electronics

Article

Empirical Analysis of Extended QX-MAC for IOT-Based WSNS

Farhana Afroz * and Robin Braun

School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, Sydney, NSW 2007, Australia
* Correspondence: farhana.afroz-1@student.uts.edu.au

Abstract: The Internet of Things (IoT) connects our world in more ways than we imagine. Wireless sensor network (WSN) technology is at the core of implementing IoT architectures. Although WSN applications give us enormous opportunities, their deployment is challenging because of the energy constraint in sensor nodes. The primary design objective of WSNs is therefore to maximize energy efficiency. Enhancing network quality of service (QoS), such as latency, is another crucial factor, particularly for different delay-sensitive applications. Medium access control (MAC) protocols are of paramount importance to achieve these targets. Over the years, several duty-cycled MAC protocols were proposed. Among them, the strobed preamble approach introduced in X-MAC has gained much interest in IoT field because of its several theoretical advantages. However, X-MAC is highly efficient only under light traffic. Under heavy traffic, X-MAC incurs high per-packet overhead and extra delay. In addition, X-MAC has several design flaws that can significantly degrade network performance. In this paper, we point out some specific malfunctions in the original X-MAC design and propose alternatives to reduce their impact. We present an energy-efficient, traffic-adaptive MAC protocol called QX-MAC that addresses the foreseen shortcomings in X-MAC. QX-MAC integrates Q-learning and the *more* bit scheme to enable the nodes to adapt the active period and duty cycle in accordance with incoming traffic. Finally, the performance of QX-MAC is thoroughly analyzed compared with other reference protocols to validate its efficacy. Our QX-MAC simulation results demonstrate substantial improvements in overall network performance in terms of energy consumption, packet loss, delay, or throughput.

Keywords: wireless sensor networks; MAC protocol; Q-learning; energy consumption; packet loss; delay; throughput

check for updates

Citation: Afroz, F.; Braun, R. Empirical Analysis of Extended QX-MAC for IOT-Based WSNS. *Electronics* **2022**, *11*, 2543. https://doi.org/10.3390/electronics11162543

Academic Editor: Hung-Yu Chien

Received: 29 June 2022
Accepted: 12 August 2022
Published: 14 August 2022

1. Introduction

Sensors are everywhere today. Due to the easy availability of low-power, low-cost microelectromechanical systems (MEMS), wireless sensor networks (WSNs) are widely used in numerous areas ranging from Internet of Things (IoT)-based environment monitoring to smart city, intelligent transportation, healthcare, military applications, and more. The concept of IoT goes in parallel with WSN technology. WSN is basically the key enabling technology of IoT architectures. The IoT is projected to have the entire physical infrastructure connected with information and communication technologies (ICT) through the use of WSN sensors [1]. Even though IoT-enabled WSN applications bring convenience to our daily life, WSNs have some specific design challenges (further detailed in Section 2) [2,3].

A typical WSN is a collection of miniature sensor nodes (also called motes) deployed in an ad hoc fashion in a sensor field. The nodes can sense and collect the environmental data from the surroundings and forward the monitored data to the sink. Users can access the collected data via the Internet. In many scenarios, sensors are deployed in areas where human intervention is not possible. Furthermore, intelligent IoT sensors are generally energy-constrained devices and need to process complex computation [4]. Therefore, the longevity of the WSN is one of the primary design factors. As WSN technology proliferates in different delay-sensitive applications, in addition to high energy efficiency, low latency and high throughput are also expected at the same time. The medium access control (MAC) layer, a sublayer of Layer 2 (data link layer) of the OSI model plays a critical role to achieve these targets.

In WSNs, energy is mainly consumed by wireless communications [5]. Since the sensor nodes use a shared radio medium to communicate with each other, the medium access and its usage need to be controlled in an energy-efficient way [6]. Over the years, a wide range of MAC protocols have been proposed for different purposes; some focus on enhancing energy efficiency, whereas others target on minimizing latency [7]. Irrespective of their targets, energy conservation in sensor nodes is generally achieved by duty cycling the radios. In duty-cycling techniques, each sensor node periodically toggles between active period (radio is ON) and sleeping period (radio is OFF) during its lifetime and the MAC protocol is the one that regulates when to turn on or off the radio.

In the literature, several contention-based duty-cycled MAC protocols (e.g., [8–13]) were designed. Among them, the strobed preamble approach introduced in X-MAC [13] has raised much interest in the IoT field because of its low energy consumption and simplicity. However, X-MAC is only very efficient under light traffic loads. Under high traffic scenarios, X-MAC incurs high per-packet overhead and extra delay [14,15]. To address this issue, we previously proposed a Q-learning-based, traffic-adaptive MAC protocol named QX-MAC in [15]. In addition to the aforementioned shortcomings, X-MAC's original design further has some specific flaws (no evaluation for the scenarios with preambles from simultaneous multiple senders and preambles being unacknowledged) that can significantly degrade network performance in terms of packet loss and energy consumption; further explained in Section 4. Besides, the basic X-MAC design does not support reliable data transmission [16]. In this article, we extend the QX-MAC protocol [15] to overcome these shortcomings and undertake the simulation analysis of extended QX-MAC. All the simulations are done in OMNeT++ 5.5.1 using the INET 4.2 framework. The main contributions of this work are as follows.

1. We identify several malfunctions in the original X-MAC design and propose alternative solutions to them.
2. We model and simulate the extended QX-MAC protocol that integrates the proposed solutions to X-MAC design flaws as well as incorporates data acknowledgment and retransmission mechanisms to resolve the reliability issue of X-MAC.
3. We perform an in-depth empirical analysis of the extended QX-MAC protocol compared with other reference MAC protocols considering two different communication models: single-hop scenario and multi-hop scenario.
4. We emphasize QX-MAC performance gains in terms of energy consumption, and QoS metrics, such as packet loss, end-to-end packet delivery delay, and throughput.

The rest of the paper is structured as follows. In Section 2, we highlight the design drivers for WSN MAC protocols. Section 3 discusses the background works for energy-efficient WSN MAC protocols. Section 4 details the problem formulation and the design and operation of the extended QX-MAC protocol. In Section 5, we report the simulation outcomes and analysis. Finally, Section 6 concludes the paper with final remarks and future works.

2. Design Considerations for WSN MAC Protocols

Though the early research on WSNs put the main focus on low-power, low data rate monitoring applications, recent WSNs are used to support more complex operations (e.g., healthcare and industrial automations), which demand reliable and efficient collection of a large volume of data in a timely manner [17,18]. Current WSN applications require a high probability of success in packet delivery (i.e., high packet delivery ratio), timely packet delivery (i.e., low latency) as well as low energy consumption (i.e., high energy efficiency). However, the delay-bounded and reliable data delivery may demand higher energy consumption, thereby shortening the network lifetime. Therefore, the design of an energy-efficient, QoS-aware MAC protocol becomes more complicated due to the highly resource-constrained nature of motes. The general performance metrics of WSN MAC protocols from the network-specific perspectives include end-to-end packet delivery delay, throughput, packet loss, reliability, energy efficiency, etc. To attain optimum network performance, the impact of the entire protocol stack should be under consideration. In this work, our main consideration is the MAC layer, therefore, we focus on the following factors that can be improved through the MAC protocol design [17].

2.1. Energy Efficiency

Low power consumption is still one of the most crucial factors for battery-powered WSNs. The lifetime of a WSN node depends on its battery lifetime. Therefore, energy must be taken at an extremely low rate from the battery to prolong the battery lifetime. The MAC layer should implement energy-efficient techniques that combat the five major causes of energy consumption, namely idle listening, overhearing, overemitting, collisions, and control packet overhead (for more details, we refer the reader to [15]). A MAC protocol can contribute to higher energy efficiency using collision avoidance methods, reducing control overhead, and tuning the duty cycle of the sensor nodes in accordance with the network traffic. With the collision avoidance techniques, the energy required to retransmit the lost packet(s) can be saved. With adaptive duty cycling, the nodes can adapt their duty cycle based on the incoming traffic load.

2.2. Traffic Adaptivity

As mentioned before, in duty-cycling techniques, each sensor node periodically toggles between the sleeping state and the active state. Since a duty-cycled sensor node in the active state typically consumes a large amount of power from its battery, it needs to be in active mode for the shortest possible duration (i.e., low duty-cycle) without intruding robust communication [19,20]. A low duty-cycled MAC protocol keeps a node in sleep mode for a larger amount of time to preserve energy, however, incurs higher latency and lower throughput, especially under bursty traffic conditions. In the opposite, higher duty-cycled MAC schemes can minimize end-to-end packet delivery delay but demands more energy consumption [21]. Therefore, the fixed duty-cycled mechanism is not suitable to concurrently ensure an optimized energy and QoS performance under dynamic traffic conditions. In such scenarios, a traffic-adaptive duty-cycling MAC solution, which allows each node to adjust its duty cycle in accordance with incoming traffic loads, is necessary.

2.3. Delay

At the MAC layer, the primary causes of the latency are the queueing and contention delay, and the waiting time for a source node to find its destination be ready to receive the data. A MAC protocol should implement an efficient contention mechanism to reduce medium access latency. For mixed traffic scenarios, where different traffic classes are allocated with different levels of priority, delay-aware scheduling algorithms can minimize the latency for high-priority traffic. For multi-hop scenarios, a source node needs to wait until its forwarding node wakes up, thus increasing delay to route the data to the destination. Additionally, for multi-hop WSNs, the performance of the routing protocols also should be considered to reduce the end-to-end packet delivery delay from the source node to the gateway [17].

2.4. Throughput

In WSNs, channel use is mostly regulated by the channel access mechanism [22]. Contention-based MAC schemes can offer high channel use when the contention level is low. However, under high traffic scenarios when the probability of collisions increases, contention-based strategies may lead to poor channel use and high packet loss. Contention-free protocols, such as TDMA-based MAC, can provide high channel use and improved throughput but at the cost of synchronization overhead. Hybrid schemes try to combine the strengths of contention-based and contention-free protocols to achieve better performance under variable traffic conditions, but sometimes add protocol overheads which may increase overall energy consumption.

2.5. Reliability

Packet loss can degrade network performance in terms of reliability [17,22]. In WSNs, packet loss mainly occurs due to collisions and transmission errors. A MAC protocol can assure high reliability in data delivery by implementing collision avoidance methods. It can exploit the acknowledgment mechanism to identify successful transmissions as well as can support retransmissions of the lost packets for packet loss recovery.

3. Related Work

Over the years, MAC protocols of WSNs have been an active research field both from academic and industrial perspectives. Some recent literature surveys of WSN MAC protocols are available in [4,21,23–25]. In this section, we will mainly emphasize the contention-based MAC approaches.

In the literature, contention-based MACs are primarily classified into two categories: synchronous MAC (e.g., [8,10,26,27]) and asynchronous MAC (e.g., [11–13,28]), where the later approach is further sub-divided into two groups: sender-initiated (e.g., [11–13,29]) and receiver-initiated MAC (e.g., [28,30,31]). In the receiver-initiated approach, the receiver wakes up periodically and broadcasts a short beacon to inform the potential senders that it is ready to receive data. One major challenge of this approach is to reduce the waiting period (i.e., idle listening) of the sender while at the same time ensuring it does not miss the wake-up beacon transmitted by the receiver [21]. The fundamental technique used in the sender-initiated approach is low power listening (LPL) or preamble sampling method.

B-MAC [11] is a sender-initiated MAC protocol that uses LPL and an adaptive preamble sampling method to achieve low-power communication. It is the default MAC protocol for TinyOS. In B-MAC, all nodes periodically wake up and sense the medium for the preamble using clear channel assessment (CCA). If no preamble is detected within the CCA period and a node has no packet to send, it goes back to sleep. When a node has data to transmit, it first sends a long preamble for the duration of at least the sleeping period so that the potential receiver can detect it. Following the preamble, the sender sends data. A B-MAC preamble does not carry any addressing information. Instead, the data packet contains the target receiver's address. Hence, all nodes within the sending node's communication range receive the preamble as well as the data packet.

Similar to B-MAC, WiseMAC [12] is also based on the preamble sampling technique. One novel idea employed in this scheme is that at the end of data transmission, the data acknowledgment frame transmitted by the receiver carries its sampling schedules. Knowing this information, the sender can schedule its following data transmission and wake up when the receiver becomes active, thereby reducing the preamble length. This way, energy consumed in sending long preamble, overhearing, and delay is reduced. WiseMAC also uses a *more* bit in the data packet header. If a node has more packet(s) for the same receiver after the current packet transmission, it sets the *more* bit flag to 1 in the current data packet. For frequently changing topology, WiseMAC is costly in terms of per-packet overhead and latency as some sending nodes need to use long preambles in case they fail to communicate with the receiver.

In [13], authors introduced a strobed preamble approach in X-MAC that tries to minimize long preamble duration, overhearing, and per-hop latency. X-MAC is still one of the most energy-efficient MAC solutions for WSNs. In X-MAC, the long preamble is split into a stream of short preambles with small gaps in between to listen for the preamble acknowledgment (PACK) from the target node. Each short preamble carries the target receiver's ID. When the intended receiver detects a preamble, it sends back a PACK to the sender and stays awake until receiving the data packet. On the other hand, the non-intended receivers can switch to sleep mode after receiving a preamble, thereby subsiding overhearing. Upon receiving the PACK packet, the sender initiates data transmission. When the data transmission is over, both the sender and receiver return to sleep. Although X-MAC achieves higher energy efficiency, lower latency, and higher throughput compared to LPL, it is not suitable for high traffic scenarios (further discussed in Section 4).

