Received December 26, 2019, accepted January 15, 2020, date of publication January 20, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968121

Queue and Priority-Aware Adaptive Duty Cycle Scheme for Energy Efficient Wireless Sensor Networks

BASHIR A. MUZAKKARI[1], MOHAMAD A. MOHAMED[2], MOHD F. A. KADIR[2], AND MUSTAFA MAMAT[2]

[1]Department of Computer Science, Northwest University, Kano 700221, Nigeria
[2]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu 21300, Malaysia

Corresponding author: Mohamad A. Mohamed (mafendee@unisza.edu.my)

This work was supported in part by the Ministry of Education of Malaysia under Grant FRGS/1/2017/ICT03/UNISZA/02/1.

ABSTRACT In Wireless Sensor Networks (WSN), energy efficiency is a fundamental issue that requires attention in the design of communication protocols. Energy waste occurs as a result of collision and idle listening. The widely used mechanism for energy saving in WSN is duty cycling. Duty cycling coordinates the sleep/wake-up time of sensor nodes to maximize the network lifetime while achieving specific application goals such as high throughput or low latency. Most existing works focused more on static duty cycle, which cannot guarantee the desired end-to-end delay at varying network conditions. In applications with specified delay requirements, the duty cycle of every node should be adjusted separately at runtime depending on the network conditions to achieve the desired delay and energy efficiency. In this paper, we present an Energy Efficient and QoS-aware (EEQ) MAC protocol with a duty cycle scheme that adapts the node's duty cycle to the queue size and priority class of a packet to reduce the delay of high priority packets and support time-bounded delivery of priority packets. By checking the queue size and the priority class of packets in the message queue of each node, the node determines whether or not to adjust its duty cycle. In this approach, a node increases the length of its active period in the event of high traffic which provides less waiting time for the packets in the queue. The sender node informs the receiver the duration for which it has to stay awake at the beginning of data transmission, both the sending and the receiver's duty cycle is controlled based on the queue length and the priority of the packets. This approach saves energy and lessen packet latency. Finally, extensive simulation experiments were conducted to evaluate its energy performance within different network topologies. Comparisons with the existing energy-aware MAC protocol verified the effect of EEQ on improving the energy efficiency and extending the lifespan of WSNs.

INDEX TERMS WSN, duty cycle, delay, idle listening, energy-efficiency.

I. INTRODUCTION

Wireless Sensor Network (WSN) is an array of sensors with the capability of sensing and transmitting real-time data to the sink [1]. These sensors are battery-powered [2] and set up in a harsh terrain making it difficult or rather impossible to replace their batteries. It is therefore essential to minimize the rate of consumption of the node's energy [3], [4] to achieve a better greater network lifetime [1], [2], [5], [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Nafees Mansoor.

Energy saving approaches in WSN via mechanisms such as energy conservation, energy transfer, and energy harvesting have been suggested in many literatures [7].

Energy harvesting, often referred to as energy scavenging [8], is an energy saving technique used to increase the node's energy from ambient or external sources [9] to prolong the network lifetime. The ambient sources include solar [10], [11], wind [12], [13], thermal [14], and radio frequency (RF) [15], while the external sources include that of electrical, mechanical or human [16]. These sources emit energy to the environment for the purpose of energy harvesting. However, these energy sources exhibit a temporary behaviour since they may not be constantly available, hence the need for an energy buffer such as a battery to store the harvested energy [7], [8].

Energy transfer is another promising technique used to extend the network lifetime, it involves a continuous or on-demand wireless transfer of energy from an energy-rich node to other nodes with energy deficiency [7]. Nevertheless, this technique suffers from too much interference as a result of concurrent data and energy transfer between the nodes.

Energy conservation is the economical and judicious use of energy among nodes in a WSN to achieve optimal functionalities. Energy conservation can be achieved using various techniques such as the minimization of communication costs at the node by using energy efficient MAC and routing protocols [7]–[9].

Energy consumption is considered as the most fundamental issue of WSN, and it is widely affected by the communication-related functionality of the sensor node. Similar to computer, sensor nodes communication can be represented by the infamous seven layers OSI approach. Each layer has different communication functionalities, in that it consumes certain percentage of the node's energy at some rates. In this respect, we can further map the energy consumption to the specific functionalities provided at each layer. Consequently, an energy conservation mechanism can be introduced at different layers of the TCP/IP protocol stack. In fact, energy conservation in radio communication is the most effective way to extend the nodes' lifetime, due to enormous energy consumption during radio operations such as transmission and receiving. Moreover, the MAC layer is identified as the most effective, considering its ability to directly control radio communication [20] which is known to be the top most energy exhaustive operation. The fact that the MAC layer plays a significant role as the coordinator for communication among the nodes, the design choice for MAC protocol is very much dependent upon nodes' and networks' parameters such as energy consumption, packet collision, network lifetime and latency.

The MAC protocols are generally divided into two categories, schedule-based and contention-based MAC protocol. Schedule-based protocol prevents collisions, overhearing and idle listening by managing the receiving and transmitting of data according to a predetermined schedule. This collision-free protocol has some advantages in terms of energy efficiency and packet losses. However, it requires strict time synchronization among nodes. The contention-based protocol, also known as unscheduled protocol comes with no global time synchronization requirement and without the need for central coordination as to who can access the medium and when. To support the energy efficiency requirement, the protocol is responsible for switching the wireless communication module on and off known as duty cycle [1], [4], [5], [17]–[19], periodically by using preamble sensing of low level carrier.

A duty cycle is referred to as the proportion of an active period over the entire operation cycle T_{Cycle}. The active period is a sum of transmitting time $\mathrm{T_{tx}}$ and receiving time T_{rx} while T_{Cycle} is the summation of the sleep period, active period and the idle period T_{idle} [21]. Therefore, the duty cycle (DC) is represented as:

$$DC = \frac{T_{tx} + T_{rx}}{T_{Cycle}} \tag{1}$$

Duty cycling mechanism puts nodes into periodic sleep/wakeup mode. The node's transceiver is turned off in sleep mode, which greatly conserve energy, since the node's energy consumption during wakeup mode is twice more than that of the sleep mode [4].

Many sleep/wakeup mechanisms were suggested in the literature. These mechanisms use either fixed, differential or adaptive duty cycle approaches. A fixed duty cycle uses pre-defined duty cycle values. This technique is characterized by high energy waste as a result of idle listening, collision and over hearing such as in S-MAC [22]. In the differential duty cycle approach, nodes duty cycle is assigned based on the nodes' distances from the base station [23]. The major drawback of this technique is the tendency to assign a larger duty cycle in nodes that are farther from the sink which may result in battery depletion.

In the adaptive duty cycle technique, various metrics such as traffic priority, traffic load, queue size, residual energy, and network topology are used to adjust the node's duty cycle. However, packet transmission in nodes using duty cycle suffers latency which is governed by the delay encountered in the sleep mode [19], [24].

Adaptive duty cycling schemes focus more on achieving energy efficiency while fulfilling some QoS parameters such as throughput and delay [25], [26]. Achieving a low duty cycle results in high energy saving and in most cases, it leads to increased delay [4]. Various research works suggested solutions to achieve energy efficiency and desired delay for WSNs [1], [2], [5], [17], [18], [20], [27]–[30], [24], [31]–[34]. The major shortcoming of these approaches is the trade-off between energy efficiency and other QoS parameters. Also, some of the earlier works were aimed at a guaranteed delay provisioning, but they need a substantial amount of signaling from the neighboring nodes to calculate time delay, resulting in a significant overhead and resource wastage. In addition, they cannot efficiently achieve the end-to-end delay in varied traffic conditions.

