International Journal of
Network Dynamics and Intelligence (IJNDI)

Scilight

Article

Check for updates

A Q-Learning-Based Energy-Efficient Multi-Flow Two-Way MAC Protocol for Un-Derwater Acoustic Sensor Networks

Sajid Ullah Khan

Department of Information Systems, College of Computer Engineering and Sciences,
Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; sk.khan@psau.edu.sa
Received: 25 July 2026; Revised: 21 September 2026; Accepted: 30 October 2026; Published: 9 November 2026

Abstract: The Underwater Acoustic Wireless Sensor Network (UAWSN) is a remote operating system that uses sensors Rx nodes to monitor natural or physical parameters for underwater scenarios. These sensors' Rx nodes are autonomous devices and critical parts of UAWSNs. Due to the increased bit error rate and high propagation delay for the basic parameters of the physical medium in UAWSNs, conventional MAC protocols are insufficient. This research presents a Q-learning-based Energy Efficient Multi-Flow Two-Way MAC Protocol (Q-EEMFTW-MAC) for UAWSNs to address multi-flow and multi-hop data packet flows in a cooperative network environment. Our proposed Q-EEMFTW-MAC protocol allows multiple data transmission flows to operate simultaneously while being modified by variable traffic conditions. In contrast to existing underwater MAC protocols, our proposed MAC protocol provides a framework for multi-hop flows by analyzing every incoming communicating data packet in directed layer processing. During the communicating data packet flow setup, all the Tx anchor nodes within the cooperative regions should receive an acknowledgement from the neighbor nodes. The performance analysis of our proposed Q-EEMFTW-MAC is carried out through simulations by adopting/considering various variables such as network throughput, end-to-end latency energy consumption and bit error rate probability. Furthermore, our proposed protocol is compared with the existing cascading multi-flow reservation and transmission MAC protocol (CMRT) and a Q-Learning-based Adaptive MAC Protocol for UAWSNs. The experimental results showed that Q-EEMFTW-MAC can transmit up to 84.3% more data and use up to 88.6% less energy than comparable MAC protocols under the evaluated conditions. For end-to-end latency, the proposed protocol reduces packet transmission delay by 27.3%, while the Q-EEMFTW-R/N achieves a 36.3% reduction.

Keywords: underwater acoustic wireless sensor networks; energy consumption; end-to-end latency; multi-flow two-way mac protocol; Q-learning

1. Introduction

Underwater Acoustic Wireless Sensor Networks (UAWSNs) are used for applications such as environmental monitoring, disaster management, and military surveillance [1,2]. In a multi-hop UAWSN, a source node may transmit its data to the destination through one or more intermediate relay nodes when direct communication is unavailable, inefficient, or energy intensive [3,4].

In UAWSNs, MAC protocols are crucial for the success of the network as a whole. Because the primary goal of MAC protocols in UAWSNs are to prevent two connected nodes from simultaneously delivering data packets, collision of data packets can be prevented by structuring the controlling communication protocol in such a way that if two connecting nodes are somehow prevented from doing so, collision of data packets is avoided. Additionally, the energy consumption of Rx nodes is constrained by the capacity of underwater sensors. These kinds of energy-efficient communication methods are necessary. When terrestrial wireless sensor networks (TWSNs) and UAWSNs are compared, it is discovered that TWSNs may take use of many power sources, including wind and solar. On the contrary, UAWSNs are often powered by a battery source to perform functionality that requires periodic replacement [5]. Several further notable distinctions between TWSN and UAWSNs are discussed in [6]. In TWSNs, data packets are transmitted between nodes using electromagnetic waves that propagate at approximately 3×10^8 m/s in free space. In contrast, UAWSNs primarily use acoustic waves, which propagate through seawater at approximately 1,500 m/s, resulting in significantly higher propagation delays than those experienced in TWSNs [7,8]. Because of the high attenuation of acoustic signals underwater, UAWSNs

Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

generally require considerably higher transmission power than TWSNs [6,8]. In order to extend the working life of the nodes that make up a cooperative UAWSNs system, the communication systems in UAWSNs must have such MAC protocols that should be energy efficient. Moreover, in a UAWSN habitat, there are different factors, such as Doppler spread, high delay variation, noise, path loss, and multi-path restrict bandwidth and network energy efficiency [8]. The physical medium environment causes a high data packet bit error rate and a long propagation delay, as shown in [9].

In an underwater habitat, the difficulty of creating a handshaking-based MAC protocol stems from the peculiar properties of an acoustical signal with a significant propagation delay [10]. To significantly reduce handshake-based communication differences in UAWSNs, various approaches have been proposed in [11,12]. There are just a handful of MAC techniques that can increase channel utilization in such circumstances by sending numerous data packets in a cooperative network without adding a known propagation delay. The aforementioned issue may be solved using the Receiver Initiated Data Packet Train (RIDPT)-MAC protocol in [13]. Using a four-way handshaking technique, data packets are sent from the Tx anchor node to the sensor Rx node in the RIDPT-MAC protocol. A hybrid MAC scheme called Reverse Opportunistic Packet Appending (ROPA)-MAC was suggested by [14].