RL-MAC [32] uses the reinforcement learning algorithm to maximize energy efficiency and throughput. Similar to S-MAC, an RL-MAC node in its active period tries to exchange packets using the sensing/RTS/CTS/ACK method. The key property of RL-MAC is that it dynamically adjusts the active time depending on the node's incoming traffic and traffic characteristics [33]. RL-MAC formulates the active period reservation policy as a Markov Decision Process (MDP). Furthermore, to avoid the early sleeping problem, the sender adds the number of failed attempts to the packet header to inform the receiver about the missed transmissions attempted by the sender.

RI-MAC [28] uses a receiver-initiated approach to shorten the delay. Reference [31], an extension of RI-MAC, employs a pseudo-random sequence to regulate each node's wake-up time, which allows a transmitter to predict its receiver's wake-up time. This way, PW-MAC reduces the duty cycle of the sender as well as the receiver, thereby minimizing the energy consumption. In [30], the author proposed another RI-MAC-based MAC scheme, namely SA-RI-MAC. It uses the receiver-initiated transmission method combined with a sender-assisted contention resolution to improve channel use.

AS-MAC [34] is proposed to minimize contention, overhearing, and latency by asynchronously scheduling the wake-up time of neighboring nodes. The nodes try to shorten the preamble length by storing the wake-up schedules of their neighboring nodes. However, AS-MAC can reduce latency, it consumes more energy for large networks. SLACK-MAC in [35] is mainly proposed for low duty-cycled WSNs. In this scheme, nodes become active in a random manner, record a history of all successful data transmissions with neighboring nodes, and use this log for estimating their following activation time. Two variants of X-MAC, namely XX-MAC and EX-MAC, are modeled in [14] to improve QoS and energy performance compared to X-MAC. XX-MAC is basically designed for heavy traffic scenarios and unsuitable for dynamic traffic conditions. EX-MAC exploits the request-based *more* bit scheme to improve energy efficiency and network throughput under dynamic traffic scenarios.

4. The QX-MAC Protocol

The objective of the QX-MAC protocol is to improve QoS without limiting the energy efficiency of WSNs. QX-MAC combines the Q-learning algorithm and the *more* bit scheme to enhance the overall performance of the network. QX-MAC also deals with all the foreseen flaws in the basic X-MAC design.

4.1. Problem Formulation

QX-MAC primarily aims to overcome the shortcomings of the X-MAC protocol.

1. *High overhead and latency*
   One problem associated with X-MAC is that it incurs high per-packet overhead and extra delay under heavy traffic conditions. This problem mainly arises because X-MAC operates only in unsynchronized state where the intended receiver returns to sleep after receiving each data packet irrespective of whether its most recent sender has more packets destined to it or not. In X-MAC, for each data packet transmission, the sender follows the state transition cycle:
   CCA→SEND_PREAMBLE→SEND_DATA→WAIT_TX_DATA_OVER→SLEEP; and the target receiver follows the cycle:
   CCA→SEND_ACK→WAIT_ACK_TX→WAIT_DATA→SLEEP.
   Since the receiver switches to sleep mode upon receiving a data packet, the same operational cycle is repeated for transmitting each data packet even after the sender has more packets in its queue for the same receiver. This way, resources in terms of energy and time are consumed in transmitting preamble bursts to transfer the subsequent data packet(s).

QX-MAC addresses this issue by implementing an adaptive duty-cycled technique where the sender reserves its active period based on its queue status using Q-learning and the receiver adjusts its duty cycle following the *more* bit scheme.

2. *Reliability issue*
   X-MAC does not implement data acknowledgments. Therefore, a sending node is not aware of whether a transmitted frame has been received successfully.
   QX-MAC ensures reliability by incorporating data acknowledgments and retransmissions.
3. *Other flaws in X-MAC design*

The aforementioned missing details in the X-MAC design (Table 1) and the proposed solutions are further elaborated in the following subsection.

Table 1. Flaws in X-MAC design.

Flaws in X-MAC	Interpretation	Potential Consequence
No evaluation for the scenario with preambles from multiple senders	One sender is sending preambles during the gaps while the other sender is listening for the preamble-ACK (PACK) from the receiver.	Packet loss Energy consumption
	One sender starts sending preambles while the other sender is already in the SEND_PREAMBLE state.	
Preambles end without being acknowledged	One sender starts sending preambles while the other sender has more packets in its queue is waiting for the data-ACK (DACK) from the receiver.	Packet loss Energy consumption
	One sender receives data during the preamble gaps while sending preambles.	

4.2. Flaws in X-MAC Design

4.2.1. Preambles from Multiple Senders

As per the original X-MAC design, when multiple senders simultaneously are in the SEND_PREAMBLE state and a sender (attempting to transmit) detects a preamble transmitted by the other sender, it waits until getting a clear channel; while waiting, if it hears a preamble acknowledgment (PACK) from its target node, it will first back-off a random amount of time to let the initial transmitter to finish its transmission and then transmit its data without any preamble-listen cycle. As such, after receiving a data packet from the first transmitter, the target receiver will stay awake for a duration to receive data from the other transmitter(s) and this extended awake period needs to be equal to the maximum duration of the transmitters back-off period. For simplicity, this technique is called SMT (simultaneous multiple transmitters) in this paper.

Although in X-MAC, the SMT mechanism was designed to provide higher throughput, lower latency, and reduced energy consumption, the advantage of this technique depends on the proportion of transmitters that overhear a PACK. If there are several transmitters waiting to send out, both transmitters and receivers will have a long overhearing period that may impact the energy efficiency. Moreover, more than one transmitter may attempt to transmit at the same time, which eventually will result in packet loss due to collisions. In fact, the real effect of this mechanism on the network performance is not quantified in the original X-MAC paper [6]. In [6], the authors experimentally show that in reality, the SMT mechanism causes a higher loss rate with no significant improvement in throughput and in conclusion recommend not to implement this mechanism. Furthermore, in the INET [36] X-MAC model, the SMT mechanism is not used and no alternative method is proposed to address the SMT issue. Hence, the problem still exists for the scenario when multiple transmitters are in SEND_PREAMBLE state at the same time. For a single-hop star topology, this scenario can have two possible cases.

Case 1: During a preamble acknowledgment gap, a sender, *S1*, fully receives a preamble from another sender, *S2*. In such case, with X-MAC, the two source nodes (i.e., *S1* and *S2*) alternately keep sending preambles until receiving an acknowledgment (PACK) from the target receiver (*R1*) or their preamble duration expires. If *R1* receives a preamble from a particular sender (for example, *S1*), it replies with PACK. Since a X-MAC sending node receives the PACK without comparing the destination address in the PACK header with its own address, both *S1* and *S2* receive the PACK and thus send their data packets. In consequence, at least one packet becomes lost. Thus, for the lost data packet, the resource is consumed for nothing.

For the above particular scenario, one possible solution is: during the listening time (i.e., preamble acknowledgment gap), if a sender receives a preamble of another sender, it returns to sleep in order to avoid packet loss and save energy. This solution has two interpretations. First, If *S1*, for example, is already in the SEND_PREAMBLE state and later *S2* enters into the SEND_PREAMBLE state and fully receives a preamble from *S1*, then *S2* will go back to sleep. In such a situation, the proposed solution can address the packet loss and energy consumption issue as well as ensure fairness. Second, *S1* first enters into the SEND_PREAMBLE state and then *S2* enters and sends a preamble during the listening time of *S1*, which in turn causes *S1* to back to sleep. In such a case, this solution minimizes energy consumption and packet loss at the cost of fairness. Although fairness is an important metric for wireless networks, in WSNs, since all source nodes co-operate for a single application, per-hop fairness can be compromised provided the network's application-level performance is not degraded [8]. Considering all of these facts, in our proposition, the above mentioned solution is implemented.

Case 2: *S1* is sending preambles and meanwhile *S2* starts sending preambles during the listen period of *S1*. However, *S2* initiates preamble transmission at such time instant that its preamble is not completely detected by *S1*. Therefore, both *S1* and *S2* continue sending preambles until their preamble duration expires. As in X-MAC, the sender does not check the header of PACK to verify if this PACK is intended for it, both *S1* and *S2* will receive it. However, the sender nearer to the receiver receives the PACK and thereby starts data transmission earlier than the one located further away. The far located node receives the PACK and sends the data packet a bit later. As a result, the later node causes energy consumption and packet loss due to overemitting.

To overcome the above problem, one simple solution implemented in our proposition is: after a PACK reception, a sender first reads the destination address field of the PACK header and compares it with its own. If the destination address matches with the node's MAC address, it switches to SEND_DATA state; otherwise it goes to sleep, thereby combating overemitting.

4.2.2. Preambles End without Being Acknowledged

As discussed before, in the X-MAC design, when a node has data to transmit, it sends a series of short preambles until receiving a PACK from the target receiver after which it transmits the data frame. However, the X-MAC proposition does not clarify whether data should be sent in case the preamble duration expires without any PACK received by the sender. This situation can arise when the PACK packet is lost or the target receiver is not ready to receive the preamble. In such a case, INET [36] X-MAC implementation considers that data is transmitted in anyway, emphasizing the former possibility. Taking this into consideration, to the best of our knowledge, in the original X-MAC paper or in subsequent papers analyzing X-MAC (with DACK implemented), there is no evaluation for the scenarios with (i) one sender in SEND_PREAMBLE state while the other sender waiting for the data acknowledgment (DACK); and (ii) one sender receives data during the preamble gap in SEND_PREAMBLE state.

Case 1: Let us consider a scenario when a sender, *S1,* is waiting for DACK and in the meantime, another sender, *S2,* starts sending a short preamble as it finds the medium free. If *S1* has more packet(s) to transmit, after receiving DACK, it will send the next packet in line. Therefore, for *S2*, the preamble-listen cycle ends up without any PACK followed by sending the data that will eventually be lost. To address this issue, in this proposition we implemented that when *S2* receives DACK intended for *S1*, it stops sending preambles and turns off its radio after scheduling a random wakeup time.

Case 2: When a node, *S1*, in the SEND_PREAMBLE state receives a data packet during its listening time, it implies that another source node is already in the SEND_DATA state. In that situation, *S1*'s preambles are never acknowledged as the receiver is busy communicating with the other node. This may lead to packet loss from both nodes (due to collisions) or at least from *S1* (due to overemitting), thereby causing energy consumption. One simple solution to overcome this is to allow *S1* to turn off its radio immediately after receiving a data packet that is not destined to it.

4.3. Q-Learning

Reinforcement learning (RL) is a class of solution methods by which an agent learns a policy of how to map states to actions so as a long-term numerical reward is maximized. The agent learns in an interactive environment using a trial-and-error method. In each step of the interaction, the agent receives the current state of the environment, chooses an action from the action space, generates an output, and receives a reward. A reinforcement learner must explicitly explore all actions to discover the optimal action (i.e., the action that gives the highest reward for a given state) [37–39].

Q-learning [40] is one of the most widely used RL algorithms in which an agent learns how to act optimally in Markovian domains. The goal of an agent is to find the best actions that maximize the total reward. In Q-learning, the agent is trained up through a series of distinct stages called episodes. In each episode, the agent observes the current state of the environment, s, chooses and performs an action, a, observes the new state, $s^{'}$, receives a reward, r, and updates the Q-value according to the Bellman equation:

$$Q^{new}(s,a) \leftarrow Q(s,a) + \lambda[r + \varphi\, maxQ(s^{'},a^{'}) - Q(s,a)] \tag{1}$$

In (1), $Q(s,a)$ is the current Q value at state s when action a is taken and $Q^{new}(s,a)$ is the updated Q value for (s,a) pair. The discount factor, φ $(0 < \varphi < 1)$, provides preferences either to long-term rewards $(if\ \varphi \rightarrow 1)$ or immediate rewards $(if\ \varphi \rightarrow 0)$. The learning rate or step size, λ $(0 < \lambda < 1)$, defines to what extent new learned value replaces the old value. The term $a^{'}$ denotes all possible actions for new state $s^{'}$, $maxQ(s^{'},a^{'})$ denotes the maximum Q value at state $s^{'}$ and $\varphi\, maxQ(s^{'},a^{'})$ represents the discounted estimate of optimal future value. Q-learning stores Q values for all states in a states-actions table called Q-table. Initially, Q-table values are taken arbitrarily, usually set to zero.

Epsilon-Greedy Method

In the Q-learning algorithm, the epsilon-greedy method is used for choosing an action for a given state. This method makes use of the exploration-exploitation trade-off to find the optimal Q-table. During learning, the agent needs to explore all possible actions by randomly choosing them. As the training progress and the Q values nearly converge to their optimal values, the agent can act greedily by choosing the action with the highest Q value for a given state.