In this paper, we propose a scheme that adaptively adjusts the node's duty cycle according to the queue length and the priority class of packets. This scheme concentrates on ensuring energy saving and minimizing delay by optimising the active period, since most of the energy consumption occurs during that period. The contributions of our work are pointed as follows:

1. This paper proposes a scheduling algorithm that uses an optimized Random Early Detection algorithm to provide low queuing delay for priority packets with an exponential weighted moving average to solve the problem of starvation suffered by low priority classes

by keeping the value of the average queue length below the minimum threshold.

2. An adaptive duty cycle scheme is proposed to adapt the node duty cycle to the queue length and the priority class of the packet. In this scheme, node duty cycle value is assigned based on the length of its queue and the packet's priority class.
3. The proposed scheme is validated using numerous experiments conducted under different network conditions to evaluate the significance of energy preservation, extended network lifetime and minimize packet delay.

The paper is structured as follows: Section II reviews the existing literature on different duty cycle approaches. Section III describes the proposed adaptive duty cycle scheme. Sections IV and V present performance evaluation and discussion respectively.

II. RELATED WORKS

S-MAC [22], designed to minimize idle listening, collisions, and overhearing by placing the nodes into listen/sleep periods. The listen periods in S-MAC is fixed while the duration of the sleep period relies upon a predefined application-based duty cycle factor. In S-MAC, the listen period is split into SYNC and Data periods. Throughout the SYNC period, a node receives a SYNC packet from its neighbors and store it. In the data period, exchanges of data packets occur which include a request-to-send (RTS), clear-to-send (CTS), DATA, and acknowledgment (ACK) messages. High latency occurs in S-MAC as a result of its fixed sleep periods; to solve this problem an adaptive listening mechanism was introduced in T-MAC [35].

T-MAC [35] was proposed to improve the energy saving of S-MAC [22], especially under adaptable traffic condition and to solve the S-MAC's fixed duty cycle by prematurely sending nodes back to sleep mode in the absence of any event for a given period known as 'Time Active (TA)' period. T-MAC achieves better energy efficiency compared to S-MAC by reducing collision and idleness since nodes go back to sleep mode in the absence of any activity during the TA period at the detriment of high latency and reduced throughput.

S-MAC [22] and T-MAC [35] are regarded as the baseline protocols in WSN for sleep/wake-up and adaptive duty cycling, they emphasize more on ensuring optimal node duty cycle to extend network lifetime, while trading-off QoS requirements [36], [37].

In DutyCon [38], a feedback controller manages the duty cycle and the end-to-end delay to achieve excellent energy efficiency as well as the desired delay. The duty cycle is controlled in proportion to the node's single-hop delay condition as well as the real packet delay; these are quantified using timestamps. U-MAC [39] is an improvement of T-MAC [35], proposed to provide a balance between energy and latency in WSN. Nodes use utilization function, which is the ratio of the actual two-way communication performed by the node to tune their duty cycle in the whole active period. Nodes suffer long idle listening during their active period in a situation where the utilization function is low. In ADQ [40], a control-based approach is proposed to dynamically adjust the duty cycle interval of a node. A pre-set queue size threshold, constrained to a predetermined value aimed at energy conservation and low end-to-end delay, controls the duty cycle. Both analytic and simulation results validates the efficiency of the ADQ's control-based scheme.

III. PROPOSED DUTY CYCLE SCHEME

In Energy Efficient and QoS-aware (EEQ) MAC protocol, the node's duty cycle is varied based on queue size and the priority class of the packet in the message queue of each node. When a packet from the sensing environment arrives at the node, a classifier checks the class of the packet whether it is of high, medium or low priority and places it in the appropriate queue. The scheduler determines the next packet to send, in this case, high priority packets always get transmitted ahead of medium and low priorities. The scheduler systematically selects high priority packets as long as the queue is not empty and then it continues with the medium followed by the low priority packets, and to ensure energy efficiency, the node's duty cycle is dynamically adjusted according to queue length and the priority class of the packet.

A. DETERMINING THE QUEUE LENGTH

We divide the active time T_{ta} into an equally spaced number of timeslots N given by:

$$T_{ta} = NT_{timeslot} \tag{2}$$

where $T_{timeslot}$ is the time period of one timeslot. T_{ta} is the total time for an active period.

The increase in queuing delay raises the number of time slots in the active period T_{ta} whereas a decrease in delay drops the number of time slots. The queuing delay is directly dependent on the length of the queue and, hence the average queue length of high priority class is estimated using a low pass filter with an exponential weighted moving average [41] given by:

$$Q_{avg} \longleftarrow (1 - fl) * Q_{avg} + fl * q \tag{3}$$

where Q_{avg} is the average queue length of high priority class, q is the instantaneous queue length and, fl is the low-pass filter. The low-pass filter is set to 0.01 to decrease the variability of the instantaneous queue length by slightly fluctuating with time, causing a small delay jitter. The average queue length is kept at minimal by adaptively adjusting the weight of the respective queues, ensuring a slight average queuing delay. The weight is a service rate allocated to each queue during transmission.

The queuing delay is controlled by the Random Early Detection (RED) [42] algorithm. In Enhanced RED as given in Algorithm 1, the Q_{avg} is compared to two thresholds, the minimum threshold th_{min} and the maximum threshold th_{max}, to determine the desired and acceptable queuing delays respectively.

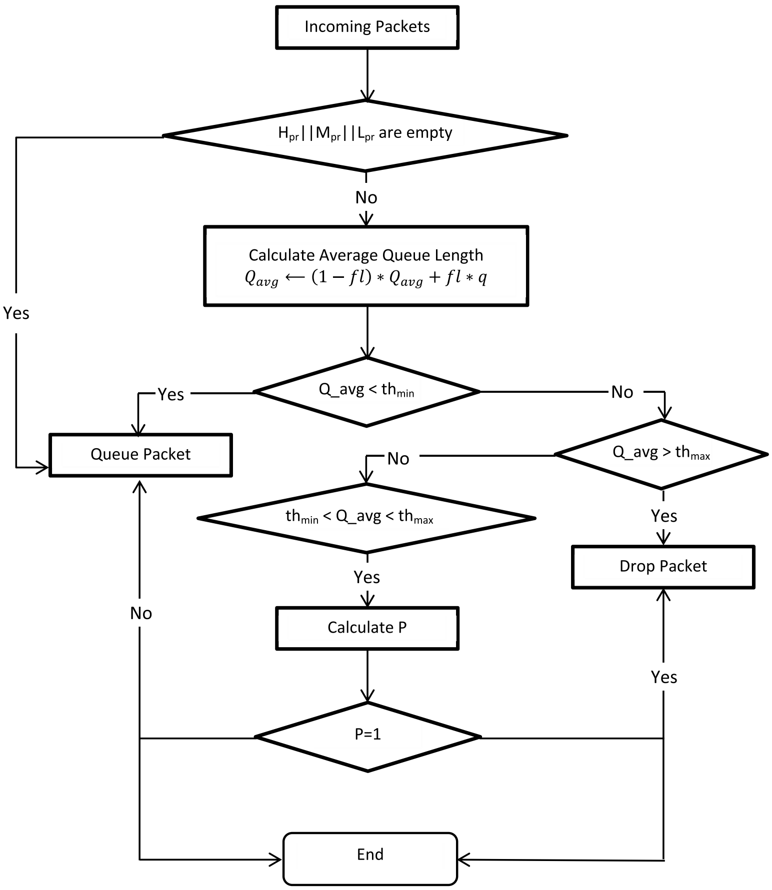


FIGURE 1. **Flowchart of the enhanced RED algorithm.**

Algorithm 1 Enhanced Random Early Detection (ERED) Algorithm

```
Hpr = High priority queue
Mpr = Medium priority queue
Lpr = Low priority queue
Incoming Packets;
if(Hpr || Mpr ||Lpr queue are not empty)
  {
    Calculate queue length
Q_avg⟵(1-fl)*Q_avg+fl*q;
      if(Q_avg < thmin)
        {
          queue the packet;
        }
    else
      if(Q_avg > thmax)
        {
          drop the packet;
        }
    else
      if(thmin < Q_avg< thmax)
        {
          calculate P, if P=1 drop the
packet;
        }
    }
else
queue the packet;
```

If the Q_{avg} is smaller than th_{min}, the packet is queued, and if Q_{avg} is larger than th_{max} then the packet is always dropped. If the Q_{avg} is between the th_{min} and th_{max}, then the newly arriving packet is dropped with some probability P. If the delay of high priority queue exceeds th_{max} the QoS performance degrades rapidly. The flowchart of the Enhanced RED algorithm is shown in Figure 1.

