Analysis of QoS Parameters for Energy Efficient Routing in Wireless Sensor Networks

Nandkishor Sirdeshpande¹, Mayurkumar Dobaria², Vishwanath Udupi³

¹Department of Electronics and Communication Engineering
²Gogte Institute of Technology, Belgaum - 590008, Karnataka, India
³vishwa_u@yahoo.com

Abstract- Wireless Sensor Networks (WSNs) are required to provide energy efficiency and different level of Quality of Service (QoS) based on the type of applications. Due to resource constraints like memory, power sources, bandwidth and processing power in WSNs, the energy efficient routing and QoS support becomes an important issue. In this paper we propose Analysis of QoS Parameters for Energy Efficient Routing protocol (we shortly named as AQSER) that maximize the network lifetime through traffic balancing across multiple nodes. AQSER differentiates Real Time (RT) and Non- Real Time (NRT) traffic to allow high important traffic to reach destination within acceptable delay reduces average end-to-end delay by sending traffics over multiple paths and increases the throughput through introducing error correction code. AQSER uses the residual energy, buffer size, Signal-to-Noise Ratio (SNR), Packet Reception Ratio (PRR), Received Signal Strength Indicator (RSSI), Required Number of Packets retransmission (RNP) and Average Probability of Packet Reception (APPR) to predict the best next hop through the paths construction phase. Based on the concept of service differentiation the AQSER protocol uses queuing model to handle both real time and non-real time traffic. Protocol dynamically changes the node in case of node failure while sending the data on the paths and decreases the energy consumption in case of node failure by means increase the lifetime of the network. By means of computer simulations, we have evaluated and studied the performance of our routing protocol and compared it with another protocol. Simulation results have shown that our protocol achieves lower average delay and energy consumption and increases packet delivery ratio then other protocol.

Keywords- Energy Efficiency, Error Correction, Multipath Routing, Quality of Service, Wireless Sensor Networks.

I. INTRODUCTION

In recent years, Wireless Sensor Networks (WSNs) has become one of the cutting edge technologies for low power wireless communication. These low power technologies have created low power, low cost and multifunctional devices, which are used to react to changes in the physical phenomena of their surrounding environments. The sensor nodes consist of small battery, a tiny microprocessor, a radio transceiver, and a set of transducers that used to get the information that reflects in the environment of the sensor node. The fast development of low power and small size sensor devices, the significant development of distributed signal processing, adhoc network protocol and pervasive computing have collectively set new vision for WSNs [1][2].

The WSNs consists number of sensor devices that connected to each other using wireless interface to achieve a common task and report the gathered data to the destination. The data collection process can be of different type [3] (e.g. continuous, query based, event driven). There are various applications (target tracking in battlefields, habitat monitoring, civil structure monitoring, forest fire detection, and factory maintenance) of WSNs depending on the environment it is used.

By considering the some properties of the WSNs such as high network density, limited power, large scale deployments, and dynamic topology have posed many challenges in the design and development of WSNs. All these challenges have demanded energy aware and QoS protocol designs at all layers of the protocol stack [4].

To provide the best Quality of service (QoS); the efficient utilization of sensor energy resources and maximization of the network lifetime is necessary, but these two things were and still are the main design considerations for wireless sensor networks. The QoS such as latency, throughput, packet loss and jitter have not yet gained a great focus from the research community. However, the generated data from the sensors have different attributes depending upon the type of application, where it may be delay sensitive and reliability demanding data. The increasing interest of multimedia applications in the WSNs demands delay and throughput in order to reach the data to the destination. Allowing multimedia applications in WSNs requires energy and QoS support in different layers of the protocol stack in order to have efficient utilization of the network resources. Thus, QoS and energy efficient routing is an important topic in sensor networks research. Refer to [5], and [6] for the surveys on QoS based routing protocol in WSNs.
Due to the limited transmission range of the wireless sensor nodes, multiple hops are usually needed for nodes to exchange information between each other in the network. Thus, routing becomes a critical issue in designing sensor networks. Routing in sensor networks is a very challenging task and different from routing in either wired or other wireless networks, because of the many special characteristics of WSNs. There are many routing techniques designed for WSNs have been proposed in [7] [8]. In these proposals, the unique properties of the WSNs have been considered. Depending on the protocol operation this routing technique classified in to negotiation based, query based, QoS based and multipath based. The negotiation based protocols have the objective to eliminate the redundant data by including high level data descriptors in the message exchange. In query based protocols, the sink node initiates the communication by broadcasting a query for data over the network. The QoS based protocols allow sensor nodes to make a tradeoff between the energy consumption and some QoS metrics before delivering the data to the sink node [9]. Finally, multipath routing protocols use multiple paths rather than a single path in order to improve the network performance in terms of reliability and robustness. Multipath routing establishes multiple paths between the source-destination pair. The Multi-path routing protocols have been discussed in the literature [10]. We focus on the supporting QoS through multipath routing.