In the epsilon-greedy method, the probability of exploring actions, also called the exploration rate "epsilon" (ε), is initially set to 1 followed by progressively reducing it so that the agent can exploit its past experience (see Figure 1). In an episode, E, for a given state, s, an action is chosen as per Equation (2).

$$Chosen\ action = \begin{cases} \underset{a}{max}Q(s,a); & if\ r > \varepsilon \\ Random\ a; & if\ r < \varepsilon \end{cases} \tag{2}$$

where, r is a random number on the interval (0, 1) and $\underset{a}{max}Q(s,a)$ is the action corresponding to the maximum Q value. The epsilon is reduced as per Equation (3).

$$\varepsilon = \varepsilon_{min} + (\varepsilon_{max} - \varepsilon_{min}) * exp(-\gamma * E) \tag{3}$$

where, ε_{min}, ε_{max} are the minimum and maximum epsilon, respectively, and γ denotes the decay rate. The decay rate defines the rate at which ε will decay [39]. We use an exponential decay rate by which the probability of exploration reduces as the episode proceeds [41,42]. In our model, we set ε_{max}, ε_{min}, and γ to 1, 0.05, and 0.00001, respectively.

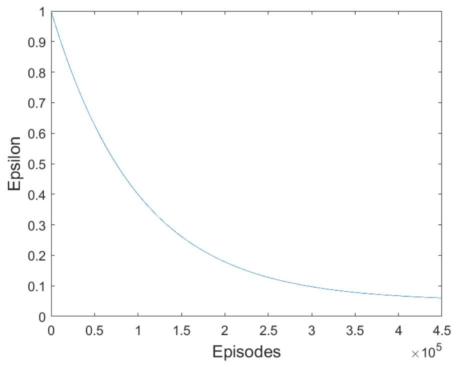


Figure 1. Variation in epsilon value over time as the training progress. This plot results from an epsilon decay rate of 0.00001.

4.4. QX-MAC Operation

The QX-MAC protocol employs the strobed preamble mechanism in which the long preamble used in basic preamble sampling protocols is replaced by short preamble bursts. A key feature of QX-MAC is that it dynamically adjusts both the sending and receiving node's active period and duty cycle according to the sender's traffic loads.

The frame header structure of QX-MAC is depicted in Figure 2a. Similar to WiseMAC, the QX-MAC frame header contains an extra control field carrying a single bit (*more* bit) flag to indicate the sender's queue status (i.e., the number of packets in the queue). When the sleep time is over, a node wakes up and enters into the CCA state to check the activity in the medium. If no preamble is detected during the CCA period, the node checks its queue status after CCA timer expires. If the node's queue is not empty, it moves to SEND_PREAMBLE state and starts transmitting a series of short preambles with small gaps in between to listen for the preamble-ACK (PACK) from the target receiver. To receive a data packet, a potential receiver must need to listen to a complete preamble. After detecting a short preamble, the intended receiver sends back an acknowledgment (PACK) and remains awake to receive the upcoming data packet from its sender. On receipt of the PACK, the sending node stops sending short preambles and switches to the SEND_DATA state to transmit a data packet. If the sender has more packet(s) in its queue for the same receiver, it sets *more* bit to 1 in its data frame, thereby signaling the receiver to extend its active mode for the next data packet from it, as shown in Figure 2b.

After transmitting a data packet, the sender waits for the data acknowledgment (DACK) from its receiver. If DACK is not received within a certain duration, the sender retransmits the data packet, thereby enhancing the reliability. In case of successful data transmission (i.e., DACK is received), at the sending end, the sender reserves its active time (T_r) to transmit its subsequent data packet(s) through the Q-learning algorithm. The sender considers the queue status, i.e., the number of packets queued for transmission as the *initial state*, chooses an active period (*action*) from the action space using the epsilon-greedy method, and sequentially transmits the next data packet(s) within the allocated active period. When the reserved active period is over, the sender observes its new queue status (*new state*). The *new state* actually represents the quality of the action taken. Based on the *new state*, a reward is given and Q-table is updated according to Equation (1). A positive reward (+1) is given if the queue is empty; otherwise, a negative reward (−1) is provided, as depicted in Figure 3.

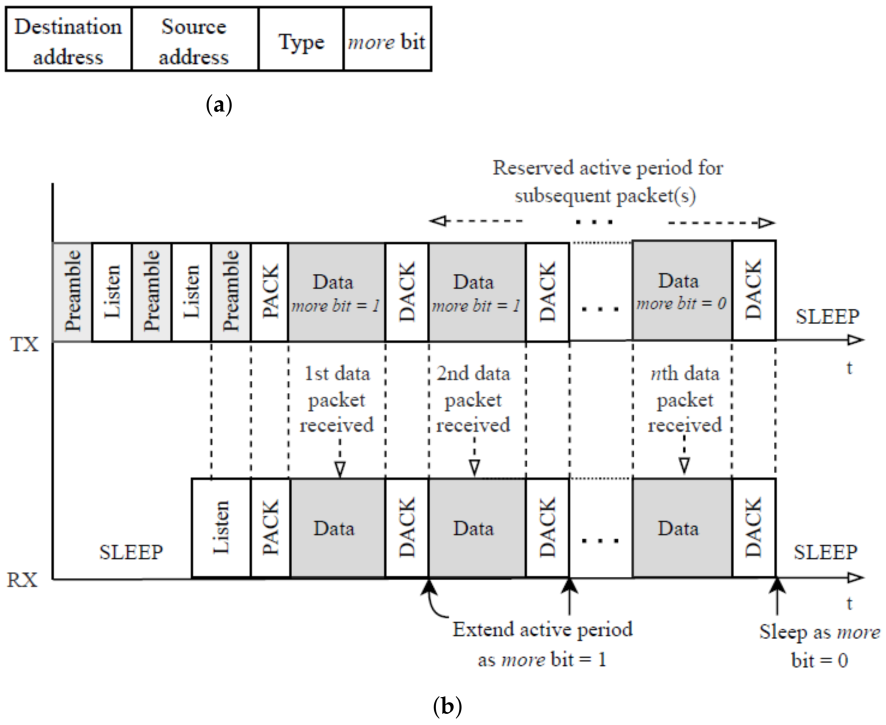


Figure 2. Overview of the proposed QX-MAC protocol. (**a**) QX-MAC frame header structure; (**b**) An example of data transmission from a transmitter (TX) to the receiver (RX) using QX-MAC.

At the receiving end, after receiving each data packet, the target receiver reads the *more* bit in the frame header. It also sends back DACK to acknowledge the receipt of the data packet. When DACK transmission is over and it reads the *more* bit = 1, instead of going back to sleep, it extends its active period, as depicted in Figure 4. With the *more* bit set to 1, the target receiver and the sender are synchronized with each other. Therefore, the transmitter can transfer the following data packet in line straight away. Every time a sending node transfers a data packet, it sets or clears the *more* bit in the frame header based on its queue status. Hence, the receiver can decide whether to remain active or turn off its transceiver. Whenever the receiver finds the *more* bit = 0, indicating the current sending node has no more packet to transmit, the receiver switches to SLEEP state. The sender also moves to sleep mode once data transmission is over.

In QX-MAC, since each preamble packet carries the intended receiver's ID, the non-intended receivers (i.e., overhearers) can turn off their radio after receiving a preamble. After the first data packet transmission, if the sender has more packet(s) destined not for the last target receiver but for a different receiver, the former will behave like an overhearer. In the context of multiple sending nodes concurrently in SEND_PREAMBLE state, if a sending node fully receives a preamble from another sending node, it returns to sleep to avoid packet loss and energy consumption. During the SEND_PREAMBLE state, if a sender receives a PACK packet whose destination address does not match with its own MAC address, it turns off its radio to avoid the collisions. In case of a sender in SEND_PREAMBLE state receives DACK intended for other node, it halts preamble transmission and switches to SLEEP mode after scheduling a random wake-up time.

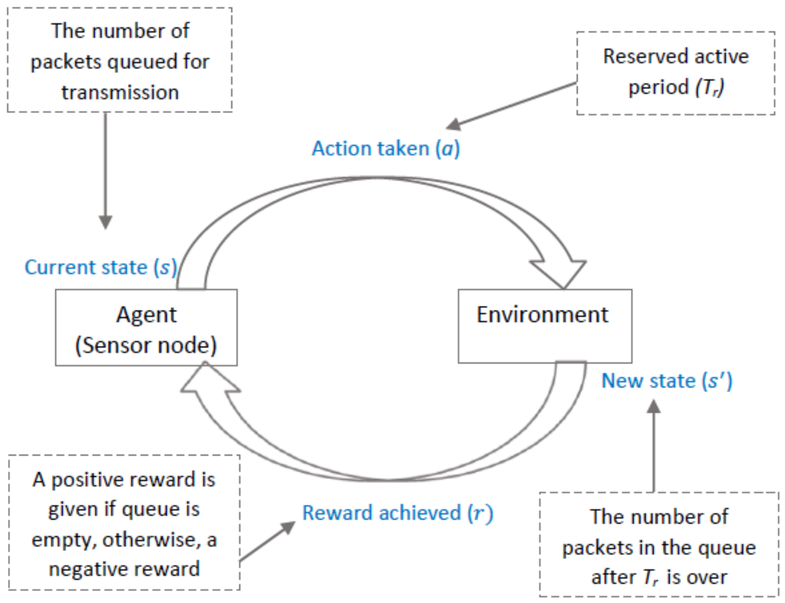

The number of packets queued for transmission
Reserved active period (T_r)
Action taken (a)
Current state (s)
Agent (Sensor node)
Environment
New state (s')
A positive reward is given if queue is empty, otherwise, a negative reward
Reward achieved (r)
The number of packets in the queue after T_r is over

Figure 3. Q-learning to reserve active period.

QX-MAC Energy Consumption

To analyze the energy consumption in transmitting or receiving a data packet, let us assume P_{tx}, P_{rx}, and P_s are the power consumption to send, receive and sleep, respectively; T_l and T_s represent the periodic listen duration and sleep duration of a node, respectively; T_p and T_{pack} denote the short preamble duration and preamble-ACK duration of the preamble-listen cycle; T_{dack} is the data acknowledgment duration; and T_h and T_d are the header duration and data duration, respectively. T_d includes header duration and message duration.

Similar to X-MAC, QX-MAC has periodic listening as its background energy consumption. The energy consumption due to periodic listening, E_{pl}:

$$E_{pl} = P_{rx}T_l + P_sT_s \tag{4}$$

To detect a transmission in the medium, the periodic listen duration should follow the below condition:

$$T_l \geq T_a + 2T_p \tag{5}$$

The minimum T_l is considered in our computation because this is sufficient to detect a transmission. If a sender *S1* has more than one packet in its queue to transmit to a target node *R1*, the energy consumption to transmit the first data packet (unsynchronized state):

$$E_{tx}^{unsyn} = \left(\frac{(T_s + T_p)^2}{2(T_l + T_s)(T_p + T_{pack})} + 1\right)\left(P_{tx}T_p + P_{rx}T_{pack}\right) + P_{tx}T_d + P_{rx}T_{dack} \tag{6}$$

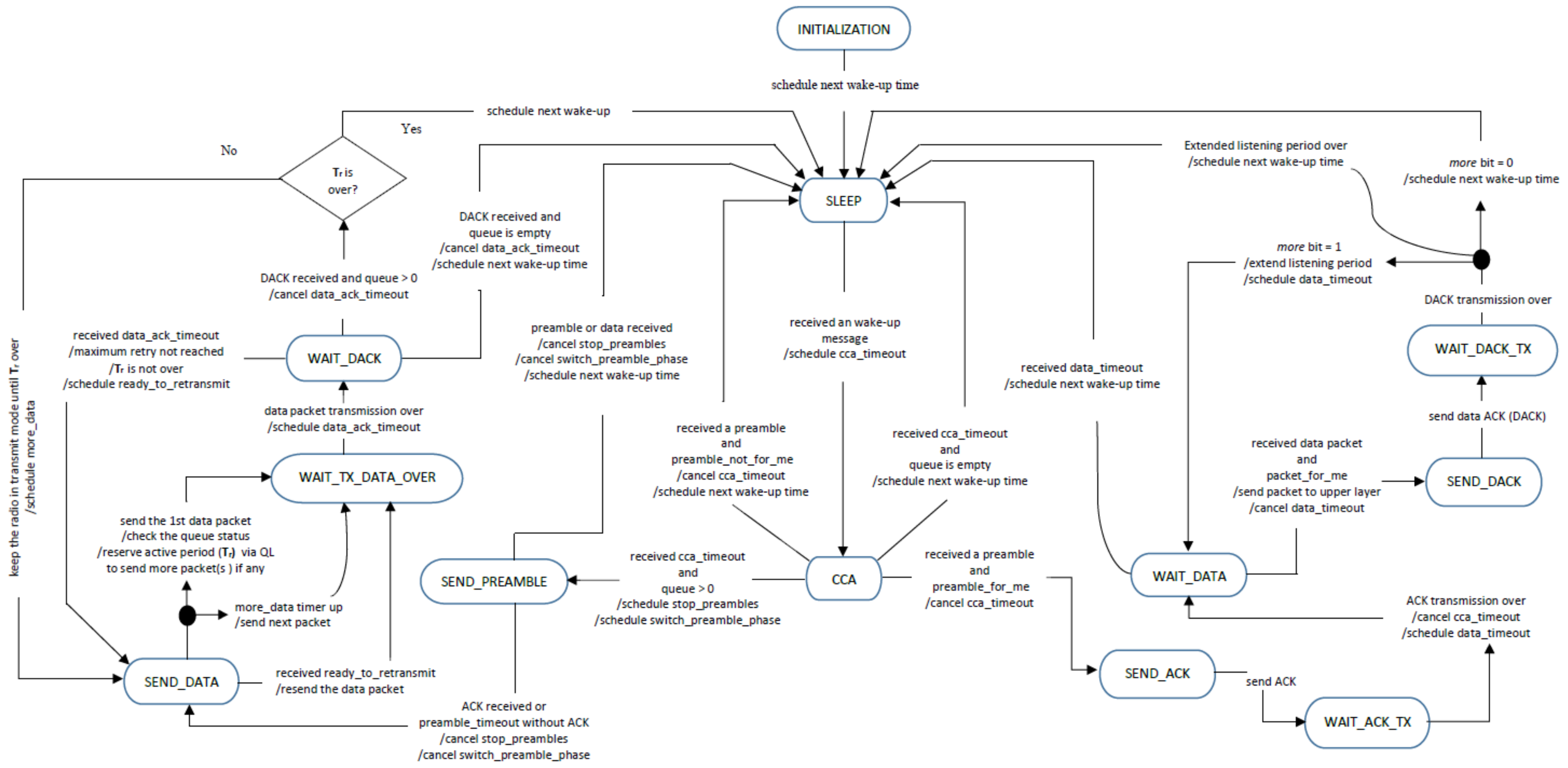


Figure 4. Finite state machine (FSM) of proposed QX-MAC.