By keeping the value of Q_{avg} below th_{min}, a lower queuing delay is achieved. To achieve this, the weight of the high priority class should be proportionally increased once the Q_{avg} exceeds the th_{min}. However, the weight of the high priority class cannot exceed its upper limit after Q_{avg} reaches th_{max}, otherwise, it will lead to packet clustering.

A linear relationship exists between the weight of the priority class and Q_{avg}. Let us assume the initial weight of a high priority packet is w_{pr}, then the weight function, $f\left(Q_{avg}\right)$ of the high priority class is given by:

$$f\left(Q_{avg}\right)=\begin{cases} w_{pr}, & Q_{avg}\in[0,0.5] \\ \dfrac{\left(upper-w_{pr}\right)*\left(Q_{avg}-th_{min}\right)}{th_{max}-th_{min}}+w_{pr}, & \\ & Q_{avg}\in[0.5,2] \\ upper, & Q_{avg}\in[2,full] \end{cases} \quad (4)$$

where *upper* is the upper limit which the high priority class can reach and Q_{avg} is the average queue length of high priority packets. Assuming the total weight is 1, then $H_{pw} + M_{pw} + L_{pw} = 1$, where H_{pw} is the weight of the high priority class, M_{pw} and L_{pw} are the weights of medium and low priority classes respectively. The upper limit for H_{pw} should be set at 0.7, the rest of the weight to be shared by M_{pw} and L_{pw}.

Since a shared medium has a fixed weight, therefore any increase in the value of w_{pr}, the weights values of the medium and low priority classes must decrease. In this context, the weight of the L_{pw} is shifted to the H_{pw}, if it is not enough and the value of w_{pr} has not reached the *upper* limit, then part of the M_{pw} weight will also be shifted to H_{pw}. However, when the Q_{avg} of the high priority class drops below th_{max}, the weight values taken from the low or medium priority classes will be returned back.

To achieve little or no queuing delay for the high priority class, we set the values of th_{min} and th_{max} to 0.5 and 2 packets respectively. We assumed packets reach the queue in every time slot based on Poisson distribution, with mean arrival rate λ packets per second. All packets are then classified based on their delay requirement. Here, the priority class of the packet is kept in the packet header and queued in the First-In-First-Out (FIFO) buffer for onward forwarding.

To avoid congestion at a node, packet transmission only takes place during the node's active period. Due to the duty cycle operation, a node is only active for a T_{ta} period, given by equation (5),

$$T_{ta} \geq \left[(1+e_i)\, Q_{avg}\delta + (2+e_i)\, \delta_{i,relay}\right] T_{\infty} T_{pkt} \tag{5}$$

where T_{ta} is the total time for an active period, e_i is the packet error rate and $1 + e_i$ is used to estimate the transmission rate, Q_{avg} is the average queue length, δ is the priority class, $\delta_{i,relay}$ is the relayed traffic rate, T_{∞} is a long enough interval, and T_{pkt} is the average period to transmit a packet to other nodes together with medium access overhead.

B. DETERMINING THE PACKET PRIORITY AND DELAY

The delay requirement for a WSN is defined by D_r where $r = 0, 1, 2$, such that $D_0 > D_1 > D_2$, which means the delay requirement for Class 2 is more strict than Class 1 and Class 0 respectively. Therefore, packets that queued in Class 0 are regarded as the low priority, packets that queued in Class 1 have the medium priority and packets that queued in Class 2 have the highest priority denoted by L_{pr}, M_{pr} and H_{pr} respectively.

Packets are placed in their respective queues for transmission to their various destinations. Algorithm 2 presents the priority-based packet transmission. We assume that the delay requirements are set based on the applications' QoS demand.

Once the queue is in a ready state, the classifier in the active node assigns the generated packets into their respective queues, packets in the high priority queue are transmitted in an FCFS sequence and are transmitted ahead of packets in other priority queues. In an event where a medium priority or low priority packet is transmitting and a high priority packet arrived, the transmission is pre-empted for the transmission of the high priority class as depicted in Figure 2. After the transmission, the medium priority packets are pre-empted for the low priority packets to transmit for the remaining timeslot to avoid starvation in the low priority queue.

Algorithm 2 Priority-Based Packet Transmission Algorithm

```
timeHpr = Hpr timeslot
timeMpr = Mpr timeslot
timeslots = Duration of timeslot
wpr = Weight of priority packet
queuestate == ready;
  {
  if(time_Hpr <= timeslot)
    {
     transmit Hpr as FCFS;
    }
     else
  if(time_Mpr <= timeslot)
    {
     transmit Mpr as FCFS;
    }
     else
    {
     transmit Lpr as FCFS;
    }
if(new Hpr packet arrives)
    {
     calculate wpr and insert in Hpr;
    }
     else
    {
       preempt Mpr packets & process Lpr
for the
       remaining time;
       Sleep;
    }
}
```

C. ADJUSTING THE DUTY CYCLE

In this scheme, the node's duty cycle is determined by the average queue length and the priority class of the packet. We introduce a duty cycle measurement DC_m which is used to calculate and assign the duty cycle requirement for a transmission period. We measure the DC_m to check the suitability of the node's duty cycle to the transmission period, given by:

$$DC_m = \frac{D_{ti}}{\left[(1+e_i)\,\delta + (2+e_i)\,\delta_{i,relay}\right] T_{\infty} T_{pkt}} \tag{6}$$

where D_{ti} is the duty cycle time. The duty cycle strive to keep the value of DC_m close to 1 and is calculated every i second. Three conditions determine the possible value of DC_m.

1. ($DC_m \leq 1$) which means the duty cycle is suitable for the current transmission period, therefore, the duty cycle remained unchanged. This occurs mostly either when a node is further away from the sink node or the node does not generate data it only relay data or traffic relay rate is low.
2. ($DC_m = 1$) in this condition, node is assumed to be transmitting generated data for either M_{pr} or L_{pr} traffic as well as packet relay which may cause a little queuing