In this paper, we propose AQSER protocol: Analysis of QoS Parameters for Energy Efficient Routing in WSNs to recover from node failures and achieves load balancing through splitting up the traffic across a set of available node-disjoint paths, there by minimize the energy consumption. The protocol dynamically changes the route in case of node failure while sending the data, hence decreases the overhead and energy consumption in the sensor networks. Furthermore, AQSER increases the reliability of data delivery through sending the data with additional error correction codes computed from the data itself. This technique increases resiliency to path failures and increases the probability so that an enough portion of the packet is received at the destination to recover the original data message without incurring excessive delay through invoking data retransmissions. AQSER uses the residual energy, buffer size, Signal-to-Noise Ratio (SNR), Packet Reception Ratio (PRR), Received Signal Strength Indicator (RSSI), Required Number of Packets retransmission (RNP) and Average Probability of Packet Reception (APPR) to predict the best next hop through the paths construction phase. AQSER consists of queuing model that handles real time and nonreal time traffic through service differentiation by giving real time traffic higher priority than nonreal time traffic.

The remainder of the paper is organized as follow: In section II, we describe and discuss some of the related work. We detail our proposal in section III. Section IV presents the performance evaluation. Finally, we conclude the paper in section V.

II. RELATED WORK

Due to the lack of resources of the sensor node, the QoS based routing in sensor networks is a challenging problem. This problem got the attention from the research community, where many proposals are made. Some QoS oriented routing proposals are surveyed in [5] and [6]. Here, we discuss some proposals related to our protocol.

In [11] author proposed Sequential assignment Routing (SAR) protocol, it takes three parameters in the account to make the routing decision, they are energy resources, packet priority level and QoS on each path. Multiple paths are created between sink and source using tree routing. The nodes with low residual energy and low QoS are avoided during path construction phase. To transmit the data to the sink, the additive QoS metric and a weighted coefficient associated with the priority level of the packet are used as QoS metric to select the path. Multiple paths increases fault tolerance, but SAR suffers from maintaining routing tables and QoS metrics at each sensor node.

K. Akkaya and M. Younis in [12] proposed a cluster based QoS aware routing protocol that employs a queuing model to handle both real-time and non real time traffic. The protocol only considers the end-to-end delay. The protocol associates a cost function with each link and uses the K-least-cost path algorithm to find a set of the best candidate routes. Each of the routes is checked against the end-to-end constraints and the route that satisfies the constraints is chosen to send the data to the sink. All nodes initially are assigned the same bandwidth ratio which makes constraints on other nodes which requires higher bandwidth ratio. Furthermore, the transmission delay is not considered in the estimation of the end-to-end delay, which sometimes results in selecting routes that do not meet the required end-to-end delay.

Lou W and Kwon Y [13] proposed Hybrid Multipath Scheme for Secure and Reliable Data Collection (H-SPREAD) which combines the introduced path construction process in N-to-1 Multipath Routing Protocol with a hybrid data transmission technique to improve reliability and security of data transmission in wireless sensor networks. However, since this approach utilizes the N-to-1 multipath routing algorithm to construct multiple paths, this protocol may suffer from the effects of wireless interference. Therefore, high packet loss ratio caused by interference can reduce the probability of successful packet retrieval at the sink node. Moreover, H-SPREAD only improves reliability and security of data delivery in the network, but it cannot enhance security of individual nodes.
Felemban, E., C.G. Lee and E. Ekici [14] have proposed multipath multispeed (MMSPEED) protocol; it is designed based on the cross-layer design approach between the network and MAC layer to provide QoS differentiation in terms of reliability and timeliness. From a timeliness perspective, MMSPEED extends the SPEED protocol [15] through introducing multiple speed levels to guarantee timeliness packet delivery. Accordingly, for $M$ virtual speed layers there exists $M$ different SetSpeeds. In this protocol, data packets are assigned to the appropriate speed layer to be placed in the suitable queue according to their speed category. After that, data packets are serviced in the FCFS policy. This mechanism ensures that highest priority packets are serviced before low-priority packets. MMSPEED provides a probabilistic QoS guarantee in two different domains through combining geographic forwarding technique with a multipath routing approach. Therefore, using geographic routing with greedy forwarding does not necessarily improve network performance metrics. This protocol cannot support long-life applications.