Moreover, developing a MAC technique based on handshakes in [12,14] made it possible for several communication nodes to take part in a typical handshake. Similarly, some researchers proposed a solution using the sender-initiated technique, whereas the others utilized the receiver-initiated technique. In this study, I am considering investigating handshaking strategies for enhancing sensor network performance. In addition, to lessen unanticipated impacts on the receiver caused by concealed neighbors. All ports can maintain a critical path away from the collision produced by hidden nodes with the help of the Hybrid Sender and Receiver protocol in [14], which should have been appropriately defined by a delay interval. As mentioned in [15,16] recent research have utilized RL to MAC in UAWSNs. A learning-based MAC protocol's fundamental benefit is its capacity to create and maintain a (semi-)optimal transmission strategy in dynamic situations. Furthermore, these algorithms can usually be implemented in a decentralised fashion with little overhead in communications, which is very helpful for networks with limited resources functioning in uncertain underwater settings. In order to maximise network longevity, the research in reference [17] specifically used a slotted carrier-sensing multiple access (slotted CSMA) in conjunction with an RL algorithm to optimise packet transmission via the optimal relay node and subchannel to the sink. However, this method requires temporal synchronisation, which is difficult to do underwater, and frequent control packet exchanges for neighbour finding, which may result in overuse of energy. Additionally, the design calls for multichannel communication, which is unfeasible in UAWSNs due to the severely constrained channel capacity. Furthermore, since CSMA requires lengthy guard intervals, the huge propagation delays inherent in UASNs make it largely inefficient.

The slotted floor accusation multiple access (S-FAMA-MAC) protocol in UAWSN s combines a handshaking technique with a collision detection framework [18]. In this protocol, when the packets are sent out at the start of an event, the length of the packet is kept proportional to the longest propagation delay. However, the needless length of the area reduces the throughput performance on such a basic level. The protocol may also analyze the impact of hidden nodes in any scenario.

By enhancing channel usage, the bidirectional concurrent MAC (BIC-MAC) protocol is introduced in [19], to send communicating data packets from Tx anchor node to sensor Rx node continuously for every successful handshaking mechanism. The protocol uses a data packet bursting mechanism, which allows the Tx anchor node and sensor Rx node combination to send out many cycles of bidirectional data packets. As a result, the whole collection of communicating data packets is transferred in multiple discrete bursts. However, the multi-flow case is not taken into account while designing this protocol. The aforementioned protocol is overly complicated, which in turn increases the overall network throughput, and it necessitates a great deal of data packet control, which has an impact on systems performance.

Moreover, maintaining multi-hop channels in series without a delay, [20] mentions CMRT-MAC for UAWSNs (CMRT-UAWSNs). Where communicating data packets are transmitted to sensor Rx node continuously. To improve channel utilization, the aforementioned MAC protocol employs a method that depends on communicating data packet train information. Although these methods improve channel coordination, their repeated reservation procedures and sensitivity to long acoustic propagation delays can limit efficiency in multi-hop and multi-flow underwater networks. A few recent works presented in [21,22] used Q-learning approach for energy-efficient channel availability, collision mitigation, relay selection, and adaptive communication. These methods improve underwater communications. However, their simulation environments differ from this study in terms of channel constraints, learning parameters, and traffic assumptions.

Although reinforcement-learning-based and energy-efficient MAC protocols have improved adaptive channel access and energy utilization in underwater acoustic sensor networks. However, they have some limitations. Existing RL-based MAC approaches mainly adopt learning mechanisms to optimize individual channel-access or transmission decisions, while traditional energy-efficient MAC protocols mainly optimize for reducing collisions, idle listening, retransmissions, or control overhead. However, these schemes offer only limited support for the joint coordination of multi-flow multi-hop and two-way packet transmission in a unified adaptive MAC framework. Conversely, the proposed Q-EEMFTW-MAC integrates Q-learning with multi-flow two-way communication so that communicating nodes can adapt their MAC decisions according to the current transmission state while supporting both forward and backward packet flows. Therefore, the proposed approach is different from existing methods in simultaneously dealing with adaptive MAC decision-making, multi-flow simultaneous communication, two-way packet transmission, end-to-end latency, and energy efficiency in multi-hop UAWSNs.

Our proposed Q-EEMFTW-MAC protocol enables simultaneous operation of several data packet transmission flows while adapting to changing traffic conditions. In contrast to other underwater MAC protocols already in use, our proposed MAC protocol offers a foundation for multi-hop flows by performing directed layer processing on each incoming communication data packet. All requests from the Tx anchor nodes are expected to get acknowledgements from nearby nodes within the cooperative region during the deployment of a data packet flow. Table 1 shows the comparison of the proposed and existing MAC approaches. The following are the major contributions of this work:

a. Proposed a protocol Q-EEMFTW-MAC, to address the discrepancies caused by multi-hop and multi-flow underwater scenarios.
b. Developed an algorithm for the calculation of Tx anchor node missing MFDP, which comprises of three states, i.e., (i) delay_data, (ii) Wait_retry, (iii) Tx_data, with a radio frequency of 1 KHZ.
c. An algorithm for the calculation of relay or sensor Rx node missing MFDP is proposed, which comprise of four states, i.e., (i) Tx_retry, (ii) Tx_data, (iii) Rx_MFDP, (iv) Wait_retry, where the backward_flag = 1 in RDP.
d. Propose an algorithm that elaborates the Tx anchor node transmission in backward flow, which comprises of two states, i.e., (i) Wait_flow, (ii) Backward _Rx, where the Backward_flag = 1 in confirm MFDP, where number of flows = n + 1
e. A network topology is developed to evaluate the proposed Q-EEMFTW-MAC protocol, consisting of static multi-hop nodes deployed within a 6000 m × 6000 m area. The grid spacing between adjacent nodes is fixed at 1000 m, whereas the communication range of each node is set to 1500 m (1.5 times the grid spacing).
f. An end-to-end latency analysis of the suggested protocol is carried out, and compared with the current MAC protocol, in terms of distance, number of nodes, number of flows, and BER.

Table 1. Comparison of proposed and existing underwater MAC approaches.

MAC Approach	Learning-Based Adaptation	Energy Efficiency	Multi-Flow Support	Two-Way/Backward Flow	Multi-Hop Operation
Existing energy-efficient MAC	No/limited	Yes	Limited	Limited	Yes
Existing RL-based MAC	Yes	limited	Limited	Limited	Yes
BIC-MAC	No	Yes	Limited	Yes	Yes
CMRT-MAC	No	Yes	Yes	Limited	Yes
Proposed Q-EEMFTW-MAC	Yes	Yes	Yes	Yes	Yes

The remaining portion of this research is as follows:

Section 2 presents the proposed Q-learning technique, while Section 3 briefly presents the proposed Q-EEMFTW-MAC. Section 4 introduces the results and their discussion, while Section 5 conclude this research.

2. Proposed Q-L Approach

Operating without reliance on a pre-established model of the environment, Q-learning is a model-free RL method. With circumstances that might change suddenly, this feature enables it to adapt to contexts with stochastic rewards and transitions. The phrase “model-free” emphasises how Q-learning is adaptable and strong in a variety of settings since it does not presuppose any particular structure or behaviour of the environment. Its simple update process makes it especially suitable for applications requiring a single agent. In real life, Q-learning begins with parameters that are randomly initialised and learns by constantly interacting with its surroundings. The algorithm gradually improves its decision-making skills by fine-tuning its approach based on the rewards it accrues by assessing the value, or “Q-value”, of adopting certain actions in various situations.

$$Q^{z}(C_z, X_z) = (1 - \beta) \cdot Q^{z-1}(C_z, X_z) + \beta \cdot [I_z + \gamma \cdot \max_{X'} Q^{z-1}(C_{z+1}, X') \quad (1)$$

The action performed during the current time is represented by X^Z in the learning process under discussion and denotes the action selected by the communicating node in that state, and C^Z denotes network state observed at decision step z. At this moment, the learning rate for sensor Rx nodes is represented as α_z. The Q-values, $Q^Z(C^Z, X^Z)$ and Q^{Z-1}, correspond to the current and previous periods, respectively. The incentive received is denoted by I^Z during the shift from period $^{Z-1}$. I^Z denotes the immediate reward obtained after performing the selected action, whereas Y is the discount factor that determines the contribution of future rewards to the current decision. The learning rate (β) controls the degree to which newly observed information overrides prior knowledge and takes a value in the interval (0, 1]. A value of 1 means that the node considers only the most recent observation, whereas a value close to 0 results in very slow learning. The discount factor (γ) determines the importance of future rewards and takes a value in [0, 1]; γ = 0 means that only the immediate reward is considered, whereas γ = 1 means that future rewards are not discounted. The exploration probability (ε) is the probability that a node selects a random action instead of the action with the highest Q-value; ε = 0 corresponds to pure exploitation.

In the Q-EEMFTW-MAC protocol, Q-learning helps a node choose the best MAC operation for its current network. The node observes its MAC state, including packet status and data flow. Then, it chooses to send data, wait, retry, or do backward-flow communication. After executing the selected action, it gets feedback. Reduced delay and energy expenditure are treated as a favorable outcome, whereas unsuccessful transmission, retransmission, collision, or excessive waiting produce an unfavorable outcome. The Q-value updates help the node choose better actions over time. Table 2 shows how Q-learning links to MAC operations in this protocol.

Table 2. Q-learning components and their roles in the proposed Q-EEMFTW-MAC protocol.

Q-Learning Component	Role in Q-EEMFTW-MAC
Agent	Every communicating node that decides when to transmit using adaptive MAC
State	The current MAC and network conditions include packet and flow availability, transmission status, acknowledgment status, and the state of both forward and backward flows.
Action	Choose the right MAC operation. This could mean sending data, waiting, retrying, or handling backward-flow transmission.
Reward	Positive feedback is given when packets are delivered successfully and efficiently with low delay. Negative feedback is given if a transmission fails, a collision occurs, a retransmission is needed, or the delay is too high.
Q-value	The expected long-term benefit of choosing a specific MAC action in the current network condition
Learning rate	Determines newly observed transmission outcomes update the existing Q-value
Discount factor	Determines the importance assigned to future rewards
Learning objective	Choose MAC actions that help multiple data flows succeed while reducing delay, cutting retransmissions, and saving energy.