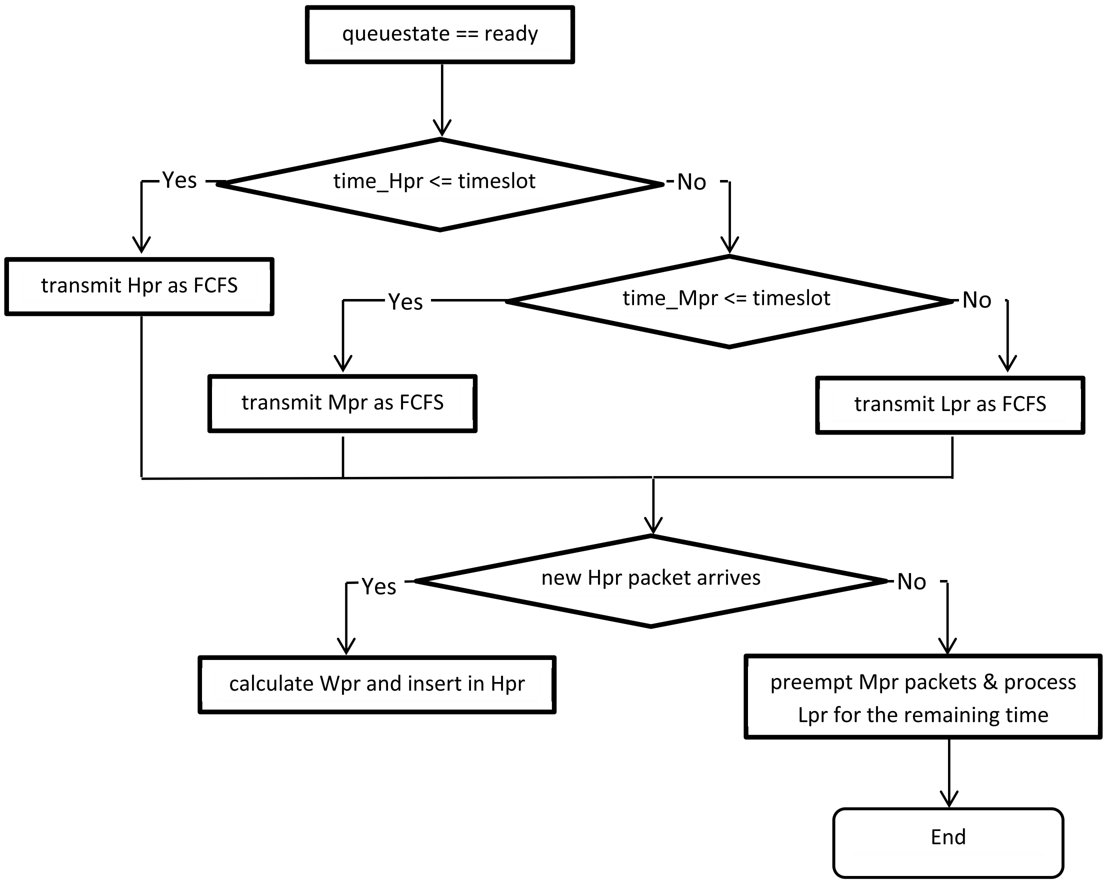


FIGURE 2. Flowchart for the priority-based transmission algorithm.

delay. A rise in queuing delay requires an increase in the duty cycle.

3. ($DC_m > 1$) in this situation a bigger duty cycle is required for the transmission period, this can be due to the node's close proximity to the sink node where the traffic load is high or it is engaged in the transmission of generated H_{pr} traffic.

The DC_m is set with two thresholds, DC_{min} and DC_{max} which denotes the minimum and the maximum duty cycle respectively.

Algorithm 3 Duty Cycle Assignment Algorithm

if ($Q_{avg} < th_{min}$)
  assign DC_{min}
else if ($Q_{avg} > th_{min} || Q_{avg} < th_{max}$)
  double DC_{min}
else if ($Q_{avg} => th_{max}$)
  assign DC_{max}

In a low traffic load condition where the queue length value is less than or equals to the value of th_{min} the default duty cycle value DC_{min} will be assigned, otherwise, the traffic load is assumed to be high. Therefore the duty cycle will be adapted the current traffic condition, to either double the default or assign DC_{max}.

Before sending sync packets, the duty cycle is adjusted based on the average queue length Q_{avg} and the priority class δ of its packets.

D. ENERGY MODEL

The energy consumption of a single node that transmits a packet directly to the sink node is denoted by E_c. It is equal to the sum of E_s and ET_{ta}, where E_s represents the energy spent by the radio in sleep period and ET_{ta} is the energy consumed during the node's active period, such that:

$$E_c = E_s + ET_{ta} \tag{7}$$

The total energy spent by $node_i$ in transmitting data to $node_j$ is represented as E_{ij}. It is equivalent to the sum of E_s and E_{tij}

$$E_{ij} = E_s + E_{tij} \tag{8}$$

where E_{tij} represents the energy consumed by $node_i$ to send a packet to $node_j$. $\sum E_{ij}$ represents the total energy consumption by a single node for transmission in multi-hop communication. The multi-hop communication can achieve energy efficiency only if $\sum E_{ij}$ is smaller than E_c such that:

$$\sum E_{ij} \leq E_c \tag{9}$$

$$\sum E_s + ET_{ta} \geq (E_s + E_{tij}) = E_s + \sum E_{ij} \tag{10}$$

TABLE 1. Simulation parameters.

Parameters	Values
Number of Nodes	40
Network Area	500 * 500
Topology	Grid Topology
Simulation Time	300s
Channel Frequency	2.4GHz
Channel Type	Wireless Channel
Traffic Type	CBR
Queue Size	100
Packet Size	1024KB
MAC Protocols	DutyCon, U-MAC, ADQ and EEQ
Initial Energy	1000J
ET_a	T_x 17.4mA, R_x 18.8mA
E_s	20 µA
DCmin	10%
DCmax	40%
Residual Energy	10J

The energy model is validated using simulation, the energy cost for transmission is calculated based on the values supplied by CC2420 radio chip [43] datasheet for accurate estimation.

IV. PERFORMANCE EVALUATION

We evaluate the performance of the proposed scheme using Network Simulator 2 [44] (NS-2) under different traffic conditions and compare with the existing duty cycling mechanisms. We made the following general assumptions:

1. All sensors in the WSN are homogeneous, having the same initial energy and sensing range.
2. In a multi-hop transmission, nodes can be able to adjust their transmission range to use the least energy required to reach the next-hop node and the sink node. Therefore, energy consumption during transmission is determined by the distance between the source node and the next-hop node.
3. All nodes generate the same amount of data; therefore, packet size is not dependent on the condition of the sensing environment.
4. We assume an ideal transmission condition between all nodes, where every packet reaches its destination successfully.
5. We assume a symmetric radio channel, where the amount of energy required for transmission from node *i* to node *j* is the same amount required for transmission from node *j* to node *i*.

To evaluate the effectiveness of the proposed algorithm we use Network Simulator 2 (NS2) to perform different simulations. We compare the results of the EEQ algorithm with the existing duty cycle schemes such as DutyCon, U-MAC, and ADQ. As shown in Table 1, we set up 40 nodes in 500m × 500m in a grid pattern. Constant bit rate traffic is generated having a packet size of 1024 bytes. All nodes are set with a default duty cycle of 10% each. Packets arrive at the sink according to a Poisson distribution, and the average packet arrival rate is dynamic so as to examine the effect of different traffic conditions. A similar network is considered in all the three-network arrangements having nodes' initial energy equal to 1000 joules. The delay requirement in DutyCon has a pre-set value of 0.7s while for U-MAC, the default duty cycle is set at 10% is the minimum threshold, and 40% is set to be the maximum duty cycle threshold. In the event of high traffic, the duty cycle is set to double the initial, and in very high traffic, the maximum duty cycle is allocated. The limits for high and low traffic conditions are set to 0.3 and 0.1 respectively. We performed the simulation for 5 minutes (300 seconds).

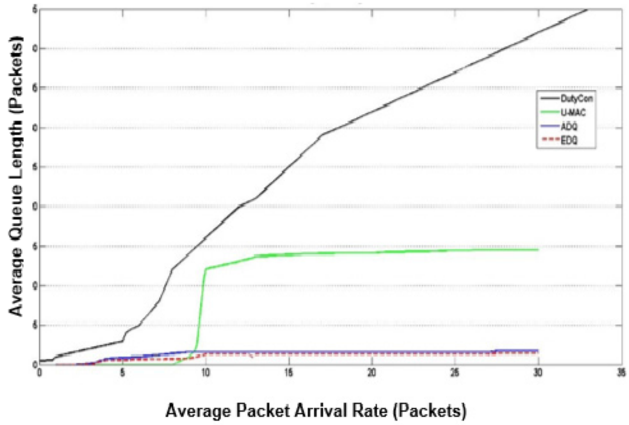


FIGURE 3. Average queue length for single hop topology.