Ghaffari, A., A. Rahmani and A. Khademzadeh [16] have proposed Energy Efficient and QoS Geographic Routing (EQGR) protocol considers reliability, timeliness and energy for selecting next optimum neighbor node for data forwarding. For reliable data forwarding the authors consider multipath forwarding with optimum link quality. For timeliness domain, they use multi queue policy for data forwarding. EQGR decreases the End- to- End delay and increases the reliable data delivery by dynamic load balancing and priority based data forwarding but the problem with this is that it does not consider any strategy for energy consumption in the network model.

A.B. Bagula, K.G. Mazandu in [17] have proposed the Energy constrained multipath routing (ECMP) that extends the MCMP protocol [18] by formulating the QoS routing problem as an energy optimization problem constrained by reliability, playback delay, and geo-spatial path selection constraints. The ECMP protocol trades between minimum number of hops and minimum energy by selecting the path that satisfies the QoS requirements and minimizes energy consumption. However, in MCMP, nodes randomly select their next-hop neighboring nodes without considering the amount of energy consumption over the chosen link. Therefore, compared to MCMP, ECMP refines the set of next-hop nodes to a smaller set through considering the energy efficiency of the links towards the neighboring nodes.

Li, S. et al. in [19] proposed Delay-Constrained High-Throughput (DCHT) protocol. It is the modified version of Directed Diffusion that propounds the idea of using multipath routing approach to support high-quality video streaming in low-power wireless sensor networks. This algorithm tries to improve data transmission reliability through forwarding each generated stream over two different paths. The utilized path reinforcement strategy and routing metric in this protocol greatly improves the performance of the original Directed Diffusion by constructing multiple high-quality low-delay paths. Each generated data stream should transmit through two different paths to provide a certain level of data transmission reliability. However, due to the random topology of the wireless sensor networks, constructing a sufficient number of node-disjoint paths to support high-rate multimedia streaming may not be feasible.

Bashir Yahya and Jalel Ben-Othman [20] Energy Efficient and QoS Aware Routing (EQSR) is one of the recently proposed protocol designed to satisfy the reliability and delay requirements of real-time applications. EQSR improve redundancy in the data transmission process. Furthermore, in order to fulfill the delay requirements of various applications, this protocol utilizes a service differentiation technique through employing a queuing model to manage real-time and non-real-time traffic. While EQSR reduces transmission delay and improves reliability, nevertheless, the FEC mechanism which is used to compute ECCs and retrieval of the original messages, imposes high control overhead. Furthermore, like DCHT, this protocol uses a flooding strategy to estimate the experienced SNR over wireless links at the initialization phase and uses these values to discover the minimum interfering paths.

III. DESCRIPTION OF AQSER PROTOCOL

In this section, we first define some assumptions, then we provide the details of multiple paths discovery and maintenance, as well as the traffic allocation and data transmission across the multiple paths.

A. Assumptions

We assume N identical sensor nodes are distributed randomly in the sensing filed. All nodes have the same transmission range, and have enough battery power to carry their sensing, computing, and communication activities. The network is fully connected and dens (i.e. data can be sent from one node to another in a multihop bases). Each node in the network is assigned a unique ID we assume that each sensor node is able to compute its Residual Energy (RE), Free Buffer (FB), as well as record the link performance between itself and its neighboring node in terms of signal-to- noise ratio (SNR), Packet Reception ratio (PRR), Received Signal Strength Indicator, Average Probability of Packet Reception (APPR), Required Number of Packet (RNP) retransmission. By examining recent link performance data, predications and decisions about path stability may be made.
\[ \text{Next hop} = \max_{\forall y \in N_x} (\alpha E_{\text{resd}, y} + \beta B_{\text{buffer}, y} + \delta P_{\text{pr}, y} + \varepsilon R_{\text{rsi}, y} + \gamma I_{\text{interference,xy}} + \theta R_{\text{rnp},y} + \rho A_{\text{appr},y}) \]

