S. Lee, "Data-centric multiobjective QoS-aware routing protocol for body sensor networks," Sensors, vol. 11, no. 1, pp. 917–937, 2011.

[14] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 5, no. 6, pp. 738–754, 2006.

[15] M. Chen, T. Kwon, S. Mao, Y. Yuan, and V. C. M. Leung, "Reliable and energy-efficient routing protocol in dense wireless

sensor networks," *International Journal of Sensor Networks*, vol. 4, no. 1-2, pp. 104–117, 2008.

[16] M. Razzaque, M. Alam, M. Rashid, and C. Hong, "Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, Kyung Hee University, Las Vegas, Nev, USA, January 2008.

[17] J. Elias and A. Mehaoua, "Energy-aware topology design for wireless body area networks," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 3409–3410, Ottawa, Canada, June 2012.

[18] N. Ababneh, N. Timmons, J. Morrison, and D. Tracey, "Energy-balanced rate assignment and routing protocol for body area networks," in *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA '12)*, pp. 466–467, Fukuoka, Japan, March 2012.

[19] NICTA, "Castalia," National ICT Australia, 2011, http://castalia.npc.nicta.com.au/.

[20] A. Boulis, "Castalia, wireless sensor network simulator, NICTA," 2011, https://forge.nicta.com.au/docman/view.php/301/592/Castalia+-+User+Manual.pdf.