A. SIMULATION RESULTS

The simulation was iterative in order to obtain results with the utmost confidence. The results of the EEQ-MAC duty cycle were compared with the results of DutyCon, U-MAC, and ADQ protocols. The data generated from the trace file is used to conduct the qualitative analysis. The parameters evaluated are average queue length, end-to-end delay, and energy consumption under three different network topologies; single hop, linear multi-hop and multi-hop to test the effectiveness of the protocol in terms of energy conservation and end-to-end delay as well as exposing the effects of multi-hop transmission.

1) SINGLE HOP NETWORK TOPOLOGY

In this topology, nodes connect directly to the sink to send their sensed data packet for further processing. WSN applications such as a traffic surveillance system, healthcare systems, and other delay-sensitive applications rely on single-hop communication to monitor and report events. As sensors in these applications are battery-powered, they can benefit from the scheme's energy efficiency and the quality of service (QoS) control.

The graph in Figure 3 shows that the average queue length for DutyCon rises almost linearly as the traffic load increases. The cause of this increase is because the slack

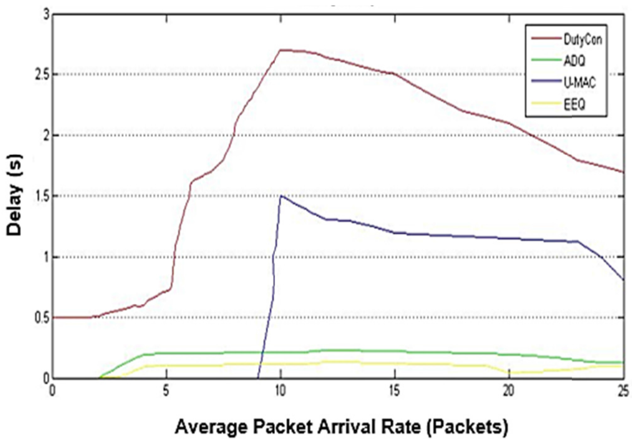


FIGURE 4. Average delay for single hop topology.

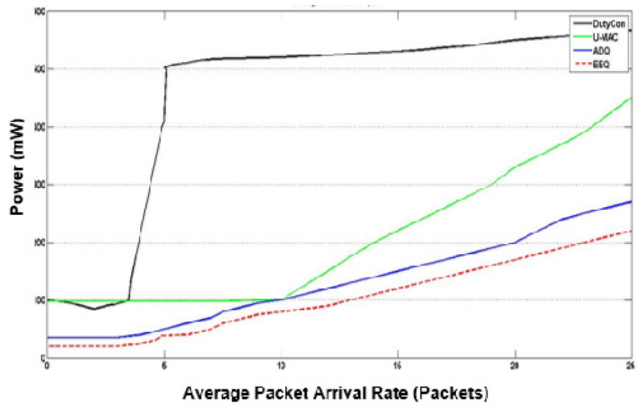


FIGURE 5. Average power consumption for single hop topology.

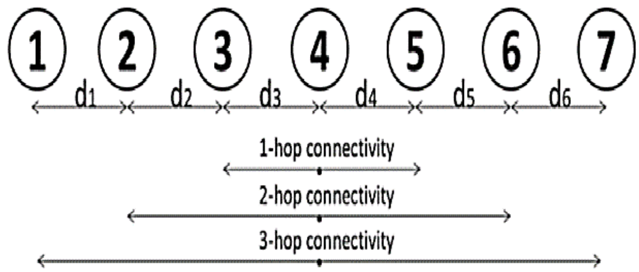


FIGURE 6. Linear multi-hop network topology with 1-hop, 2-hops and 3-hops.

time information controls sleep time in DutyCon. Hence, the inability of DutyCon to adapt to high traffic conditions, causes the queue length to grow. Large queue length means longer waiting time for packets in the queue before being processed which results in additional queuing delay. For U-MAC, the graphs show an unpredictable rapid increase in the queue length before getting it stabilized at around 10 packets per second. ADQ scheme, which works on the adaptive approach efficiently, control the behaviour of the queue and does not let more packets in the queue, however, EEQ scheme indicates more efficiency in controlling the queue regardless of the traffic load. The main reason is the use of a packet priority mechanism and by adaptively adjusting the duty cycle based on the weight of the queue, this ensures a slight average queuing delay.

The result of the average queuing delay for single-hop topology is presented in Figure 4. We calculated the average delay under different network loads by changing the packet arrival rates. The graph shows an acceptable delay response for DutyCon protocol under a low network load. However, as the network load increases from 5 packets per second onward, DutyCon does not adequately control the average delay. Due to this limitation, DutyCon is not suitable for a real-time and delay-sensitive traffic like voice or video communication. U-MAC shows a very low delay until around 9 packets per second of the packet arrival rate, while ADQ and EEQ schemes indicated a larger delay compared to U-MAC. This is as a result of light traffic load, ADQ and EEQ algorithms delays packet transmission until their queue thresholds reaches its minimum value. As the traffic rate increases beyond 9 packets per second, our proposed technique shows a lower average delay, and at the same time, it shows stability and consistent behaviour under high network load. That makes it useful in real-time communication and delay-sensitive video or voice traffic.

The results proved that our proposed solution efficiently manages the duty cycle of the nodes in which they consumed very less energy under low, medium, and high traffic loads. Because of its moderate energy consumption, EEQ increases the network lifetime considerably and thus makes it a reliable candidate to be used in WSN applications where battery replacement is not possible.

Figure 5 shows the average energy consumption graph of all the four algorithms. ADQ algorithm also performed well in this topology because it also controls the duty cycle using adaptive queue management. The rate of power consumption increases linearly in ADQ and EEQ mechanisms because the nodes spend much time in the transmission state. However, simulation reveals that DutyCon and U-MAC protocols are not energy efficient especially under high traffic.

2) LINEAR MULTI-HOP NETWORK TOPOLOGY

In linear multi-hop network topology shown in Figure 6, nodes transmit data to the sink in a linear fashion, in this case, some nodes handle more traffic than the others, which results to fast energy drain that leads to a disconnected network. This topology has become an interesting research field due to its simplicity.

In this section, we will present and discuss the results of linear multi-hop network topology. First, we will show the queue length behaviour of ADQ and our proposed technique, we assumed that data packets are generated from nodes 1, 2, and 3 respectively. We also assumed that packet forwarding occurs through the intermediate nodes to the sink via a three-hop count, that is $H = 3$. Then we will discuss the average delay as we increase the number of hops and lastly we will present the average energy consumption of all the algorithms.

Figure 7 shows the average queue length behaviour for a linear multi-hop network under different network loads.

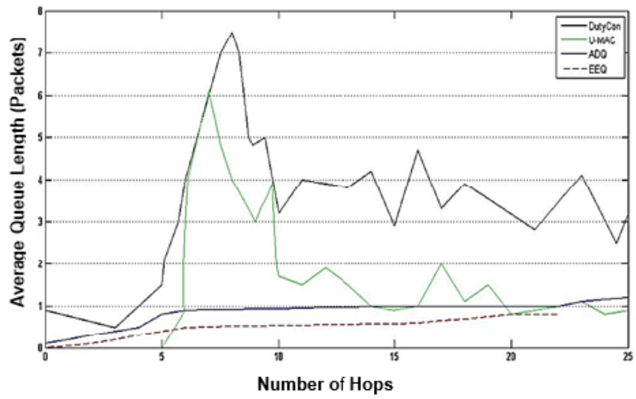


FIGURE 7. Average queue length for linear multi-hop topology.