Moreover, if a node has data to send but does not get confirmation from the next node. The node then checks multiple options, like waiting or retrying. If sending works quickly and efficiently, it gets a better reward, and the Q-value goes up. If it fails or needs more tries, the reward is unfavourable. Next time, the node will pick the action that worked best before.

The reward function of Q-EEMFTW-MAC is proposed to enhance effective packet transmission while restraining high energy consumption, excessive delay, and retransmissions. Accordingly, the reward at decision step *t* can be represented as.

$$R_t = w_s S_t - w_e E_t - w_d D_t - w_r R_t^{(ret)} \quad (2)$$

where S_t means the successful packet delivery, E_t represents the energy cost of the selected MAC action, D_t denotes the resulting transmission delay, and $R_t^{(ret)}$ denotes the retransmission or failed-transmission penalty.

Similarly, Table 3 shows the ablation study of the reward components.

In the proposed Q-EEMFTW-MAC, a learning rate (β = 0.05) is adopted for gradual Q-value updates and stable learning. The Q-values are iteratively updated during learning based on the observed state, the action taken by the MAC, and the reward received. If the difference between successive Q-value updates is sufficiently small and the preferred action does not change from iteration to iteration, then the algorithm converges. After the convergence of the algorithm, the learned Q-values are used to decide the MAC operations for efficient packet transmission, waiting, and retransmission.

Table 3. Ablation study of planned reward components.

Reward Configuration	Reliability Term	Energy Term	Delay Term	Retransmission Penalty
Full reward	Yes	Yes	Yes	Yes
Without energy term	Yes	No	Yes	Yes
Without delay term	Yes	Yes	No	Yes
Without reliability term	No	Yes	Yes	Yes
Without retransmission penalty	Yes	Yes	Yes	No

3. The Proposed Q-Learning-Based Energy Efficient Multi-Flow Two-way (Q-EEMFTW-MAC) Protocol

The proposed Q-EEMFTW-MAC protocol aims to increase the performance of the existing CMRT-MAC and the DR-ALOHA-Q, used for UAWSNs in [20,21]. The suggested MAC protocol regulates the control frame structures and time intervals of the delivered data packets. Multi-flow and multi-hop data packet forwarding using the backward data packet techniques are considered in the simulation. The state transition diagrams of the Tx anchor node and the sensor Rx node, which elaborate the proposed algorithm, are presented in Sections 3.6 and 3.7, respectively.

A single communication node may create many data flows in the proposed Q-EEMFTW-MAC protocol, enabling the Tx anchor node to deliver different data packets via numerous flows, as shown in Figure 1. Furthermore, multiple sinks are considered in the network architecture model.

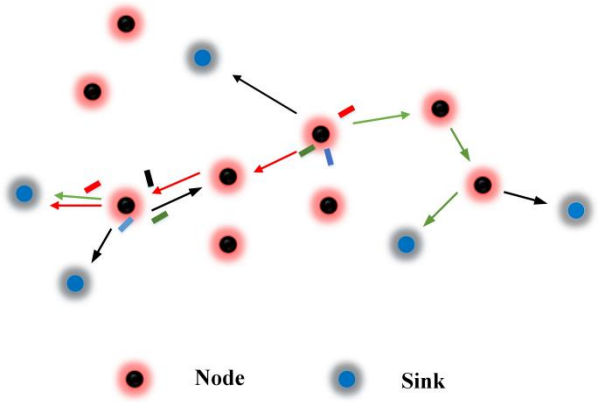


Figure 1. Network model of Q-EEMFTW-MAC protocol.

3.1. Communication Flow Setup

As illustrated in Figure 2, the proposed Q-EEMFTW-MAC protocol is designed to organize various Two-Way communicating data packets that require multi-hop connections by channel reservation. When planning the allocation of communication channels, it considers of any incomplete requests for communication in the flow setup as well as all pending communicating data packets in the routing layer.

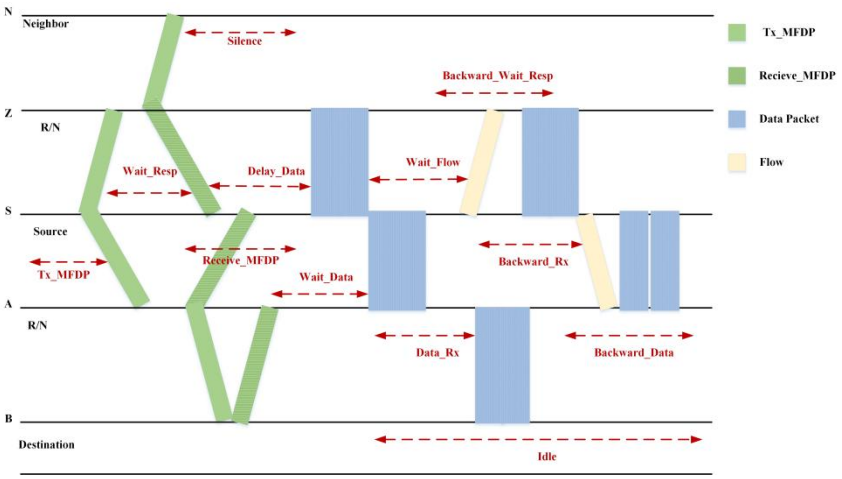


Figure 2. The Q-EEMFTW-MAC Protocol states.