In Equation (6), $P_{tx}T_p + P_{rx}T_{pack}$ represents the energy consumed for a single preamble-listen cycle. For sending a data packet, minimum one such cycle is required, therefore the +1 in Equation (6). Additional cycles might be needed based on the wake-up time of the target receiver. $\frac{(T_s+T_p)^2}{2(T_l+T_s)(T_p+T_{pack})}$ represents the average number of extra cycles. For X-MAC, the energy consumption to send each data packet is similar to Equation (6) [29]. However, for QX-MAC, the energy consumption to transmit each subsequent data packet from node *S1* to *R1* (synchronized state):

$$E_{tx}^{syn} = P_{tx}T_d + P_{rx}T_{dack} \tag{7}$$

In order to receive the first data packet from *S1*, in addition to E_{pl}, *R1* needs to consume extra energy. This additional energy does not have the energy spent in receiving a short preamble as is already counted within E_{pl}. Hence, the extra energy needed to receive the first data packet (unsynchronized state):

$$E_{rx}^{unsyn} = P_{tx}T_{pack} + P_{rx}T_d + P_{tx}T_{dack} \tag{8}$$

The energy consumption to receive each subsequent packet(s) from node *S1* (synchronized state):

$$E_{rx}^{syn} = P_{rx}T_d + P_{tx}T_{dack} \tag{9}$$

5. Simulation Results and Analysis

To evaluate the performance of QX-MAC, several simulations are carried out in OMNet++ 5.5.1 [43] with the INET 4.2 framework [36]. We simulate the QX-MAC protocol considering a single-hop model that exists in a star topology, as well as a multi-hop communication model, which is present in a linear topology. In both cases, we use UDP as the transport layer protocol.

5.1. Single-Hop Scenario

In the following two subsections, we investigate QX-MAC performance with different (i) packet arrival rates (Section 5.1.1) and (ii) number of nodes (Section 5.1.2).

5.1.1. Effects of Packet Arrival Rate

In this experiment, we evaluate the performance of QX-MAC compared to the EX-MAC, X-MAC, and the B-MAC protocols while increasing the packet inter-arrival time from 0.1 s to 1 s with 0.1 s increment. We consider a single-hop scenario where all nodes are arranged in a star topology with the sink node or gateway (GW) positioned at the center. For all nodes, packet arrival follows a Poisson process with exponential inter-arrival time between packets. Each sensor node transmits a 10 byte UDP packet in every 0.5 s to the sink, which eventually forwards each of the received data packets to a server via a wired connection. Since QX-MAC, EX-MAC, and X-MAC design allow setting different slot durations for different nodes, we set the sink slot duration 5.55 times shorter than that of the sensor nodes because it needs to receive and relay data packets from all the sensor nodes. The slot duration parameter basically defines nodes' sleep duration. Table 2 summarizes the key simulation parameters for all four protocols. To obtain simulation results, for each protocol, at each considered packet arrival rate, we perform five simulation runs with different random seeds. For statistical accuracy, five runs are then averaged.

Table 2. Simulation Parameters.

Approach	Parameter	Value
Common	Header length	2 bytes
	Message length	10 bytes
	Preamble duration	0.83 ms
	Bit rate	19,200 bps
	Queue capacity	20 packets
	Send interval	0.5 s
	GW slot duration	45 ms
	Transmit power	2.24 mW
	Receiver sensitivity	−100 dBm
	Simulation time	1000 s
QX-MAC	Learning rate (λ)	0.5
	Discount factor (φ)	0.618
	Decay rate (γ)	0.00001
	Maximum transmission attempts	2 retries

Packet Loss

For packet loss, we first record the ratio of the total number of packets received to the total number of packets sent [36] and compute the packet loss as below:

$$Packet\ Loss = 1 - \frac{Total\ number\ of\ packets\ received}{Total\ number\ of\ packets\ sent} \tag{10}$$

As observed in Figure 5, with QX-MAC, the server receives the maximum number of data packets for all considered packet inter-arrival times. The effect of the packet arrival rate on the network packet loss is shown in Figure 6. As seen, the network achieves the minimum packet loss with QX-MAC and the maximum loss with B-MAC. With all four MAC solutions, on average, the packet loss remains steady at a higher level until the inter-arrival time of 0.5 s; afterward it slowly decreases with the further increment of the inter-arrival time. Up to the inter-arrival time of 0.5 s, when the packet arrival rate exceeds the service rate (i.e., inter-packet arrival time $>$ send interval), the network suffers from higher packet loss because of congestion. However, Figure 6 also clearly demonstrates that in the inter-arrival time range [0.1 s–1 s], on average, QX-MAC can carry 99.74% of traffic, while EX-MAC, X-MAC, and B-MAC can transfer a maximum of 92.18%, 88.5%, and 67.3% of traffic, respectively. This is due to the fact that in the synchronized state, QX-MAC enables the transmitter to directly transmit all of the data packets in line to the receiver; therefore, within the simulation time, more packets are delivered to the receiver. The retransmission mechanism and the methods used in QX-MAC to address the flaws in X-MAC design also contribute to the very low packet loss with QX-MAC.

Energy Consumption

To quantify the per-packet energy consumption, for each protocol, we record network total energy consumption (sum of energy consumption of all sensor nodes and the sink node) followed by dividing it by the respective total number of received packets.

$$Energy\ Consumption = \frac{Sum\ of\ energy\ consumption\ of\ all\ nodes\ and\ sink}{Total\ number\ of\ packets\ received} \tag{11}$$

In Figure 7, it is depicted that QX-MAC significantly improves network performance in terms of energy consumption per received packet. For all considered packet arrival rates, the per-packet energy consumption for the QX-MAC protocol stays almost at the same level of around 3.12 mJ (on average). QX-MAC, on average, offers an energy savings of 5.31%, 24.81%, and 80.33% over the EX-MAC, X-MAC, and the B-MAC protocols, respectively. This is because QX-MAC allows the source node and target node to adjust their duty cycle in accordance with the source node's queue status, thereby saving energy. In the synchronized state, (i) a QX-MAC sender does not require to consume any additional energy for the preamble-listen cycle(s) in order to transfer the following data packet(s), and (ii) the intended receiver extends its active time knowing the information that the sender has more packet(s) destined to it which in turn implies that the receiver can receive the subsequent data packet(s) straightway (i.e., idle listening issue is drastically reduced). Furthermore, QX-MAC addresses the other flaws in X-MAC that cause energy consumption. For all these reasons, in terms of energy efficiency, QX-MAC outperforms the other three reference protocols even though it implements data acknowledgment and retransmissions.

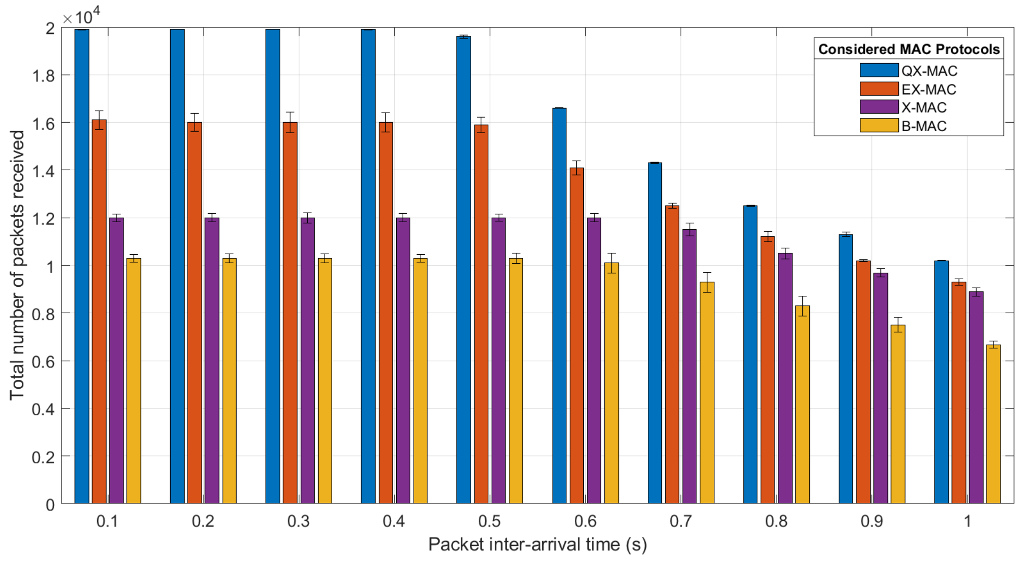


Figure 5. Total number of packets received by the sink with varying traffic for the single-hop scenario.

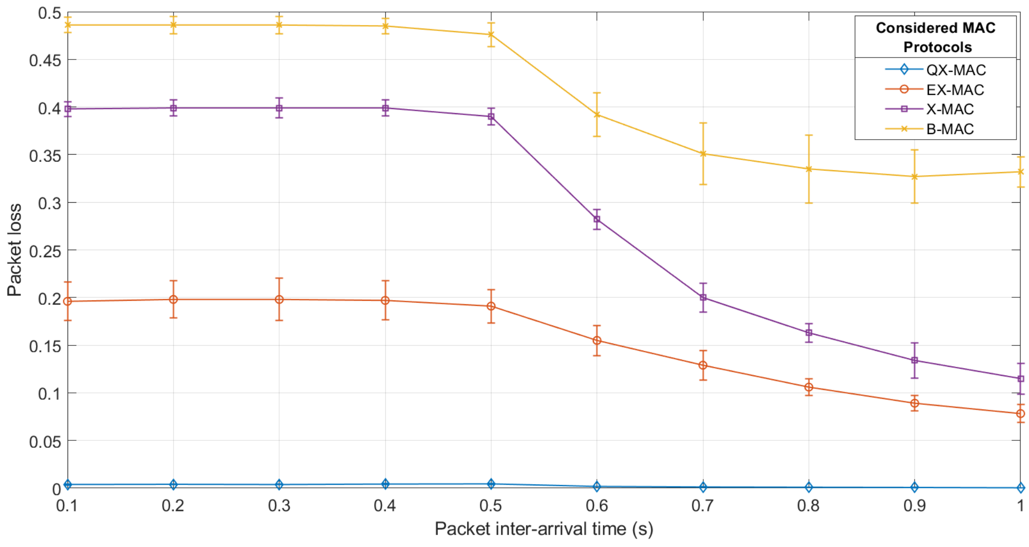


Figure 6. Packet loss with varying traffic for the single-hop scenario. For details, see Equation (10).

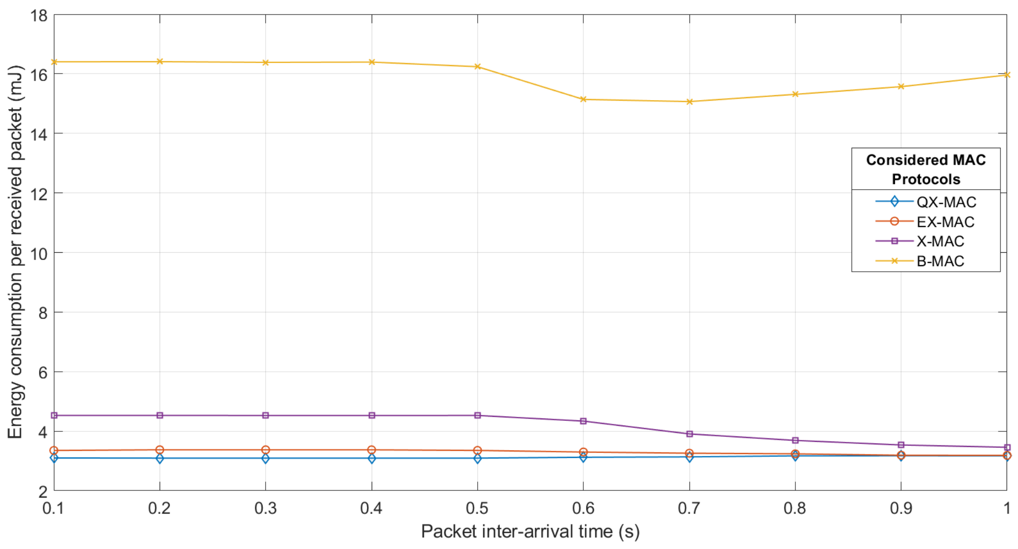


Figure 7. Energy consumption per received packet with varying traffic for the single-hop scenario.