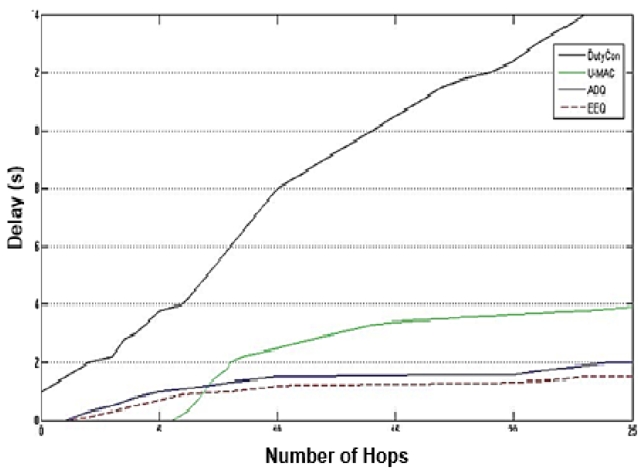


FIGURE 8. Average delay for linear multi-hop topology.

The simulation results of DutyCon and U-MAC algorithms are almost identical since they show a similar pattern under a low traffic load. However, the average queue length is not stable with the increased traffic load as observed earlier in single-hop topology. The unstable average queue length may be due to the design strategy for adapting the sleep interval of nodes to the queuing delay incurred by the increasing rate of an incoming packet. However, this leads to a change in delay response, which will cause inevitable jitter in the network. About the ADQ scheme, our proposed EEQ algorithm performed comparatively better regardless of the traffic load. The reason for maintaining the stabilized queue level by EEQ scheme is the dynamic adjustment of the weight, which not only prevents packet loss likely to be caused by queue overflow but also conserves energy.

Compared to single-hop topology, the average delay in linear multi-hop topology is reduced to half as shown in Figure 8. We quantify the average delay of flows from sources to the sink node. EEQ performed exceptionally well under 10 hops. It can be observed that the delay increases proportionally with the number of hops.

Moreover, the overall average delay pattern of the EEQ algorithm is significantly better than the U-MAC algorithm. Compared to the ADQ scheme, our proposed algorithm

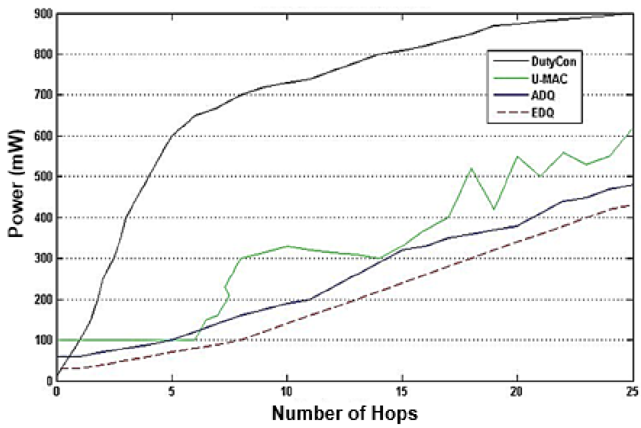


FIGURE 9. Average energy consumption for linear multi-hop topology.

outperformed the ADQ algorithm for network topology comprises of 20 or fewer hops from the source to the sink. This considerable improvement in the average delay is accounted for due to the efficient management of queue size, which reduces unnecessary queuing delay.

Figure 9 shows the average energy consumption for DutyCon, U-MAC, ADQ and EEQ algorithms. DutyCon and U-MAC consume a high amount of energy compared to ADQ and EEQ schemes as a result of the different schedule assignment with different duty cycle for individual node by DutyCon and U-MAC.

The use of ACK packets to piggyback the time of the next sleep leads to asynchronous behaviours of both the sender and the receiver nodes as the number of hops increases. U-MAC shows inconsistent behaviour due to which it becomes unsuitable for energy-constrained networks. The reason for energy wastage in DutyCon and U-MAC is the large queue length they accumulated. The simulation shows that both EEQ and ADQ algorithms effectively saves energy irrespective of network size, this is because of their ability to effectively manage the queuing delay. The proposed algorithm's efficient energy-balancing approach extends the overall lifetime of the network.

3) MULTI-HOP NETWORK TOPOLOGY

In this section, we present the results for the multi-hop. The performance of the proposed scheme is evaluated using three dominant parameters namely average queue length, energy consumption, and end-to-end delay in different traffic conditions. The number of hops is pre-set to H=3 with an assumption that only nodes 1, 2, and 3 generate packets with various packet arrival rates.

Figure 10 above shows plots of the average queue length with different packet arrival rates. A packet in DutyCon and U-MAC queues rises above their respective thresholds which afterward indicated some significant inconsistencies. This inconsistent queuing delay will cause variable end-to-end delay, and thus packet loss may occur. On the other hand, the EEQ algorithm successfully maintains a reasonably small queue size due to which packets have to wait for a very

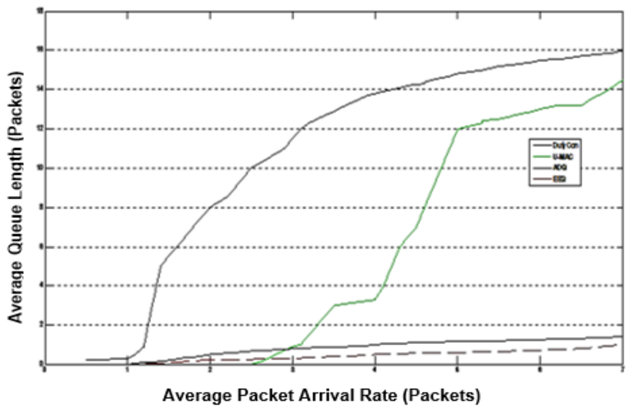


FIGURE 10. Average queue length in multi-hop topology.

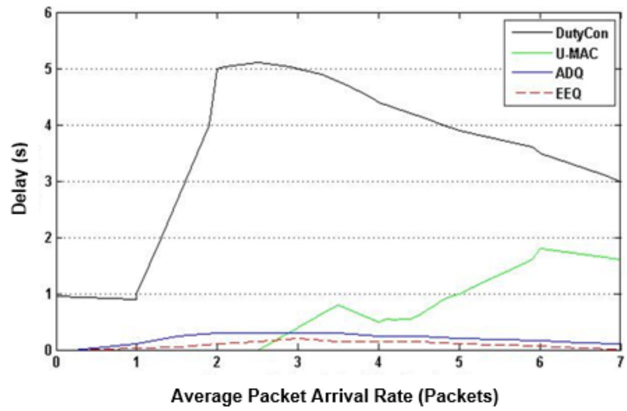


FIGURE 11. Average delay in multi-hop topology.

minimal time in the queue and thus, queue delay becomes predictable. Predictable queue delay can trigger sink nodes to become active and ready to receive the upcoming packet.

The multi-hop network is a complex topology in which there are multiple senders, intermediate nodes, and hops; it can as well have one or more sink nodes. Our proposed algorithm has shown a tendency to adopt varying network conditions while maintaining minimum delay from sender to sink nodes. This is possible due to packet classification schemes based on traffic type and other measurable parameters like the average waiting time in the queue.

Figure 11 shows the average end to end delay in a multi-hop network arrangement. DutyCon algorithm is a major deviant among all the four schemes, and it shows an exceptionally high end-to end-delay failing to cope with the needs of a modern sensor networks. High and unpredictable delays, patterns introduce different challenges in the network, for example, it will hamper the activity and inactivity time of the node.