Where, \( E_{\text{resd}, y} \) is the current residual energy of node \( y \), \( B_{\text{buffer}, y} \) is the available buffer size of node \( y \), and \( I_{\text{interference,xy}} \) is the SNR for the link between \( x \) and \( y \). In this cost function, we only consider the residual energy of node \( y \) but not \( x \). Because of node \( y \) consumes energy for data reception and transmission if it is selected as next hop for node \( x \). We do not consider node \( x \), because whatever node \( y \) is, node \( x \) still needs to spend the same amount of energy on data transmission [22], \( P_{\text{pr}, y} \) is the packet reception ratio of node \( y \). \( R_{\text{rnp},y} \) is the required number of packet retransmissions, exclude first packet transmission.

This metric is evaluated at the sender side for each retransmitted packets. \( R_{\text{rsi},y} \) is the received signal strength indicator of node \( y \), can also be used as a metric to measure the wireless link quality. The packet reception rate is very good if average RSSI is above a certain threshold. According to their study if the RSSI value is less than the threshold, packet reception rate varied a lot. \( A_{\text{appr},y} \) is Average probability of packet reception of node \( y \), it is calculated as the

\[ \text{APPR}_{\text{neighbours}} = \sum_{i=1}^{\text{Total neighbours}} PPR_i \]

The cost associated with the individual link between two nodes is \( l_{xy} \), along the path. Then total cost \( (C_{\text{total}}) \) of the path \( P \) which consists \( K \) nodes is the sum of the individual link cost on the each path. Then total cost we have:

\[ C_{\text{total},P} = \sum_{i=1}^{K-1} l_{xy} \]

C. Path Discovery Phase

To forward the data from source to sink, the multiple paths are created between sink and source using a set of neighbours which are able to forward the data between source and sink, based on the idea of Directed diffusion [23] [24]. The paths are node-disjoint paths because these paths have advantages of the utilization of the most available resources so that they most fault-tolerant. Another advantage is that if any intermediate node fails in the node-disjoint path fails only path containing that node is affected, so that minimum impact to the diversity of the routes [23].

There are following phases to discover the path. They are:

- **First phase:** In this phase, HELLO is broadcast by each sensor node through the network to get enough information about its neighbors. Neighbor table of the each sensor is created and updates during this phase. Each neighboring table contains information about the neighbors of a particular node. Figure 2 shows the structure of the HELLO message that is broadcast by each sensor node. Link quality is given in terms of the signal-to-noise ratio (SNR) and hop count distance between a node and its neighbor node.

- **Path1 discovery phase:** In this phase each node has enough amounts of information to calculate the link cost (i.e. cost function between two nodes) to its neighboring node. Then sink node computes best next hop depends upon the cost function and broadcast Route Request (RREQ) message to best next hop. (Figure 2 shows the structure of RREQ message) Similar way best next hop computes its best next hop and send RREQ message to it, the process continue until we reach source node (shown in figure 4)

- **Path2 and Path3 Discovery Phase:** For these two paths sink sends two different RREQ messages to its next best neighboring nodes. To make paths node-disjoint, we allow only one RREQ message to accept by each node. The nodes receive more than one RREQ message receives only first RREQ message and reject all other RREQ messages (see figure 3, for an example of path construction. In this example Node T computes its next preferred neighbour finds it Node M. Pink Node T generates Route Request (RREQ) message and forwards to Node M, but Node M has been included in the path 1, then Node M simply responds to Node T with an INUSE message indicating that Node M is already selected in a routing path. Immediately Node T
searches its neighbouring table and computes the next preferred neighbour, which will be Node Y, and sends out Route Request (RREQ) message to it. Other Node Y accepts the message and continues the procedure in the direction of the source node.

![Diagram of Path Discovery](image)

**Figure 3 Example of Path Discovery**

**D. Path Maintenance**

In case of path maintenance, we can conserve the energy by reducing the overhead traffic through reducing control messages. For this reason to update cost function, we append the metrics on the data message by attaching the residual energy, free buffer, link quality, packet reception ratio, and received signal strength indicator, required number of packet retransmission and average probability of packet reception to the data message, instead of sending a KEEPALIVE message to keep multiple paths alive.