As a result, the MAC layer is able to detect changes in traffic volume and may track and monitor data packet transmission buffer durations. As shown in Figure 3, a multi-flow communicating data packet is composed of a series of control frames that are delivered over a number of hops. Moreover, the Multi-flow communication data packet contains a few cross-layers communicating data packets, such as the addresses of the ultimate sensor Rx node from which the data isbeing sent to anchor node, and the number of hops. This ultimate sensor Rx node information has been sent down by the network layer.

3.2. States Foundation of the Q-EEMFTW-MAC Protocol

A communicating node in the Q-EEMFTW-MAC protocol operates through fourteen distinct states, as illustrated in Figure 2. Six of these states are based on the existing CMRT-MAC protocol [20]. In the Tx Multi-Flow Data Packet (MFDP) state, the Tx anchor node sends an MFDP request to a repeater node. Upon receiving confirmation from the sensor Rx node, the Tx anchor node enters the Rcv_MFDP state. After receiving a batch of forward data packets, the sensor Rx node enters the Wait_Flow state and remains there until it receives a Clear-to-Send (CTS) packet from the Tx anchor node, authorizing backward-flow transmission.

A Tx anchor node is awaiting a CTS from a sensor Rx node while in the state Backward_Wait_Resp. And after transferring data packets, if a Tx anchor node detects that its backward flag is selected, then it will require to wait and restart sending the communicating data packet in backward direction. Another status is Backward_Data, in which a Tx anchor node waits for data packet transmission in order to avoid collisions induced by two-way data packet transmission. Meanwhile, The Tx anchor node enters the Backward_Data state when it accepts a CTS from the sensor Rx node and stays there until all of the backward communicating data packets have been sent. The Backward_Rx is a condition in which a sensor Rx node, receives communicating data packets in the backward direction of flow. To deal with any delayed control, there are two states: Tx_Retry and Wait_Retry. In the Tx_Retry state, A sensor Rx node sends a second control data packet called a Retry data Packet (RDP), if it doesn't get a communicating data packet from the appropriate Tx anchor node. If a Tx anchor node misses a confirm MFDP, it enters in the Wait_Retry state and waits for an RDP from the sensor Rx node.

3.3. The Q-EEMFTW-MAC Protocol Operation and Topology

The Q-EEMFTW-MAC protocol operation and topology are shown in Figure 3 and Figure 4, respectively. S stands for the Tx anchor node in Figure 4. Assume that the sensor Rx node C and sensor Rx node X have a cluster of communicating data packets on a Tx anchor node S. The direction of a single flow between the sensor Rx node X and the Tx anchor node S is maintained by the repeater nodes Z and Y. Furthermore, both Repeater nodes A and B are still in the direction of a different flow between Tx Anchor node S and Sensor Rx node C. Consequently, through repeater nodes A and B, the Tx anchor node S sends communicating data packets to the sensor Rx node C. Through repeater nodes Z and Y individually, the Tx anchor node S provides communicating data packets to sensor Rx node X. In the proposed Q-EEMFTW-MAC protocol, one new communicating control data packet is introduced, when a Tx anchor node starts to transmit MFDP to repeater node Z and repeater node A. When the Tx anchor node S transfers a Multi-flow data packet using this procedure, it enters the Wait_Resp state. By delivering a Multi-flow data packet to repeater nodes Z and A, Tx anchor node S initiates the handshaking mechanism. where the initial sensor Rx node address is held by repeater node Z, and the second sensor Rx node address is held by repeater node A. The method used to schedule these addresses depends on the distances between communicating nodes. When sending data packets across a UAWSN, the communicating nodes in the network that are closest to one another will receive precedence.

In Q-EEMFTW-MAC, the MFDP-based channel-reservation process coordinates multiple communication flows. For the Tx anchor node with multiple flows, the next-hop nodes are determined and scheduled based on the communication distance and propagation delay. The MFDP exchange reserves the required communication opportunity for every active flow before the data transmission. After a successful reservation, the data packets of different flows can be sent pipelined over different hops. Delay_Data and Wait_Flow states provide the required waiting periods between the possibly conflicting transmissions. Backward-flow communication is done via CTS-based coordination. These mechanisms reduce the probability of packet collisions by avoiding overlapping transmissions at nodes that transmit at conflicting times.

Congestion due to competing flows is handled by taking into account pending packets and incomplete communication requests when establishing flows, rather than allowing all flows to access the channel regardless of others. Missing or failed MFDP exchanges are handled with the Wait_Retry and Tx_Retry states and the RDP mechanism. As a result, failed flows can retry and are not permanently blocked from using the channel.