If a node remains active for a more extended period, it will consume extra battery, and overall it will decrease network lifetime, as shown in Figure 12. Since ADQ algorithm actively monitors specific network parameters, it tends to manipulate queue size, and coordinate sleep and active time of the nodes due to which end to end delay can be optimized according to the network traffic load.

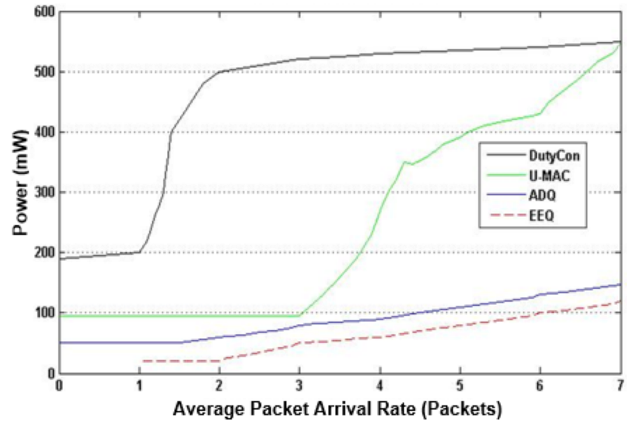


FIGURE 12. Average energy consumption in multi-hop topology.

However, our simulation results show that the EEQ algorithm outperformed ADQ also. The main reason is the smart and efficient management of the nodes' duty cycle which leads to minimum average delay.

Our proposed EEQ algorithm efficiently manages to preserve nodes' energy and thus enhancing over network lifetime. The graph shows that there is a negligible impact on energy consumption with an increasing traffic load. Thus EEQ scheme becomes suitable to operate under high network load with the increasing number of hops also. Energy consumption of all three schemes is considerably higher than the proposed EEQ technique which makes it useful to operate in environments where battery replacement is impossible.

V. CONCLUSION

This paper, presented a dynamic method to adapt the duty cycle to regulate device sleep and wake-up time of sensor nodes to maximize the network lifetime, but at the same time keeping end-to-end delay at the minimum possible level with reasonable queue length. To avoid packets to spend a more extended period in the queue, we develop a queue dispatching technique that checks and picks up the packets based on the priority class assigned to each packet. This approach significantly has reduced queuing delay. It also affects end-to-end delay and enhances coordinated the network lifetime. Simulation results of our proposed algorithm show significant improvement over the existing duty cycle algorithm being utilized. One of the unique points of EEQ algorithm is the scalability factor. Our simulation results show that the EEQ algorithm tends to scale up to meet the growing demands of the network.

REFERENCES

[1] X. Xiang, W. Liu, N. N. Xiong, H. Song, A. Liu, and T. Wang, "Duty cycle adaptive adjustment based device to device (D2D) communication scheme for WSNs," *IEEE Access*, vol. 6, pp. 76339–76373, 2018.

[2] J. Tan, W. Liu, M. Xie, H. Song, A. Liu, M. Zhao, and G. Zhang, "A low redundancy data collection scheme to maximize lifetime using matrix completion technique," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, 2019, Art. no. 5.

[3] Z. Li, J. Gui, N. Xiong, and Z. Zeng, "Energy-efficient resource sharing scheme with out-band D2D relay-aided communications in C-RAN-based underlay cellular networks," *IEEE Access*, vol. 7, pp. 19125–19142, 2019.

[4] F. Wang, W. Liu, T. Wang, M. Zhao, M. Xie, H. Song, X. Li, and A. Liu, ''To reduce delay, energy consumption and collision through optimization duty-cycle and size of forwarding node set in WSNs,'' *IEEE Access*, vol. 7, pp. 55983–56015, 2019.

[5] Q. Li, A. Liu, T. Wang, M. Xie, and N. N. Xiong, ''Pipeline slot based fast rerouting scheme for delay optimization in duty cycle based M2M communications,'' *Peer-Peer Netw. Appl.*, vol. 12, no. 6, pp. 1673–1704, Nov. 2019.

[6] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, ''A survey of network lifetime maximization techniques in wireless sensor networks,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 828–854, Jan. 2017.

[7] F. Engmann, F. A. Katsriku, J.-D. Abdulai, K. S. Adu-Manu, and F. K. Banaseka, ''Prolonging the lifetime of wireless sensor networks: A review of current techniques,'' *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–23, Aug. 2018.

[8] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, ''Energy conservation in wireless sensor networks: A survey,'' *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.

[9] M. Prauzek, J. Konecny, M. Borova, K. Janosova, J. Hlavica, and P. Musilek, ''Energy harvesting sources, storage devices and system topologies for environmental wireless sensor networks: A review,'' *Sensors*, vol. 18, no. 8, p. 2446, Jul. 2018.

[10] P. Bhuvaneswari, R. Balakumar, V. Vaidehi, and P. Balamuralidhar, ''Solar energy harvesting for wireless sensor networks,'' in *Proc. 1st Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jul. 2009, pp. 57–61.

[11] O. N. Samijayani, H. Firdaus, and A. Mujadin, ''Solar energy harvesting for wireless sensor networks node,'' in *Proc. Int. Symp. Electron. Smart Devices (ISESD)*, Oct. 2017, pp. 30–33.

[12] A. Jushi, A. Pegatoquet, and T. N. Le, ''Wind energy harvesting for autonomous wireless sensor networks,'' in *Proc. 19th Eur. Conf. Digit. Syst. Design (DSD)*, 2016, pp. 301–308.

[13] R. K. Sathiendran, R. R. Sekaran, B. Chandar, and B. S. A. G. Prasad, ''Wind energy harvesting system powered wireless sensor networks for structural health monitoring,'' in *Proc. Int. Conf. Circuits, Power Comput. Technol. (ICCPCT)*, Mar. 2014, pp. 523–526.

[14] A. M. Abdal-Kadhim and K. S. Leong, ''Application of thermal energy harvesting from low-level heat sources in powering up WSN node,'' in *Proc. 2nd Int. Conf. Frontiers Sensors Technol. (ICFST)*, Apr. 2017, pp. 131–135.

[15] O. Bjorkqvist, O. Dahlberg, G. Silver, C. Kolitsidas, O. Quevedo-Teruel, and B. Jonsson, ''Wireless sensor network utilizing radio-frequency energy harvesting for smart building applications [education corner],'' *IEEE Antennas Propag. Mag.*, vol. 60, no. 5, pp. 124–136, Oct. 2018.

[16] Z. Wan, Y. Tan, and C. Yuen, ''Review on energy harvesting and energy management for sustainable wireless sensor networks,'' in *Proc. IEEE 13th Int. Conf. Commun. Technol.*, Sep. 2011, pp. 362–367.

[17] W. Qi, ''Minimizing delay and transmission times with long lifetime in code dissemination scheme for high loss ratio and low duty cycle wireless sensor networks,'' *Sensors*, vol. 18, no. 10, p. 3516, Oct. 2018.

[18] X. Liu, M. Zhao, A. Liu, and K. K. L. Wong, ''Adjusting forwarder nodes and duty cycle using packet aggregation routing for body sensor networks,'' *Inf. Fusion*, vol. 53, pp. 183–195, Jan. 2020.

[19] T. Shu, W. Liu, T. Wang, Q. Deng, M. Zhao, N. N. Xiong, X. Li, and A. Liu, ''Broadcast based code dissemination scheme for duty cycle based wireless sensor networks,'' *IEEE Access*, vol. 7, pp. 105258–105286, 2019.

[20] B. A. Muzakkari, M. A. Mohamed, M. F. A. Kadir, Z. Mohamad, and N. Jamil, ''Recent advances in energy efficient-QoS aware MAC protocols for wireless sensor networks,'' *Int. J. Adv. Comput. Res.*, vol. 8, no. 38, pp. 212–228, 2018.