Delay

In Figure 8, we present the delay performance comparison between QX-MAC and the other reference protocols. Here, end-to-end packet delivery delay is defined as the time difference between a packet generated at the application layer of the source node and received at the application layer of the destination node, and the mean delay is computed as below.

$$Delay = \frac{Average\ delay\ of\ all\ packets\ received\ by\ the\ sink}{Total\ number\ of\ packets\ received} \tag{12}$$

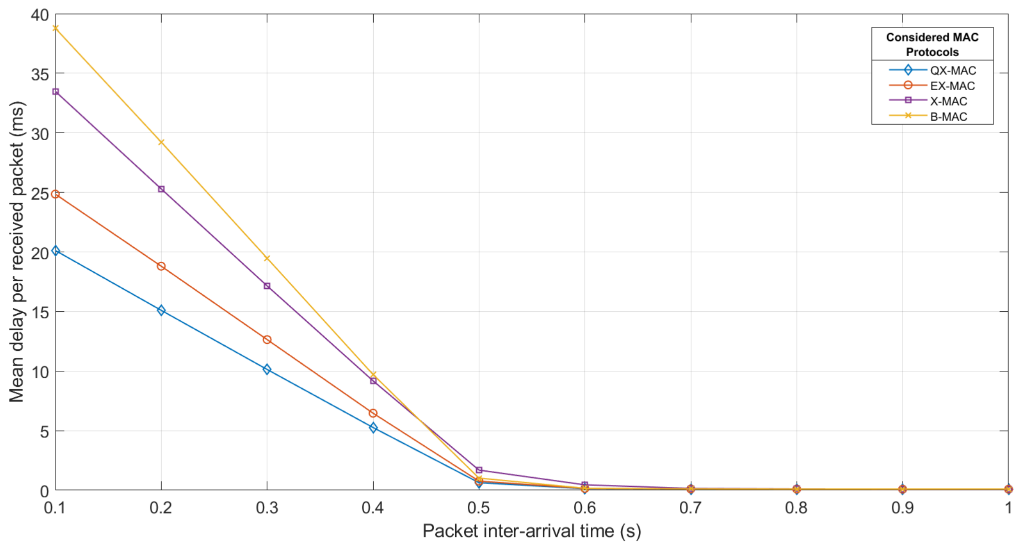


Figure 8. Mean delay per delivered packet with varying traffic for the single-hop scenario.

Analytically, there are three crucial factors that regulate the delay of a packet: (i) the waiting time of a data packet in the queue until reaching the head of the queue (queueing delay); (ii) the time interval from the time when the data packet is at the head of the queue to the time when it obtains access to the medium (contention delay); and (iii) the time interval from the time when a sender wakes up to transmit data to the time when it finds its receiver awake to receive it (in other words, the time a sender spends in preamble-listen cycle(s) until receiving PACK from the receiver).

As shown in Figure 8, compared to EX-MAC, X-MAC, and B-MAC, QX-MAC reduces the mean end-to-end packet delivery delay, even during heavy traffic. This is achieved through the inherent traffic-adaptive duty cycling approach used in QX-MAC combining the Q-learning, and the *more* bit scheme, as illustrated in Section 4.4. As explained, in the synchronized state, the QX-MAC transmitter can directly transmit all of the data packets in line to the receiver. This way the time consumed in preamble-listen cycles is saved. One may notice from the figure that while decreasing the packet arrival rate starting from 10 packets/s (i.e., inter-arrival time of 0.1 s), the mean delay to deliver a packet sharply decreases until the arrival rate becomes 2 packets/s (i.e., inter-arrival time of 0.5 s); afterward the delay is very low. This is because if the packet arrival rate decreases, the average queue length at each node decreases, which results in a lesser queueing delay. After the inflection point (0.5 s), the delay is noticeably small because from this point onward packet arrival rate is lesser than the packet send rate. After the inter-arrival time of 0.5 s, the per-packet mean delay for QX-MAC, EX-MAC, X-MAC, and B-MAC is, on average, 0.102 ms, 0.106 ms, 0.179 ms, and 0.116 ms, respectively.

Throughput

In the context of throughput metric, QX-MAC outperforms the other considered reference protocols, as depicted in Figure 9. For each packet inter-arrival time, the throughput is computed by dividing the total number of delivered packets by the total simulation time.

$$Throughput = \frac{Total\ number\ of\ delivered\ packets}{Total\ simulation\ time} \tag{13}$$

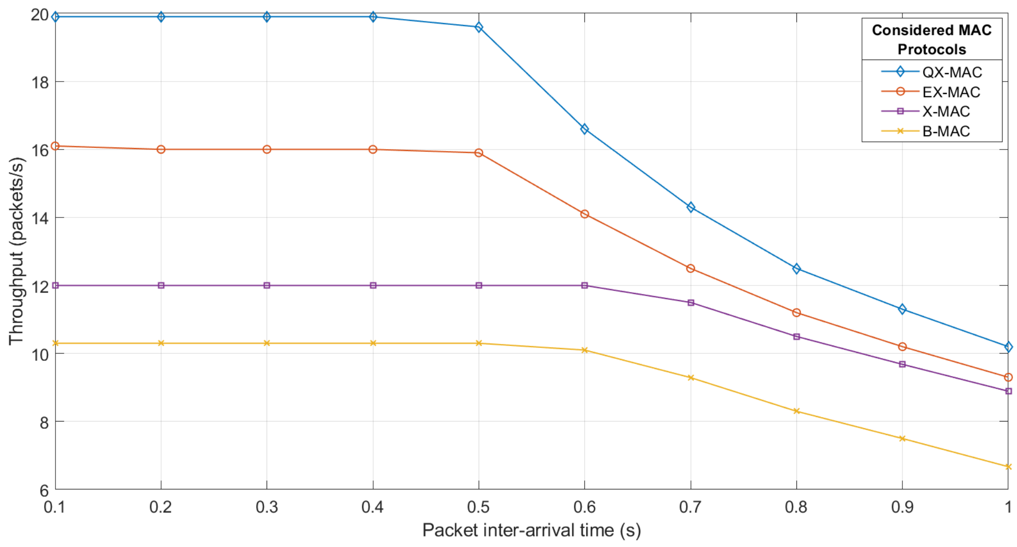


Figure 9. Throughput with varying packet inter-arrival time for the single-hop scenario.

As observed in Figure 9, for packet arrival rate of 10 packets/s, the network throughput for the QX-MAC protocol is 19.9 packets/s, while for EX-MAC, X-MAC and B-MAC, it is 16.1 packets/s, 12 packets/s, and 10.3 packets/s, respectively. It is also noticed that at the higher packet arrival rate, the achieved throughput is higher, and when the packet arrival rate falls below the send interval (0.5 s), the throughput considerably degrades. QX-MAC improves the throughput up to 24.375%, 65.833% and 93.20% compared to EX-MAC, X-MAC and B-MAC, respectively.

5.1.2. Effects of Number of Nodes

In this section, the performance analysis of the QX-MAC protocol is presented as a function of the number of nodes in the network. Here, two different data rates are chosen to investigate the performance of the QX-MAC protocol while varying the network size in the range of 5 to 25 nodes. We evaluate the performance using similar performance metrics, such as packet loss, per-packet energy consumption, mean delay, and throughput, as defined in Section 5.1.1. In the performance comparison, we exclude B-MAC in this part as it is already proven in Section 5.1.1 that X-MAC performs better than B-MAC.

In our simulations, we refer to the similar star topology as considered in Section 5.1.1. Radio model parameters (transmitter power and receiver sensitivity) are tuned such that all nodes are within the communication range of each other. We focus on a high traffic scenario where each node generates packets according to the Poisson distribution and sends them to the sink, which forwards each of the received packets to the server via a wired connection. Table 3 summarizes the simulation parameters setting for all protocols. The parameter values of λ, φ, γ and the maximum number of retires that are only applicable for QX-MAC remain the same as given in Table 2.

Table 3. Parameters used for scalability analysis.

Parameter	Value
Header length	2 bytes
Message length	10 bytes
Bit rate	19.2 kbps and 38.4 kbps
Queue capacity	30 packets
Packet generation interval	*exponential* (0.1 s)
Send interval	0.5 s
Slot duration	According to Table 4
Number of nodes	5, 10, 15, 20, 25 nodes
Transmit power	2.24 mW
Receiver sensitivity	−100 dBm
Simulation time	1000 s

Table 4. Optimum slot duration at which packet loss is minimum.

MAC Protocol	Bit Rate (kbps)	Slot Duration (s)	Number of Sensor Nodes				
			5	10	15	20	25
X-MAC	19.2	sink	0.035 s	0.045 s	0.038 s	0.038 s	0.038 s
		nodes	0.194 s	0.25 s	0.25 s	0.25 s	0.25 s
	38.4	sink	0.04 s	0.02 s	0.02 s	0.02 s	0.02 s
		nodes	0.222 s	0.111 s	0.111 s	0.111 s	0.111 s
EX-MAC	19.2	sink	0.035 s	0.045 s	0.045 s	0.045 s	0.038 s
		nodes	0.194 s	0.25 s	0.25 s	0.25 s	0.25 s
	38.4	sink	0.045 s	0.045 s	0.038 s	0.045 s	0.038 s
		nodes	0.25 s	0.25 s	0.25 s	0.25 s	0.25 s
QX-MAC	19.2	sink	0.045 s	0.045 s	0.038 s	0.038 s	0.038 s
		nodes	0.25 s	0.25 s	0.25 s	0.25 s	0.25 s
	38.4	sink	0.085 s	0.045 s	0.038 s	0.045 s	0.038 s
		nodes	0.472 s	0.25 s	0.25 s	0.25 s	0.25 s

To quantify QX-MAC scalability performance compared to EX-MAC and X-MAC, we first investigate the appropriate slot duration setting for each protocol while leaving the number of nodes constant. For a certain network size, for each protocol, we run the simulations for sink slot duration ranging from 0.01 s to 1 s with 0.0005 s increment to obtain the optimum slot duration at which packet loss is minimum. We repeat the same process for each considered number of nodes (i.e for network size of 5, 10, 15, 20, and 25 nodes). Later, at each protocol's respective optimum slot duration, we examine network performance while increasing the number of nodes. For each considered data rate, we follow the same process to generate the simulation results. The optimum slot durations for each protocol for each considered network size and data rate are tabulated in Table 4.

Packet Loss

Figure 10 compares the packet loss of QX-MAC, EX-MAC, and X-MAC while increasing the number of nodes. It is obvious from the graph that regardless of the bit rate, the packet loss increases with the increase of the number of sensor nodes. As the network size grows, more the contention in the medium, hence lower the success probability of packet transmission, and more the packet dropouts due to queue overflow and collisions.

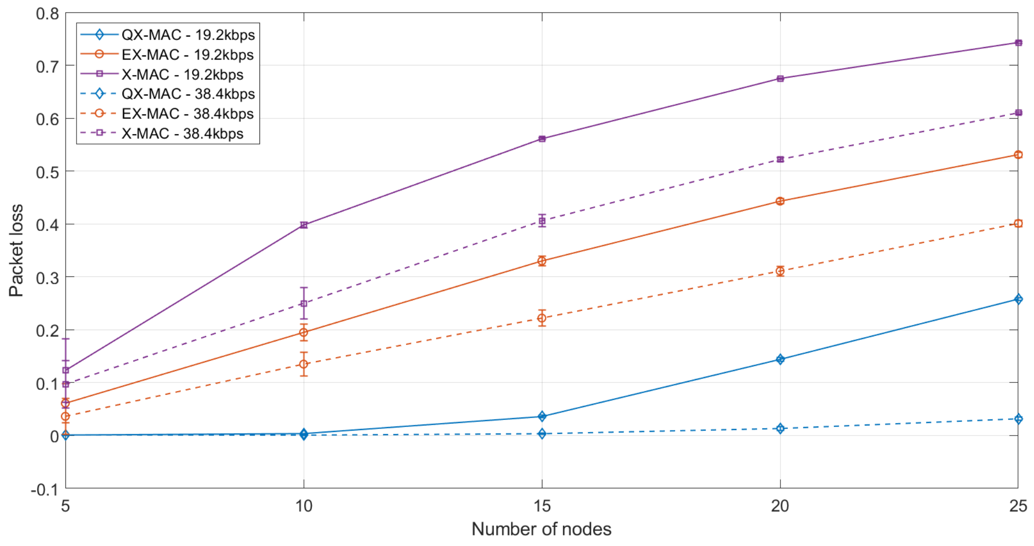


Figure 10. Effect of the number of nodes on the packet loss. For details, see Equation (10).

As seen in the figure, compared to the other two protocols, QX-MAC shows the minimum packet loss in all cases. At 19.2 kbps bit rate, with QX-MAC, the network experiences a very low packet loss (0.1%) when five nodes are considered. After that, the packet loss gradually increases when the network size grows. At 19.2 kbps, for 25 nodes, QX-MAC shows 51.41% and 65.27% lower packet loss than EX-MAC and X-MAC, respectively. At 38.4 kbps bit rate, the QX-MAC protocol provides a packet loss in the range of 0.12–3.17% while increasing the number of nodes. We observe that for all three protocols, at higher bit rate, the packet loss is lower. This is because if the bit rate is increased, packet transfer time (i.e., time to transmit a packet) decreases, thereby channel occupancy time becomes smaller, which in turn results in smaller packet collision probability, i.e., lower packet loss.