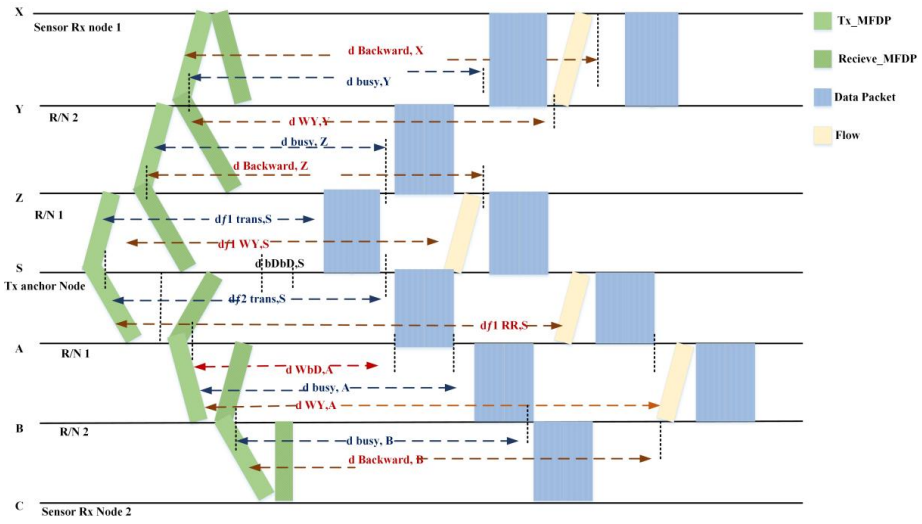


Figure 3. The Q-EEMFTW-MAC protocol operation.

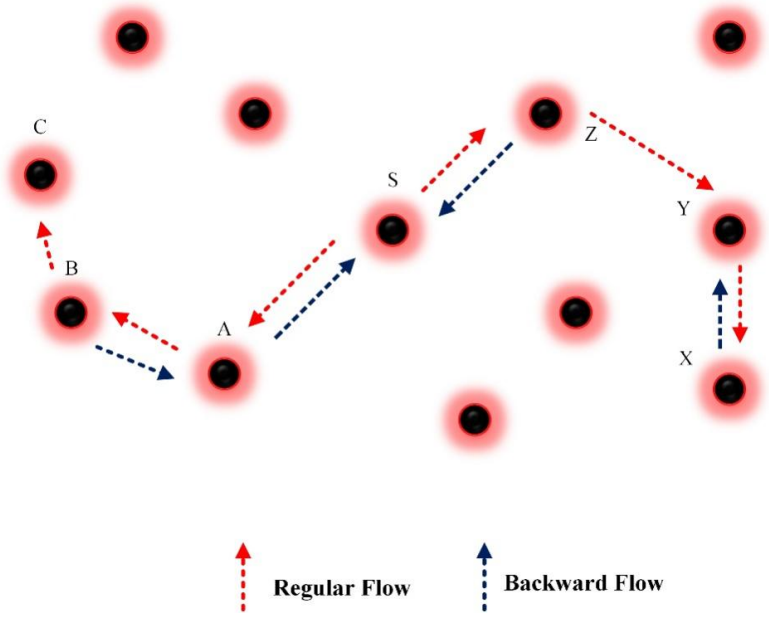


Figure 4. Network Topology of the proposed MAC protocol.

In the forward communication direction, the initial repeater node Z delivers a Multi-flow data packet in the direction of repeater node Y, whereas in the backward route, Tx anchor node S overhears Multi-flow data packets. At the same moment, the second repeater node A delivers a Multi-flow data packet in the forward direction to repeater node B, while Tx anchor node S overhears a Multi-flow data packet in the backward direction. In the proposed Q-EEMFTW-MAC protocol, I use the propagation delay between communicating nodes because of communication distances, to allow these two directional communicating flows to occur simultaneously. Tx anchor node S enters in the Delay_Data state after sending Multi-flow data packets through repeaters. Repeater node A and repeater node Z enters in the Wait_Data state while communicating. And the repeater node A follows the same procedures as repeater node Z after receiving Multi-flow data packets from Tx anchor node S.Repeater node Y receives data packets from the previous repeater node Z Multi-flow data packets at the same time and conducts the same steps as repeater node Z. This cycle of receiving a Multi-flow data packet and transmitting other Multi-flow data packet will continue until the ultimate sensor Rx nodeX and sensor Rx node C will accept the Multi-flow data packets. The Multi-flow data packet performs a key role in this case, which is to reserve information in sensor Rx nodes across various hops and various flows in a cascaded manner. As a result, it may effectively minimize the time taken by handshaking-based methods during the communication of data packet delivery times. For both backward and normal flow data packet communications, repeater node Z has the first in order scheduling priority due to its physical address. If Tx anchor node S got confirmation of Multi-flow data packets from its future repeater node Z, then the Tx anchor node S will send the communicating data packets frame to repeater node Z,

after the Delay_Data ($bD^{f1}_{bDbD,S}$) time period. However, the repeater node Z can simultaneously communicate a data packet train to another repeater node Y, with no wait for acknowledgement between data packets. Because the multi-hop communicating channels are restricted targeted and targeted towards the sensor Rx nodes. And the repeater node Y likewise forwards the data packet train with no delay. Tx anchor node S delays the data packet communication for Delay_Data ($bD^{f1}_{bDbD,S} + d_{data}$) to avoid any collisions from data packet delivery by changing the direction of the flow of data packets. The second flow of communication data packets delivery mechanism also works on the same principle as in the first. At each hop, the transmission of data packet frames continues until the ultimate sensor Rx node C is received. As a result, the repeating of data packets is much like a pipeline operation. In our opted Q-EEMFTW-MAC protocol, communication data packets from two separate flows can be provided at the same time.