[21] Y. Z. Zhao, C. Y. Miao, and M. Ma, ''An energy-efficient self-adaptive duty cycle MAC protocol for traffic-dynamic wireless sensor networks,'' *Wireless Pers. Commun.*, vol. 68, no. 4, pp. 1287–1315, Feb. 2013.

[22] W. Ye, J. Heidemann, and D. Estrin, ''Medium access control with coordinated adaptive sleeping for wireless sensor networks,'' *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[23] M. Medidi and Y. Zhou, ''Extending lifetime with differential duty cycles in wireless sensor networks,'' in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2007, pp. 1033–1037.

[24] Z. Li, Y. Liu, A. Liu, S. Wang, and H. Liu, ''Minimizing convergecast time and energy consumption in green Internet of Things,'' *IEEE Trans. Emerg. Topics Comput.*, to be published.

[25] W. H. R. Chan, ''Adaptive duty cycling in sensor networks with energy harvesting using continuous-time Markov chain and fluid models,'' *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2687–2700, Dec. 2015.

[26] P. Blasco, D. Gunduz, and M. Dohler, ''A learning theoretic approach to energy harvesting communication system optimization,'' *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872–1882, Apr. 2013.

[27] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, ''Minimizing delay and maximizing lifetime for wireless sensor networks with anycast,'' *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 515–528, Apr. 2010.

[28] K. P. Naveen and A. Kumar, ''Relay selection for geographical forwarding in sleep-wake cycling wireless sensor networks,'' *IEEE Trans. Mobile Comput.*, vol. 12, no. 3, pp. 475–488, Mar. 2013.

[29] S. Yu, X. Liu, A. Liu, N. Xiong, Z. Cai, and T. Wang, ''An adaption broadcast radius-based code dissemination scheme for low energy wireless sensor networks,'' *Sensors*, vol. 18, no. 5, p. 1509, May 2018.

[30] M. Wu, Y. Wu, C. Liu, Z. Cai, N. Xiong, A. Liu, and M. Ma, ''An effective delay reduction approach through a portion of nodes with a larger duty cycle for industrial WSNs,'' *Sensors*, vol. 18, no. 5, p. 1535, May 2018.

[31] Y. Liu, A. Liu, N. Zhang, X. Liu, M. Ma, and Y. Hu, ''DDC: Dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks,'' *J. Netw. Comput. Appl.*, vol. 131, pp. 16–27, Apr. 2019.

[32] X. Xiang, W. Liu, A. Liu, N. N. Xiong, Z. Zeng, and Z. Cai, ''Adaptive duty cycle control-based opportunistic routing scheme to reduce delay in cyber physical systems,'' *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 4, pp. 1–21, Apr. 2019.

[33] Z. Chen, A. Liu, Z. Li, Y.-J. Choi, and J. Li, ''Distributed duty cycle control for delay improvement in wireless sensor networks,'' *Peer-Peer Netw. Appl.*, vol. 10, no. 3, pp. 559–578, May 2017.

[34] H. H. R. Sherazi, L. A. Grieco, and G. Boggia, ''A comprehensive review on energy harvesting MAC protocols in WSNs: Challenges and tradeoffs,'' *Ad Hoc Netw.*, vol. 71, pp. 117–134, Mar. 2018.

[35] D. L. Miller, D. El-Ashry, A. L. Cheville, Y. Liu, S. W. McLeskey, and F. G. Kern, ''Emergence of MCF-7 cells overexpressing a transfected epidermal growth factor receptor (EGFR) under estrogen-depleted conditions: Evidence for a role of EGFR in breast cancer growth and progression.,'' *Cell Growth Differ.*, vol. 5, no. 12, pp. 74–1263, Dec. 1994.

[36] H. Li, N. Jaggi, and B. Sikdar, ''Relay scheduling for cooperative communications in sensor networks with energy harvesting,'' *IEEE Trans. Wireless Commun.*, vol. 10, no. 9, pp. 2918–2928, Sep. 2011.

[37] A. Sultan, ''Sensing and transmit energy optimization for an energy harvesting cognitive radio,'' *IEEE Wireless Commun. Lett.*, vol. 1, no. 5, pp. 500–503, Oct. 2012.

[38] X. Wang, X. Wang, G. Xing, and Y. Yao, ''Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks,'' in *Proc. IEEE Int. Workshop Qual. Service (IWQoS)*, Jun. 2010, pp. 1–9.

[39] S. H. Yang, H. W. Tseng, E. H. K. Wu, and G. H. Chen, ''Utilization based duty cycle tuning MAC protocol for wireless sensor networks,'' in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 6, pp. 3258–3262, 2005.

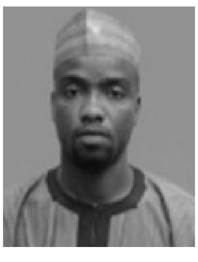
[40] H. Byun and J. Yu, ''Adaptive duty cycle control with queue management in wireless sensor networks,'' *IEEE Trans. Mobile Comput.*, vol. 12, no. 6, pp. 1214–1224, Jun. 2013.

[41] H. Wang, C. Shen, and K. Shin, ''Adaptive-weighted packet scheduling for premium service,'' in *Proc. IEEE Int. Conf. Commun. Conf. Rec. (ICC)*, vol. 6, Nov. 2002, pp. 1846–1850.

[42] S. Floyd and V. Jacobson, ''Random early detection gateways for congestion avoidance,'' *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[43] *ZigBee-Ready RF Transceiver*, IEEE Standard 802.15.4, Chipcon, 2004, p. 96.

[44] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*. Boston, MA, USA: Springer, 2012.



BASHIR A. MUZAKKARI graduated in mobile computing from the Limkokwing University of Creative Technology, Malaysia. He is currently pursuing the Ph.D. degree in computer science with Universiti Sultan Zainal Abidin, Terengganu, Malaysia. He is also a Lecturer with the Department of Computer Science, Northwest University, Kano, Nigeria.

MOHAMAD A. MOHAMED received the Ph.D. degree in mathematical cryptography. He is currently a Lecturer with Universiti Sultan Zainal Abidin. His research interests include both theoretical and application issues within the domains of information security, and mobile and wireless networking.

MOHD F. A. KADIR received the B.Eng. degree (Hons.) in electrical and electronic engineering from Mie University, Mie, Japan, in 2000, the M.Sc. degree in information technology from Universiti Utara Malaysia (UUM), in 2004, and the Ph.D. degree in engineering (system engineering) from Mie University, in 2012. Since 2006, he has been with the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), where he is currently a Senior Lecturer. His main areas of research interest are digital image processing, pattern recognition, computer security, and cryptography. He is a member of the Malaysia Board of Technologists.

MUSTAFA MAMAT received the Ph.D. degree in optimization from UMT, in 2007. He has been a Professor of computational and applied mathematics with Universiti Sultan Zainal Abidin (UniSZA), Malaysia, since 2013. He was a Visiting Professor with Universiti Tun Hussien Onn Malaysia, from 2014 to 2019, a Visiting Professor with Universitas Kanjuruhan Malang, Indonesia, from April 2016 to March 2018, and a Visiting Professor with Universitas Muhammadiyah Ponorogo, Indonesia, from August 2016 to July 2018. To date, he has successfully supervised more than 70 postgraduate students and published more than 260 research papers in various international journals and conferences. His research interest includes unconstrained optimization, such as hybrid conjugate gradient methods, three term methods, Quasi-Newton methods, and chaotic systems. He is currently an Editor-in-Chief for *Malaysian Journal of Computing and Applied Mathematics* (a UniSZA journal in applied science) and an Editor for *Indonesian Journal of Science and Technology*.