Figure 10 also shows that X-MAC shows the maximum packet loss in this scenario. This is because with more nodes in the network, (i) for each sender, there is lower probability to access the medium in a cycle (i.e., higher probability that a sender has longer queue) and (ii) a X-MAC sender can transmit only one data packet if it wins the medium. Hence, with more nodes, there is a higher probability that the queue at each X-MAC node overflows, which in turn causes considerable amount of packet dropouts.

Energy Consumption

The per-packet energy consumption graphs in Figure 11 depict that for all three protocols, the energy consumption per received packet increases with increasing nodes. When more nodes are in the network, more senders transmit packets, i.e., more energy is consumed in data transmission and reception; consequently, the network's total energy consumption increases. Once again, it is observed from the graph that at 19.2 kbps bit rate, the QX-MAC protocol offers the best energy efficiency for all cases. At 19.2 kbps, on average, QX-MAC provides an energy savings of 10.54% and 34.97% over EX-MAC and X-MAC, respectively. At 38.4 kbps, QX-MAC outperforms X-MAC by 30.20% (on average).

In this scenario, for 38.4 kbps, QX-MAC provides more energy savings than EX-MAC until 15 nodes, after which QX-MAC's per-packet energy consumption is slightly higher than that of EX-MAC. Since QX-MAC implements a combination of DACKs and retransmissions of missing packets, it provides reliable communication at the cost of increased overhead, which has an impact on the overall energy consumption of the network at a higher bit rate with a bigger network size.

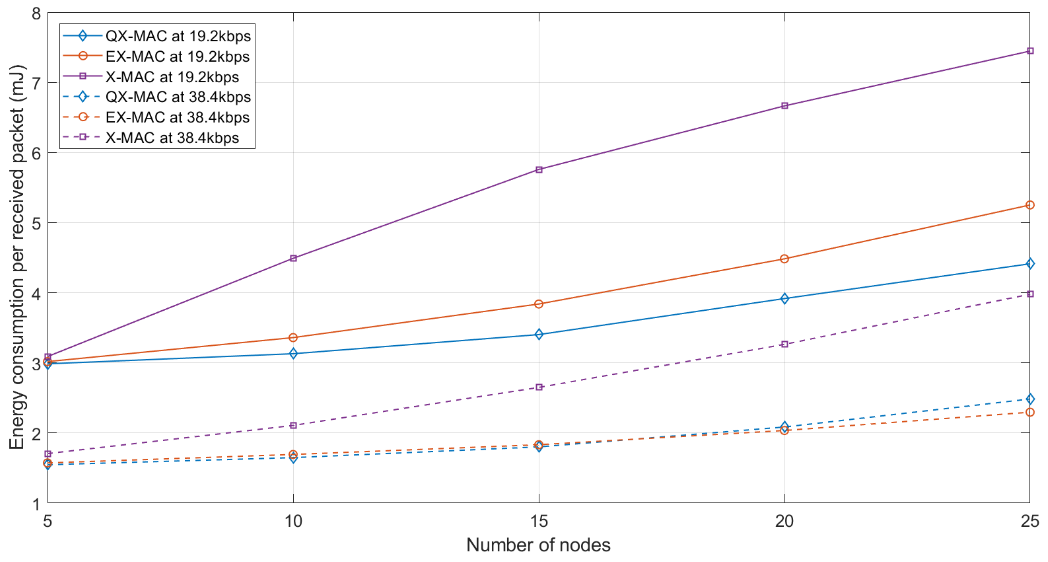


Figure 11. Effect of the number of nodes on the energy consumption per received packet.

Delay

In the context of the delay metric, regardless of the data rate and the number of nodes, QX-MAC experiences a lower mean delay per received packet compared to the other reference protocols, as seen in Figure 12. Figure 12 also shows that the average delay of a packet decreases while increasing the number of nodes. With a lesser number of nodes, the queue length at each node as well as the contention delay become smaller; however, after waking up to send a data packet, the sender needs to wait until the sink is awake. With five sending nodes in the network, there is a higher probability that each node transmits a data packet in a cycle. As the number of nodes increases, both the queueing delay and contention delay increase since a sender needs to contend with more nodes to access the medium; and this leads to the higher probability that a node will have a longer queue. However, once a sending node obtains access to the medium and receives PACK from the sink, with QX-MAC, it is able to transmit all of its queued packets straightway until its queue is empty, thereby saving time in data transmission. Consequently, the average delay of all packets received by the sink increases very slightly with an increasing number of nodes. For QX-MAC, the average delay of all packets received by the sink lies in the range of [399.5–401.6] s for a network size of [5–25] nodes. Since with QX-MAC, the number of received packets significantly increases [see Figure A1] with more nodes in the network, the mean delay per received packet eventually shows a decreasing trend with an increasing number of nodes. X-MAC experiences the highest delay because it only operates in the unsynchronized state.

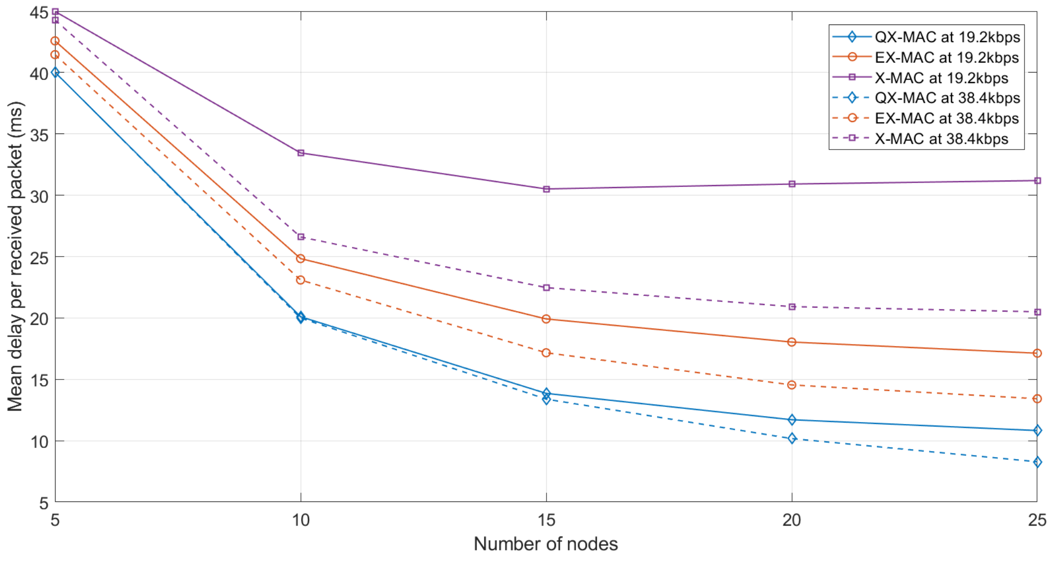


Figure 12. Effect of the number of nodes on the mean end-to-end packet delivery delay.

Throughput

Figure 13 presents the throughput achieved by the three protocols with a different number of nodes in the network. We observe that apart from X-MAC at 19.2 kbps, each protocol's throughput metric shows an increasing trend as the number of nodes increases. This is consistent with the delay metric shown in Figure 12 because when the delay decreases, the throughput should increase. In the considered scenario, when the network size grows, more packets are received by the server within the considered simulation time, which eventually increases the throughput. It is also seen from the figure that at 38.4 kbps, the throughput for each considered network size is higher than that of 19.2 kbps. At a higher rate, the time to transmit a packet decreases, so more packets can be delivered to the destination for the same simulation time. At both bit rates, X-MAC results in a very low throughput since it suffers from the drawback of higher packet loss with a larger number of nodes (see Figure 10). Irrespective of bit rates, QX-MAC clearly shows the highest throughput even when the number of nodes increases. Compared to EX-MAC and X-MAC, for the network size of 25 and at 19.2 kbps bit rate, QX-MAC improves the throughput by the factors of 1.59 and 2.88, respectively; and at 38.4 kbps, by the factors of 1.62 and 2.48, respectively.

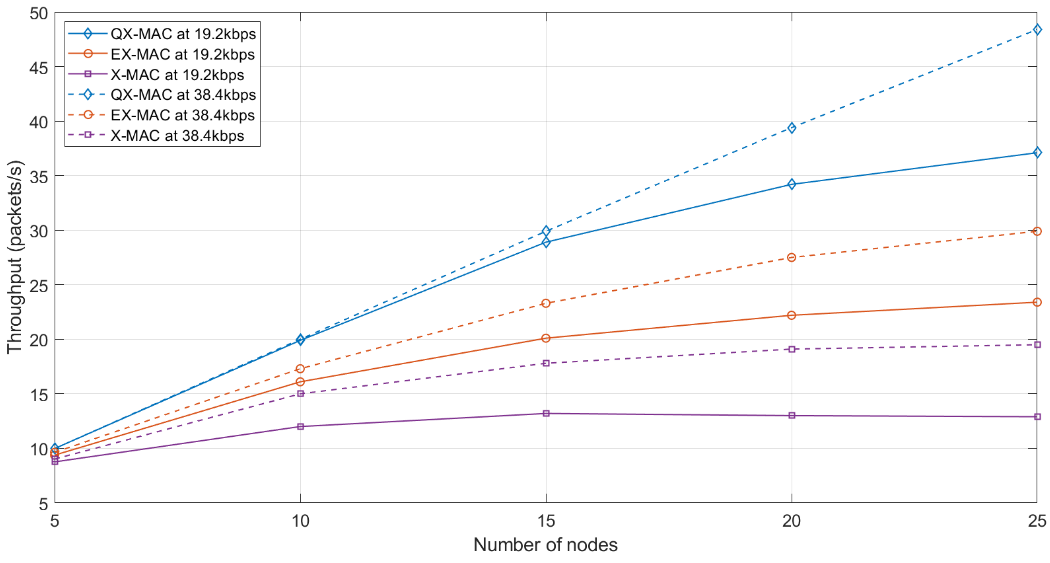


Figure 13. Effect of the number of nodes on throughput.

5.2. Multi-Hop Scenario

In this section, we consider a multi-hop scenario with linear topology to carry out the performance comparison of the QX-MAC, EX-MAC, and X-MAC protocols. Although in general, both MAC and routing protocols play a significant role in multi-hop networks, in case of linear wireless sensor networks, the routing protocols can be simplified. In this work, we, therefore, consider static routing to focus on the MAC layer. The simulation scenario (as depicted in Figure 14) is: in a sensor field, five nodes are regularly placed in a linear form with the first node (*hostA)* in the line generating data packets destined for the sink or GW (*hostB*) located at the end of the line. The source node periodically generates UDP packets and transmits them to the sink following a predefined route via three intermediate nodes (*hostR1*, *hostR2*, and *hostR3*), which basically forward the data packets to the next hop toward the destination, thereby supporting hop-by-hop data transmission. The transmitter power is set such that each node is within the communication range of its 1-hop neighbors. Each message is of 10 byte length and the interval between messages is defined by a Poisson process. In this simulation, for all MAC protocols, we set the sink slot duration 2 times shorter than that of the source and intermediate nodes.

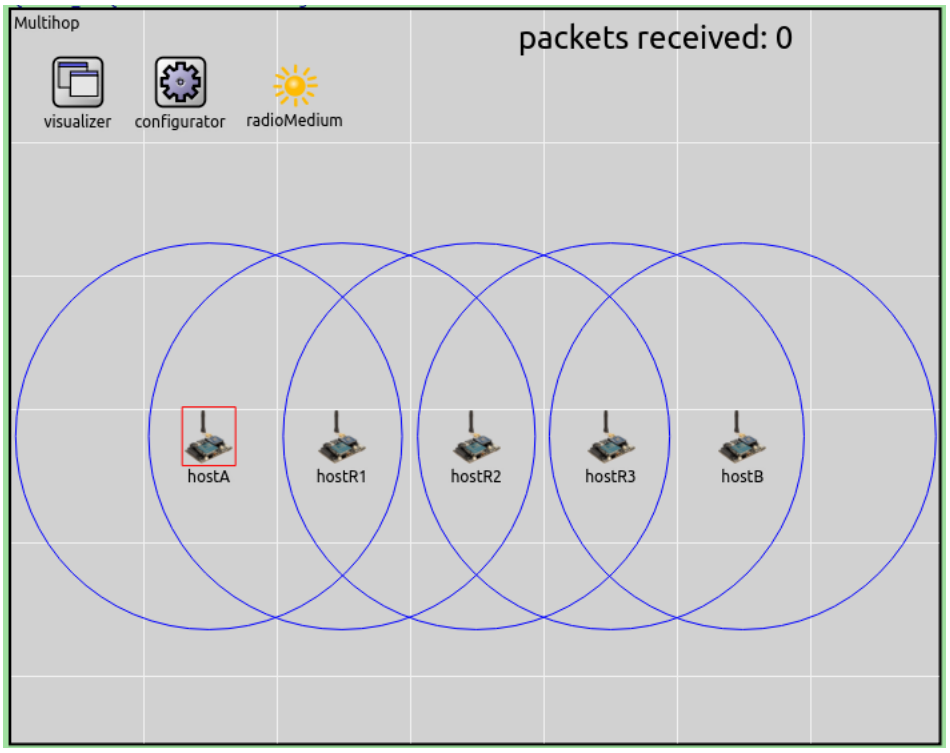


Figure 14. Multi-hop scenario in OMNeT++.