**E. Path Selection Process**

After the path discovery phase and paths constructed between source and sink. To send the real and nonreal traffic between source to sink with some bound of data delivery given by $\alpha$, we select a path for RT (i.e. highest cost among all paths) and two paths for NRT (remaining both paths) from available three paths. We assume that each path has some probability $p_i$ ($i = 1, 2, 3$) of successfully delivering the data messages to the sink. From [30] we have:

$$k = x_{\alpha} \cdot \sqrt{\sum_{i=1}^{2} p_i (1 - p_i)} + \sum_{i=1}^{3} p_i$$

Where $x_{\alpha}$ is the corresponding bound from the standard normal distribution for different levels of $\alpha$. Table I lists some values for $x_{\alpha}$.

**Table I. Some Value for the Bound $\alpha$ [24]**

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>95%</th>
<th>90%</th>
<th>85%</th>
<th>80%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{\alpha}$</td>
<td>-1.65</td>
<td>-1.28</td>
<td>-1.03</td>
<td>-0.85</td>
<td>0</td>
</tr>
</tbody>
</table>

From the out of $k$ paths we use $l$ path to transfer real time traffic and $m$ paths to transfer nonreal time traffic, where $k = l + m$. To calculate $l$ and $m$, we assume that sensor node knows the size of traffic (i.e. real time traffic and nonreal time traffic). Let $t_r$ represents the size of the real time traffic and $t_{nr}$ represents size of the nonreal time traffic, then we can calculate value of $l$ and $m$ as follow [24]:

$$l = \frac{t_r}{t_r + t_{nr}} k$$

$$m = \frac{t_{nr}}{t_r + t_{nr}} k$$

We use $l$ paths that minimize the end-to-end delay to transfer the real time traffic, which ensure that the data in a critical time delivered to the destination within required time. Each sensor node estimates the time required by RREQ packet, to travel from the node neighbor till it is received by the node itself. By taking account this time, the source node calculates end-to-end delay for each of the available paths.

**F. Traffic allocation and Data Transmission**

Our protocol employs the queuing model presented in [12] to handle both real time and nonreal time traffic in the case of coexistence of real time and nonreal time traffic in each of the sensor node. The model we employ is inspired from class-based queuing model. Two different queues are used, one instant priority queue for real time traffic, and the other queue follows the first in first out basis for nonreal time traffic. Figure 4 shows the QoS model.

The source node knows the degree of the importance of each data packet it is sending which can be translated into predefined priority levels. The application layer sets the required priority level for each data packet by appending an extra bit of information to act as a stamp to distinguish between real time and non real time packets. Based on the packet type, the classifier directs packets into the appropriate queue. The traffic allocation scheme first splits up the packets into a number of equal sized sub-packets (or data segments), and then schedules sub-packets simultaneously for transmission across the available multiple paths. Before scheduling the sub-packets, the traffic allocation scheme adds error correction codes to improve the reliability of transmission and it changes the route dynamically to increase the resiliency to paths failures, in case of node failure during transmission of the data and ensure that an essential portion of the packet is received by the destination without incurring any delay and more energy consumption through data retransmission. At the sink node, the parts are collected, reassembled, and the original message is recovered. More details are given below.
G. Data Transfer Across Multiple Paths

Data transfer is done through the two steps: packet segmentation and encoding; data forwarding and Recovery. The details are given below:

1) Data Packet Segmentation and Encoding

We divide each data packet into the equal data segments \((D_0, D_1, D_2, \ldots, D_{N-1})\) of length \(N\), and overhead part of \(M+1\) (Where \(M < N\), we append the error correction codes \((C_0, C_1, C_2, C_3, \ldots, C_M)\) of equal size of data segment to the original message (see figure 5). The data segments and error correction codes have same length of \(l\) bytes and it should be multiple of 8. Correction codes are calculated as a function of the information bits to provide redundant information. We use an XOR-based coding algorithm like the one presented in [25]. This algorithm does not require high computation power or high storage space. The correction codes are computed as follow:

\[
C_0 = D_0 \oplus D_1 \oplus D_2 \oplus \ldots \oplus D_{N-1} \\
C_1 = D_1 \oplus D_2 \oplus D_3 \oplus \ldots \oplus D_M \\
C_2 = D_2 \oplus D_3 \oplus D_4 \oplus \ldots \oplus D_{(M+1) \mod N} \\
C_3 = D_3 \oplus D_4 \oplus D_5 \oplus \ldots \oplus D_{(M+2) \mod N} \\
\vdots \\
C_M = D_M \oplus D_1 \oplus D_2 \oplus \ldots \oplus D_{(M+k) \mod N}
\]

Where \(\oplus\) represents exclusive OR operation.