3.4. Data packet transmission using the Backward Data Packet Communication Method

Through the use of the opposite multi-flow data packet technique, the control MFDP obtains a communicating data packet transmitting technique by transmitting different data packets from similar data packets flows in the backward direction. Thus, with it the use of communication channel will increase consequently.

Figure 5 shows how the proposed Q-EEMFTW-MAC protocol uses the backward data packet methodology. The backward data packet transmissions, start after the successful transfer of communicating data packets from Tx anchor node S to sensor Rx node X and Tx anchor node S to sensor Rx node C. When transferring Multi-flow data packets, every communicating node with data packets that flows in backward direction sets its backward flag to 1. When a communicating node detects a data packet in backward directional flow. It transmits a CTS data packet to the Tx anchor node S, which immediately sends the communicating data packets to sensor Rx node C.

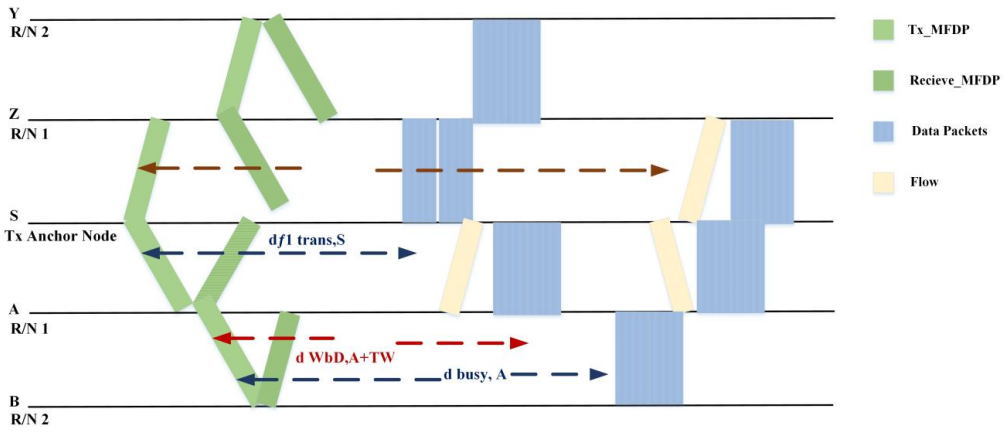


Figure 5. A single confirmation slip from the Tx anchor node.

Assume that there are various backward data packets after repeater node Z sends an acknowledgement to Tx anchor node S and sensor Rx node Xand also to repeater node Y during the first flow between Tx anchor node S and sensor Rx node X. Consequently, Tx anchor node S sets its backward flag to 1 while transferring Multi-flow data packets from Tx anchor node S to repeater node Z. Furthermore, repeater node Y also transmits Multi-flow data packets to sensor Rx node X in the same way. Tx anchor node S needs to go into the Wait Flow state after transferring data from Tx anchor node S to repeater node Z and Tx anchor node S to repeater node A. Then communicates a CTS to the repeater node Z and with it simultaneously communicates data packets to Tx anchor node S. Moreover, because there is no communication solely on a single hop distance of repeater node Y. When backward transmission of data packets from repeater node Y to sensor Rx node X is completed, repeater node Y transmits a CTS to sensor Rx node X, and sensor Rx node X immediately transmits backward communicating data packets to repeater node Y.

Assume that repeater node A has batches of data packets for Tx anchor node S and Tx anchor node S has batches of data packets for repeater node A. Repeater node A is specified as the 2nd repeater node, thus when repeater node Z performs its backward and regular communication. The repeater A node begins its backward flow of data packet. Tx anchor node S sends a CTS to sensor Rx node X, and repeater node A sends a communicating data packets to Tx anchor node S straight away. The data transmission from Tx anchor node S to sensor Rx node X operates in the similar pattern. The waiting time before a node starts its backward transmission is calculated as given in Algorithm 1.

Algorithm 1: **Calculation of Waiting Backward Time**

- **If** backward_flag = 0 where MFDP is requested for R_{n-1} **then**
- $d_{backward}[R_n] = d_{WY}[R_{n-1}] + T_{CTS}$ + network propagation delay
- **else**
- $d_{backward}[R_n] = d_{WY}[R_{n-1}] + 2T_{CTS}$ + 2network propagation delay + d_{data}
- **end if**

3.5. Data Packet Transmission Using the Request Data Packet Communication Method

1st Scenario: Tx anchor node S fails to receive a confirmation: Assume that, if a Tx anchor node S doesn't receive confirmations of Multi-flow data packets in response to the repeater node A. When repeater node A doesn't get data packets from Tx anchor node S after $d_{WD,Rx,E} + T_W$ time.Repeater node A assumes that Tx anchor node S has not received its confirmation of Multi-flow data packets, as a result, repeater node A transmits a relay data packet (RDP), and Tx anchor node S will begin transmitting second-flow communication data packets to repeater node A.