To generate simulation results, for each protocol, we perform five simulation runs for each parameter combination. The results are then averaged for statistical accuracy. The crucial simulation parameters for QX-MAC, EX-MAC, and X-MAC are summarized in Table 5.

5.3. QoS and Energy Performance

Here, we quantify the performance of the QX-MAC protocol compared to the EX-MAC and X-MAC protocols while increasing the packet arrival rate in the range of 1 to 25 packets/s. We exclude B-MAC in this part since B-MAC originally is not designed for multi-hop networks. The comparison is performed in terms of different performance metrics, such as packet delivery ratio (PDR), energy consumption, and delay.

Table 5. Simulation Parameters.

Approach	Parameter	Value
Common	Header length	2 bytes
	Message length	10 bytes
	Preamble duration	0.83 ms
	Bit rate	250 kbps
	Queue capacity	30 packets
	Sink slot duration	0.1 s
	Source/relay node's slot duration	0.2 s
	Transmit power	0.7 mW
	Receiver sensitivity	−85 dBm
	Simulation time	1000 s
QX-MAC	Learning rate (λ)	0.5
	Discount factor (φ)	0.618
	Decay rate (γ)	0.00001
	Maximum transmission attempts	2 retries

5.3.1. Packet Delivery Ratio (PDR)

In order to evaluate the effect of the packet arrival rate on the PDR, we record the total number of packets sent and received for each considered packet inter-arrival time, and define the PDR as the ratio of total number of packets received over the total number of packets sent.

As shown in Figure 15, at a lighter traffic loads (at packet inter-arrival time of 1 s), the PDR for QX-MAC, EX-MAC, and X-MAC are 100%, 99.8%, and 99.7%, respectively. As the traffic loads become heavier, (i) X-MAC's PDR results drastically fall; (ii) EX-MAC still outperforms X-MAC and retains an acceptable PDR level until packet arrival rate reaches 10 packets/s after which it gradually decreases; (iii) QX-MAC outperforms the other two protocols and maintains high PDR (around 100%) in all cases. It is observed from the figure that when the traffic loads increase, QX-MAC's PDR performance only slightly degrades from 100% (at a packet arrival rate of 1 packet/s) to 97.2% (at a packet arrival rate of 25 packets/s), while for EX-MAC and X-MAC, network's PDR declines to 43.4% and 11.2%, respectively. Therefore, it is apparent that in terms of PDR, X-MAC does not work well in multi-hop WSNs under high traffic conditions as it only operates in an unsynchronized state. The traffic-adaptive mechanism implemented in the QX-MAC protocol enables it to perform well under varying traffic loads.

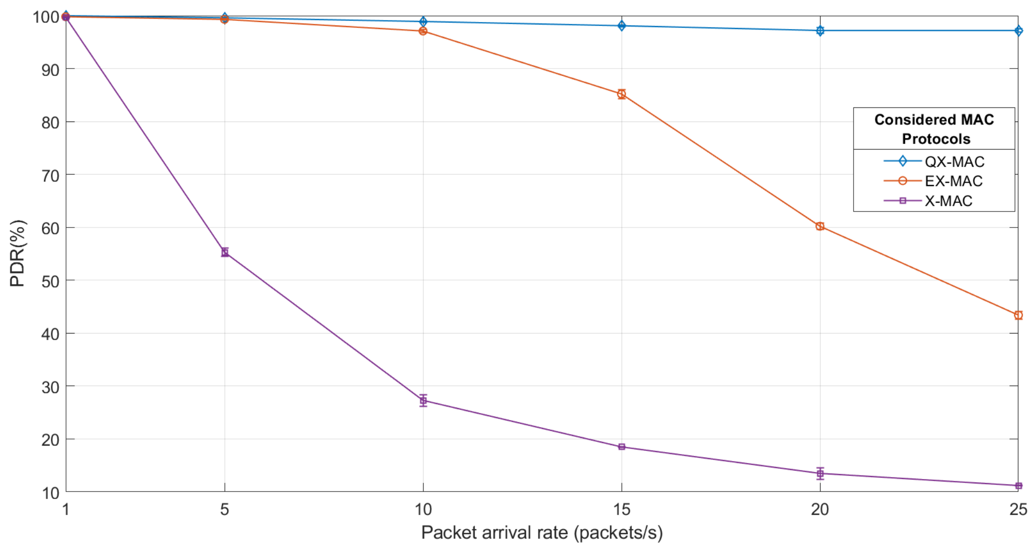


Figure 15. Effect of the packet arrival rate on the network's PDR for the multi-hop scenario.

5.3.2. Energy Consumption

To assess the energy performance of QX-MAC in this multi-hop scenario, we measure the energy consumption of the source node, intermediate nodes, and the sink and compute the energy consumption per received packet the same way as in Equation (11). As seen in Figure 16, for low traffic loads when the packet arrival rate is 1 packet/s, the overall per-packet energy consumption of QX-MAC is slightly higher than EX-MAC. This is due to the fact that a QX-MAC source or forwarding node keeps its radio on until receiving DACK from the next hop node to ensure data reliability. This DACK exchange occurs in each hop toward the destination, thereby adding up the overall energy consumption due to overhead (see Table 6). Since the total number of received packets by the sink at this low traffic rate is similar for QX-MAC and EX-MAC, the per-packet energy consumption for QX-MAC is a bit (approximately 1.41%) higher than that of EX-MAC.

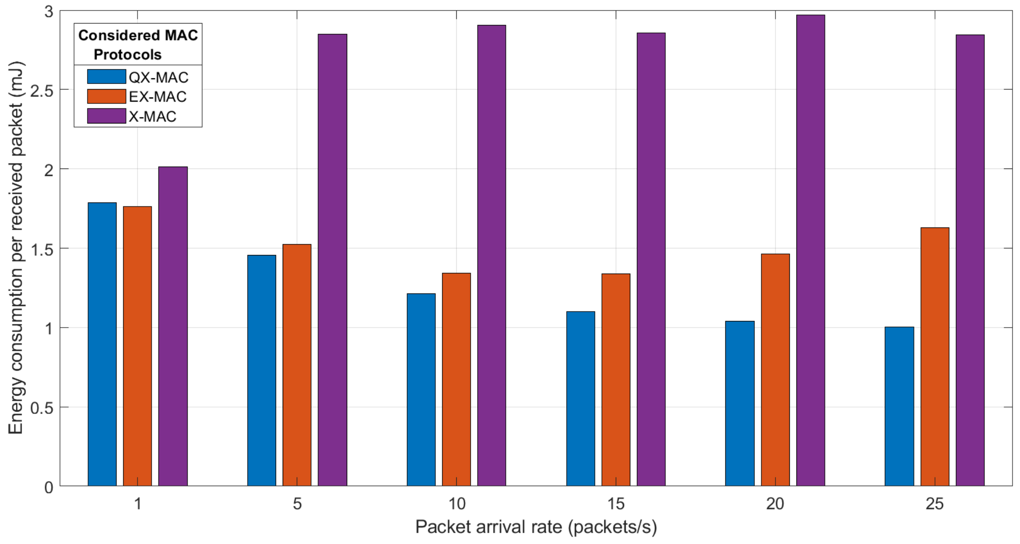


Figure 16. Effect of the packet arrival rate on the per-packet energy consumption for the multi-hop scenario.

Table 6. Network average energy consumption.

MAC Protocol	**Packet Arrival Rate (Packets/s)**					
	1	**5**	**10**	**15**	**20**	**25**
X-MAC	2.01 J	7.94 J	7.96 J	7.94 J	7.98 J	7.94 J
EX-MAC	1.75 J	7.61 J	13.19 J	17.02 J	17.73 J	17.77 J
QX-MAC	1.77 J	7.22 J	12.08 J	16.28 J	20.20 J	24.33 J

Figure 16 also demonstrates that for higher traffic loads ranging from 5 packets/s to 25 packets/s, among the three protocols, with QX-MAC, the network consumes the least amount of energy to deliver a packet to the sink. When the traffic gets heavier, although the total energy consumption (see Table 6) gets higher, QX-MAC consistently shows better energy efficiency (i.e., lower energy consumption per received packet) compared to EX-MAC and X-MAC because for all traffic rates, QX-MAC maintains significantly higher PDR (see Figure 15). Furthermore, it can be seen from the figure that as the traffic load increases, the per-packet energy consumption with the QX-MAC protocol decreases. The reason for this is that while the traffic load becomes heavier, more energy is consumed in transmitting and receiving rather than in idle listening. Regarding X-MAC, one may observe that at a higher packet arrival rate, X-MAC consumes similar energy, which is expected because with the increment of traffic loads, no improvement is seen in X-MAC in terms of the total number of received packets by the sink.

5.3.3. Delay

The mean delay of all packets received by the sink is shown in Figure 17. In this scenario, the end-to-end packet delivery delay is defined as the time difference between a packet is originated at the source and received at the destination. In Figure 17, we observe that compared to EX-MAC and X-MAC, the QX-MAC protocol experiences significantly lower mean delay irrespective of traffic loads. At the packet arrival rate of 1 packet/s, QX-MAC reduces the mean delay by 8.14% and 17.06% compared to EX-MAC and X-MAC, respectively. At higher traffic, QX-MAC results in a very slight increase in mean delay; and at the arrival rate of 25 packets/s, the mean delay experienced by QX-MAC is 83.43% and 94.37% lower than that of EX-MAC and X-MAC, respectively. For EX-MAC, as the traffic rate increases beyond 5 packets/s, the mean delay gradually rises, but still maintains the lower level than the X-MAC protocol. EX-MAC performs better than X-MAC in all cases because it employs the *more* bit scheme to deliver all the packets in the queue. However, EX-MAC cannot achieve the same level of delay performance as QX-MAC because in the synchronized state, whenever an EX-MAC node has more packets to transmit, it still spends extra time for at least one preamble-listen cycle. On the other hand, QX-MAC implements a combined approach of the *more* bit scheme and the *Q-learning* algorithm by which under high traffic loads, a QX-MAC sender in the synchronized state can directly transfer all of the data packets in line, thereby reducing delay. Regarding X-MAC, it is clearly visible from the figure that at higher traffic loads, X-MAC shows quite a poor delay performance.

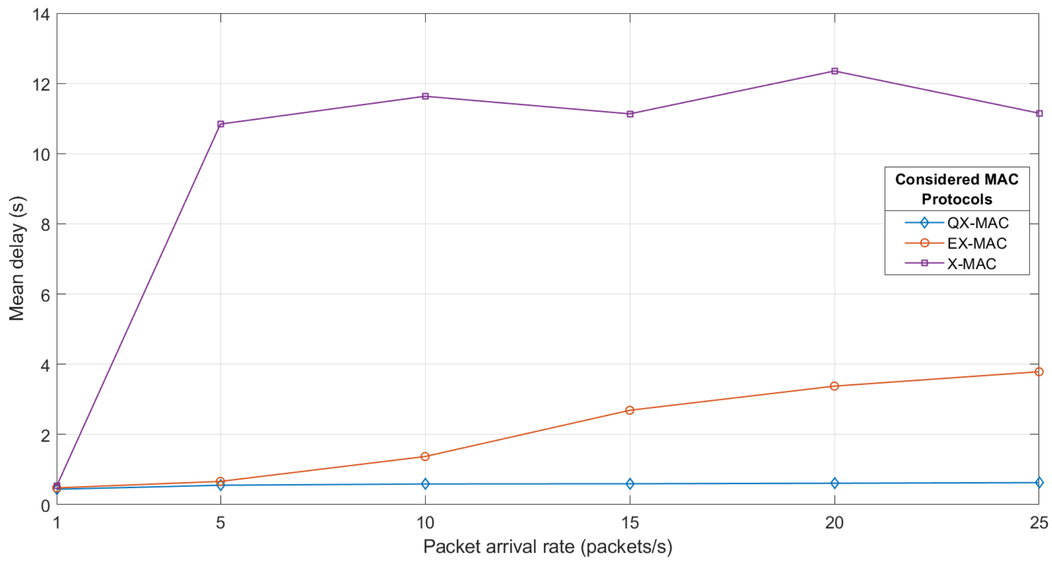


Figure 17. Effect of the packet arrival rate on the mean delay of all packets received by the sink for the multi-hop scenario.