From the above equation correction codes are generated for both types of traffic (real time and nonreal). \(N = l\) for real time traffic, and \(N=m\) for non real time traffic (See III paragraph of section E).

2) Data Forwarding and Recovery

After computing the XOR error correction codes, the data segments of the original message \((D_1, D_2, D_3, \ldots, D_N)\) along with the error correction codes \((C_0, C_1, C_2, C_3, \ldots, C_M)\) are sent out across the multiple available paths. In [26], authors have proven that if \(M\) or fewer segments are lost out of the \(N + M\) total data and overhead (correction codes), the original \(N\) message segments can be recovered using appropriate linear transformations (in our case through XOR operation). The original message could be regenerated as follow:

\[
D_1 = C_0 \oplus C_2 \\
D_2 = C_0 \oplus C_3 \\
\vdots \\
D_M = C_0 \oplus C_1
\]

Figure 5 shows the packet segmentation and the fields in each segment. We use a fixed segment length, and segments must have a length that is a multiple of 8, to allow proper offset specification. The UP field is an encoded piece of information that represents the current value of metrics used in the cost function to avoid excessive control packets to maintain and keep routes alive. Each node along the path, after updating its neighboring table with this information, changes this value by its current metrics and forwards the sub-packet into its next hop. The Identification field (ID) is a unique identifier assigned to each message being fragmented. The field is used by the sink node to reassemble messages without accidentally mixing segments from different messages. More Segments (MS), this bit is set to 1 for all segments except the last one, which has it set to 0. When the segment with a value of zero in the MS bit is seen, the receiver knows that it has received the last segment of the message. Segment Offset (off), this field solves the problem of sequencing segments by indicating to the receiver where in the original message each particular segment should be placed. The PT field is an extra bit, used to distinguish between real time and nonreal time packets, in order for the packet classifier to forward each packet to its appropriate queue.
H. Node failure case

After computing the XOR error correction codes, the segments of the original message \((D_0, D_1, D_2, \ldots, D_N)\) along with the error correction codes \((E_0, E_1, E_2, \ldots, E_M)\) are sent out across the multiple available paths as shown in figure 7.

While sending data on the path suppose if any node is about to fail then protocol dynamically changes the node in place of failed node and keep sending data on the new path (for example as shown in figure 6 if we are sending data on blue dashed path and node H is about to fail due to its less energy. When its energy goes below certain threshold value it will inform all its neighbours K, G, F, E, and C. the node with highest path cost in the routing table of node H will send Connection Request (CREQ) messages to the disconnected nodes of the path i.e. B and M. once Connection Request (CREQ) messages are accepted by node B and node M, then again path start sending the remaining data. The structure of the Connection Request (CREQ) message is shown in figure 7.

### Table II

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network field</td>
<td>1000m X 1000m</td>
</tr>
<tr>
<td>Number of Sensors</td>
<td>200</td>
</tr>
<tr>
<td>Number of sink/Number of Source</td>
<td>1 / 1</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>75m</td>
</tr>
<tr>
<td>Packet Size (Data + Overhead)</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Sub-packet Size</td>
<td>256 bytes</td>
</tr>
<tr>
<td>Transmit Power</td>
<td>15mW</td>
</tr>
<tr>
<td>Receive Power</td>
<td>13mW</td>
</tr>
<tr>
<td>Sleep mode Power</td>
<td>0.015mW</td>
</tr>
<tr>
<td>Energy Threshold</td>
<td>2 Joules</td>
</tr>
<tr>
<td>Initial Battery Power</td>
<td>100 Joules</td>
</tr>
<tr>
<td>MAC Layer</td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>Max Buffer Size</td>
<td>1024 K-bytes</td>
</tr>
<tr>
<td>Buffer Threshold</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Weights ((\alpha, \beta, \gamma, \delta, \epsilon, and \upsilon))</td>
<td>Respectively 1/1/1/1/1/1</td>
</tr>
</tbody>
</table>

Our simulation consists of a field of 1000m x 1000m and a field consists 200 sensor node randomly placed. All the nodes have identical transmission range of 75m. The sink node is situated at the upper right corner of the simulation field, and the source node is situated on the left bottom corner. Table II shows the simulation parameters.

We investigate the performance of the AQSER protocol in a multi hop network topology. We study the impact of changing the packet arrival rate on end-to-end delay, packet delivery ratio, and energy consumption. We change the packet arrival rate at the source node, and measure the end-to-end delay for both real time and non real time traffic. The real time traffic is set to 20% of the generated traffic. We compare our protocol with the EQSR protocol.

### A. Average End-to-End Delay

The average end-to-end delay is the time required to transfer data successfully from source node to the sink node.

Figure 8 shows the average end-to-end delay of the AQSER and EQSR protocols in case of real time and non real time traffic. In this experiment, we have change the packet arrival rate at the source node, and measure the end-to-end delay for both real time and non real time traffic. From the results, it is clear that AQSER protocol successfully differentiates network service by giving high real time traffic absolute preferential treatment over low priority traffic. The real time traffic is always combined with low end to end delay. AQSER protocol performs better in the case of non real time traffic than the EQSR protocol, because of the overhead caused by the queuing
model the delay of the nonreal time traffic is higher than the real time traffic. Furthermore, for higher traffic rates the average delay increases because the AQSER protocol gives priority to process real time traffic first, which causes more queuing delay for nonreal time traffic at each sensor node.

B. Packet Delivery Ratio

The average delivery ratio is the number of packets generated by the source to the number of packets received by the sink node. In this experiment,

We set the failure ratio to 10% to test the protocols behavior under the presence of path failures. Figure 9 shows the average delivery ratio. Obviously, AQSER protocol outperforms the EQSR protocol in the case of path failures, AQSER uses dynamic routing technique and forward error correction (FEC) to retrieve the original message, which is not implemented in the EQSR protocol. Implementing a FEC and dynamic routing technique in the routing algorithm enhances the delivery ratio of the protocol as well as minimizes the overall energy consumption especially in the case of route failures.

C. Average Energy Consumption

The average energy consumption is the average of the energy consumed by the nodes participating in message transfer from source node to the sink node. Figure 10 shows the results for the energy consumption.

From the figure 10, we note that AQSER slightly outperform EQSR protocol; this is because of the dynamic routing in case of path failure so that this will reduce the control overhead messages so that energy consumption will reduce in that case. From figure 10 we can note that AQSER outperforms the EQSR protocol in case of RT and in case of NRT delay is more this is because overhead induced by the queuing model and error codes computation. However, many studies have concluded that average delay increases as a result of increased energy savings. Thus a minimum tradeoff with delay should be made to reduce the energy expenditure.

To evaluate the energy consumption under path failures, we setup another simulation experiment, where we set the nodes failure ratio to 10% of the total nodes. Figure 11 shows the results. Obviously our AQSER protocol outperforms the EQSR protocol in case of real time traffic. For AQSER protocol, we observe that the energy consumption almost stays the same as in figure 12 except very slight increases at higher traffic rates. This is because AQSER protocol easily recovers from path failures and be able to reconstruct the original message through the use of the FEC algorithm.
V. CONCLUSION

Here we have presented AQSER protocol. This protocol designed for WSNs to provide service differentiation between RT and NRT traffic. Our protocol uses the multi-path with the Forward Error Correction (FEC) technique and dynamic node change to overcome from the node failures without invoking flooding for path discovery. Flooding is important because it consumes energy and consequently reduces the network lifetime.

Our AQSER protocol uses the residual energy, node available buffer size, signal-to-noise ratio, and packet reception ratio, required number of packet retransmission, and received signal strength indicator to predict the next hop through the paths construction phase. AQSER splits up the transmitted message into a number of segments of equal size, adds correction codes, and then transmits it over multiple paths simultaneously to increase the probability that an essential portion of the packet is received at the destination without incurring excessive delay. AQSER protocol handles both real time and non real time traffic efficiently, by employing a queuing model that provides service differentiation. AQSER dynamically changes the node in case of node failure in order to conserve the energy.

Through computer simulation, we have evaluated and studied the performance of our routing protocol and compared it with the AQSER protocol. Simulation results have shown that our protocol achieves lower average delay and higher delivery ratio than the AQSER protocol.

As a future work, we are intended to deeply analyze the performance of our protocol and study the impact of the node failure, network size, path length, and buffer size on the performance metrics.

REFERENCES