6. Conclusions

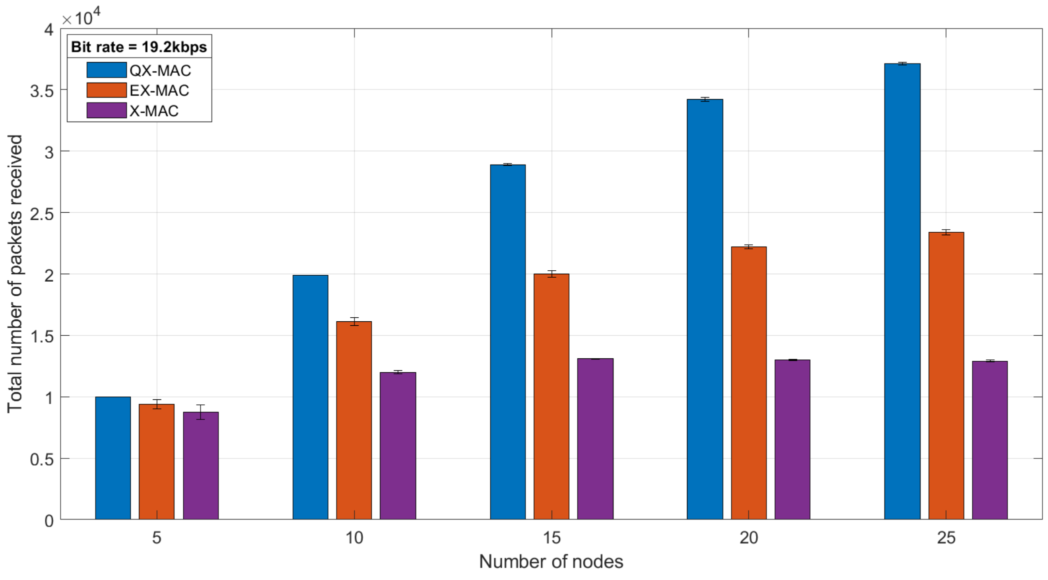
Although X-MAC is one of the most widely accepted approaches for IoT networks, it achieves high energy efficiency and low latency only under low traffic scenarios. In this paper, a contention-based, traffic-adaptive WSN MAC scheme, namely QX-MAC, is presented, that primarily aims to improve QoS without compromising the energy gains in dynamic traffic scenarios. QX-MAC addresses the foreseen limitations and omissions in basic X-MAC design. In short, QX-MAC exploits the Q-learning algorithm and *more* bit scheme to tune the active period and duty cycle of the sender and the target receiver based on the sender's queue status; implements the proposed solutions to overcome the identified design flaws in X-MAC; as well as incorporates the data acknowledgment and retransmission mechanism to ensure reliability in data transmission. We validate the effectiveness of our proposed approach by means of numerous OMNeT++ simulations considering single-hop and multi-hop communications. For a single-hop scenario, we evaluate the performance of QX-MAC while varying packet arrival rate to validate its efficacy under dynamic traffic loads. We also investigate QX-MAC performance while changing network size (i.e., the number of nodes in the network) under high traffic with different data rates. Finally, to further examine the proposed model in a multi-hop scenario, we carry out simulations considering a linear topology with different packet arrival rates. Throughout the evaluations, it is clearly observed that compared with other reference protocols, on average, QX-MAC offers lower energy consumption, higher PDR, lower delay, or higher network throughput. Evaluating the performance of our proposed model in real testbeds could be a future work direction.

Author Contributions: Conceptualization, F.A.; methodology, F.A.; validation, F.A.; formal analysis, F.A.; investigation, F.A.; writing—original draft preparation, F.A.; writing—review and editing, F.A.; visualization, F.A.; supervision, R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

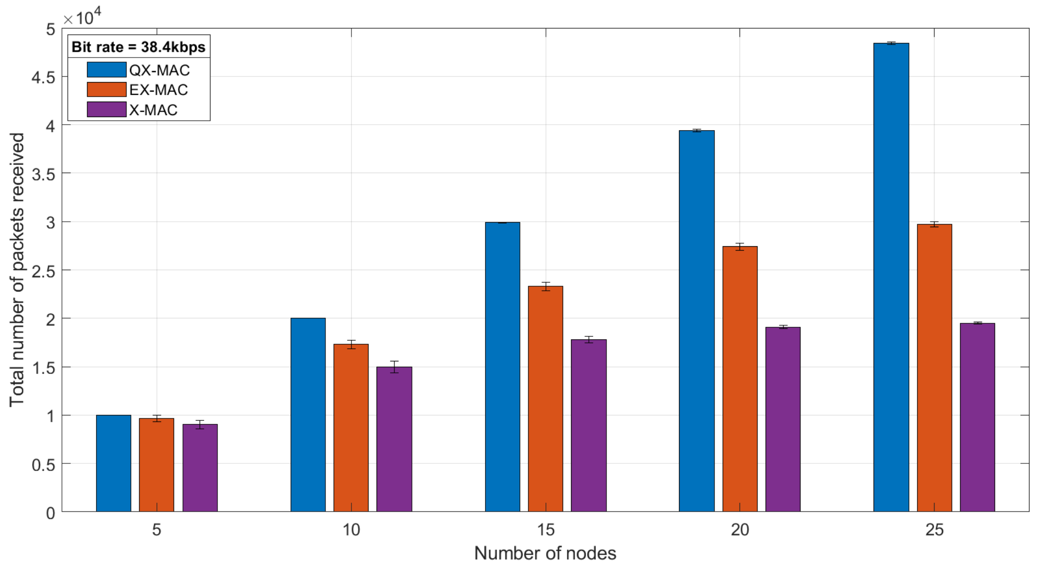
Conflicts of Interest: The authors declare no conflict of interest.

Appendix A



(**a**)

Figure A1. *Cont.*



(**b**)

Figure A1. Effects of number of nodes on the total number of packets received by the server. (**a**) At bit rate = 19.2 kbps; (**b**) At bit rate = 38.4 kbps.

References

1. Internet of Things: Wireless Sensor Networks. Available online: https://www.iec.ch/basecamp/internet-things-wireless-sensor-networks (accessed on 20 December 2021).
2. Matin, M.A.; Islam, M.M. Overview of wireless sensor network. In *Wireless Sensor Networks-Technology and Protocols*; Matin, M.A., Ed.; IntechOpen: Rijeka, Croatia, 2012.
3. Guo, F.; Yu, F.R.; Zhang, H.; Li, X.; Ji, H.; Leung, V.C. Enabling massive IoT toward 6G: A comprehensive survey. *IEEE Internet Things J.* **2021**, *8*, 11891–11915. [CrossRef]
4. Kumar, A.; Zhao, M.; Wong, K.J.; Guan, Y.L.; Chong, P.H.J. A comprehensive study of IoT and WSN MAC protocols: Research issues, challenges and opportunities. *IEEE Access* **2018**, *6*, 76228–76262. [CrossRef]
5. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
6. Beaudaux, J.; Gallais, A.; Montavont, J.; Noel, T.; Roth, D.; Valentin, E. Thorough empirical analysis of X-MAC over a large scale internet of things testbed. *IEEE Sens. J.* **2013**, *14*, 383–392. [CrossRef]
7. Oller, J.; Demirkol, I.; Casademont, J.; Paradells, J.; Gamm, G.U.; Reindl, L. Has Time Come to Switch from Duty-Cycled MAC Protocols to Wake-Up Radio for Wireless Sensor Networks? *IEEE/ACM Trans. Netw.* **2016**, *24*, 674–687. [CrossRef]
8. Ye, W.; Heidemann, J.; Estrin, D. An energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA, 23–27 June 2002; Volume 3, pp.1567–1576.
9. Van Dam, T.; Langendoen, K. An adaptive energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003; pp. 171–180.
10. Lu, G.; Krishnamachari, B.; Raghavendra, C.S. An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In Proceedings of the 18th International Parallel and Distributed Processing Symposium, Santa Fe, NM, USA, 26–30 April 2004; pp. 224–231.
11. Polastre, J.; Hill, J.; Culler, D. Versatile low power media access for wireless sensor networks. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 95–107.
12. El-Hoiydi, A.; Decotignie, J.D. Low power downlink MAC protocols for infrastructure wireless sensor networks. *Mob. Netw. Appl.* **2005**, *10*, 675–690. [CrossRef]
13. Buettner, M.; Yee, G.V.; Anderson, E.; Han, R. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, Boulder, CO, USA, 31 October–3 November 2006; pp. 307–320.
14. Afroz, F.; Braun, R.; Chaczko, Z. XX-MAC and EX-MAC: Two Variants of X-MAC Protocol for Low Power Wireless Sensor Networks. *Adhoc Sens. Wirel. Netw.* **2022**, *51*, 285–314.
15. Afroz, F.; Braun, R. QX-MAC: Improving QoS and Energy Performance of IoT-based WSNs using Q-Learning. In Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 4–7 October 2021; pp. 455–462.
16. Anwander, M.; Wagenknecht, G.; Braun, T.; Dolfus, K. Beam: A burst-aware energy-efficient adaptive mac protocol for wireless sensor networks. In Proceedings of the 2010 Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany, 15–18 June 2010; pp. 195–202.

17. Yigitel, M.A.; Incel, O.D.; Ersoy, C. QoS-aware MAC protocols for wireless sensor networks: A survey. *Comput. Netw.* **2011**, *55*, 1982–2004. [CrossRef]
18. Di Marco, P.; Park, P.; Fischione, C.; Johansson, K.H. TREnD: A timely, reliable, energy-efficient and dynamic wsn protocol for control applications. In Proceedings of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; pp. 1–6.
19. Djiroun, F.Z.; Djenouri, D. MAC protocols with wake-up radio for wireless sensor networks: A review. *IEEE Commun. Surv. Tutor.* **2016**, *19*, 587–618. [CrossRef]
20. Buratti, C.; Martalò, M.; Verdone, R.; Ferrari, G. *Sensor Networks with IEEE 802.15. 4 Systems: Distributed Processing, MAC, and Connectivity*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
21. Afroz, F.; Braun, R. Energy-efficient MAC protocols for wireless sensor networks: A survey. *Int. J. Sens. Netw.* **2020**, *32*, 150–173. [CrossRef]
22. Souil, M. Contribution to Quality of Service in Wireless Sensor Networks. Ph.D. Thesis, Université de Technologie de Compiègne, Compiègne, France, 2013.
23. Quintero, V.; Estevez, C.; Orchard, M.; Pérez, A. Improvements of Energy-Efficient Techniques in WSNs: A MAC-Protocol Approach. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1188–1208. [CrossRef]
24. Ketshabetswe, L.K.; Zungeru, A.M.; Mangwala, M.; Chuma, J.M.; Sigweni, B. Communication protocols for wireless sensor networks: A survey and comparison. *Heliyon* **2019**, *5*, e01591. [CrossRef] [PubMed]
25. Sadeq, A.S.; Hassan, R.; Sallehudin, H.; Aman, A.H.M.; Ibrahim, A.H. Conceptual Framework for Future WSN-MAC Protocol to Achieve Energy Consumption Enhancement. *Sensors* **2022**, *22*, 2129. [CrossRef] [PubMed]
26. Du, S.; Saha, A.K.; Johnson, D.B. RMAC: A routing-enhanced duty-cycle MAC protocol for wireless sensor networks. In Proceedings of the IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, Anchorage, AK, USA, 6–12 May 2007; pp. 1478–1486.
27. Liu, C.J.; Huang, P.; Xiao, L. TAS-MAC: A traffic-adaptive synchronous MAC protocol for wireless sensor networks. *ACM Trans. Sens. Netw. (TOSN)* **2016**, *12*, 1–30. [CrossRef]
28. Sun, Y.; Gurewitz, O.; Johnson, D.B. RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, Raleigh, NC, USA, 5–7 November 2008; pp. 1–14.
29. Morshed, S.; Heijenk, G. TR-MAC: An energy-efficient MAC protocol exploiting transmitted reference modulation for wireless sensor networks. In Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Montreal, QC, Canada, 21–26 September 2014; pp. 21–29.
30. Henna, S. SA-RI-MAC: Sender-assisted receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *International Conference on Mobile Lightweight Wireless Systems*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 81, LNICST, pp. 120–135.
31. Tang, L.; Sun, Y.; Gurewitz, O.; Johnson, D.B. PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In Proceedings of the 2011 IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 1305–1313.
32. Liu, Z.; Elhanany, I. RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks. In Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, USA, 23–25 April 2006; pp. 768–773.
33. Kosunalp, S.; Chu, Y.; Mitchell, P.D.; Grace, D.; Clarke, T. Use of Q-learning approaches for practical medium access control in wireless sensor networks. *Eng. Appl. Artif. Intell.* **2016**, *55*, 146–154. [CrossRef]
34. Jang, B.; Lim, J.B.; Sichitiu, M.L. An asynchronous scheduled MAC protocol for wireless sensor networks. *Comput. Netw.* **2013**, *57*, 85–98. [CrossRef]
35. Aby, A.T.; Guitton, A.; Lafourcade, P.; Misson, M. SLACK-MAC: Adaptive MAC protocol for low duty-cycle wireless sensor networks. In *International Conference on Ad Hoc Networks*; Springer: Cham, Switzerlnad, 2015; pp. 69–81.
36. INET Framework. Available online: https://inet.omnetpp.org/ (accessed on 21 December 2021).
37. Sutton, R.S. Introduction: The challenge of reinforcement learning. In *Reinforcement Learning*; Springer: Boston, MA, USA, 1992; pp. 1–3.
38. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
39. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
40. Watkins, C.J.C.H. Learning from Delayed Rewards. Ph.D. Thesis, King's College, Cambridge, UK, 1989.
41. Lim, S.; Yu, H.; Lee, H. Optimal Tethered-UAV Deployment in A2G Communication Networks: Multi-Agent Q-Learning Approach. *IEEE Int. Things J.* **2022**. [CrossRef]
42. Mathew, A.; Roy, A.; Mathew, J. Intelligent residential energy management system using deep reinforcement learning. *IEEE Syst. J.* **2020**, *14*, 5362–5372. [CrossRef]
43. OMNeT++. Available online: https://omnetpp.org/ (accessed on 21 December 2021).