2nd Scenario: Tx anchor node S fails to receive a confirmation: The Tx anchor S node doesn't obtain confirmations from both repeater nodes Z and A. When repeater node Z doesn't receive data packets from Tx anchor node S after $d_{WD,E} + T_W$ time, repeater node Z sends a RDP, and Tx anchor S node immediately begins to transmitting data packets for the 1st flow to repeater node Z. When repeater node A doesn't receive communicating data packets from Tx anchor node S after $d_{WD,A} + T_W$ interval. Repeater node A transmits a relay RDP, and Tx anchor node S immediately begins transmitting data packets for the 2nd flow to repeater node A.

3rd scenario: If A middle node fails to receive confirmation: Assume that the repeater node Z doesn't receive confirmation of a Multi-flow data packets from repeater node F because of data packet collision. Repeater node Z at that instant will broadcasts RDP and will immediately begin relaying the data packets to repeater node F.

4th Scenario: An intermediate sensor Rx node fails to receive confirmation: Assume that repeater node B doesn't get confirmation from sensor Rx node C regarding its request. When sensor Rx node C doesn't get the data packets from repeater node B after $d_{WD,C} + T_W$ time. Then the sensor Rx node C sends a repeater data packet (RDP) to node B immediately, which begins transmitting data packets to sensor Rx node C.

5th Scenario: Assume that once Tx anchor node S has sent the repeater Multi-flow data packets, the repeater communicating node does not receive the desired Multi-flow data packets. The repeater doesn't wake up, and the Tx anchor sender node S waits for the repeater node to deliver a RDP. Tx anchor node S will assume that its request had been lost, if it does not get a RDP. In the case it will transmit a Multi-flow data packet again after entering the Wait_Retry state. For the better understanding of all the scenarios, Figure 6 can be referred.

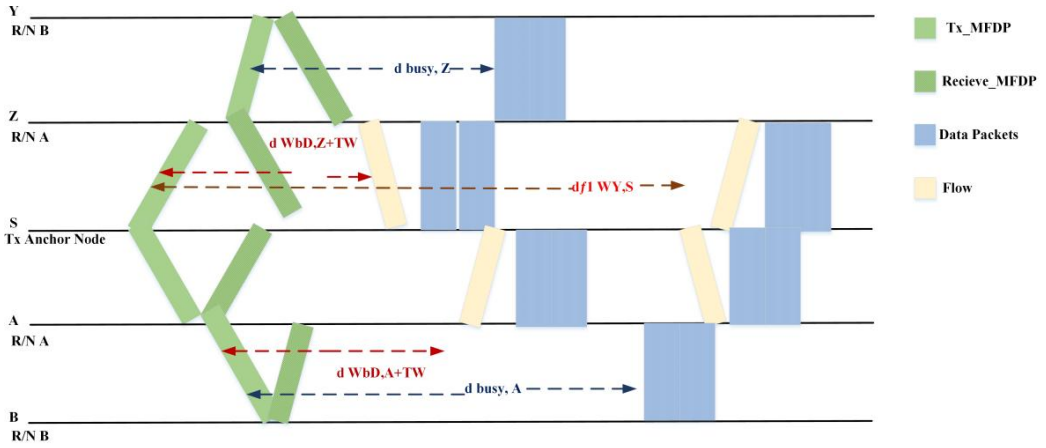


Figure 6. All confirmation slips from the sender.

3.6. Anchor Node State Transitions

Figure 7 depicts the transition state diagram of a Tx anchor node S in the proposed Q-EEMFTW-MAC protocol. When there is no activity on a Tx anchor node S, it maintains its idle condition. When a Tx anchor node S creates a new data packet while the channel is idle, the communicating node shifts to the Tx anchor node S multi-flow data packet state.

The Tx anchor node transmits multi-flow data packets to the repeater node (R/N) via multi-flows in the Tx_anchor node S multi-flow data packet state. Tx anchor node S enters into the Wait_Resp state after sending

multi-flow data packets, where it waits for the time to receive a multi-flow data packet from the associated repeater nodes. The Tx anchor node S estimates T^{n}_{RMFDP} the Multi-flow data packets reception time in this state. The Tx anchor node S shifts to the Rcv_Multi-flow data packet state if the transmitting time of a node is equal to the receiving time of a Multi-flow data packet for n number of flows. The Tx anchor node S enters into the Wait_Resp state after receiving Multi-flow data packets for a given number of flows for n. In the Wait_Resp state, the Tx anchor node S waits for the next receiving Multi-flow data packets T^{RMFDP}_{n+1} to arrive. If the receiving time for Multi-flow data packets increases, it again returns to the Rcv_Multi-flow of data packets to n + 1 number of flows. This state shifting between Wait_resp and Rcv_Multi-flow data packets continues until the amount of flows n for multi-flow data packets transmission would be less than the flow number of constructions. If f = total-number-flow, the Tx anchor node S goes to the Delay_Data state. The Tx anchor node S estimates $d^{n}_{trans,S}$ time to sends communicating data packets for similar streams of n in the Delay_Data state. In this condition, the Tx anchor node S waits for the d_{bDbD} period. A Tx anchor node S can transition from state Delay_Data with one of two or more states, depending on the situation.

1st scenario: if the Tx anchor node S gets all of the relevant acknowledged Multi-flow data packets, it enters in to the Tx_Data state and begins sending data packets to that same repeating nodes. The Tx anchor node S remains in the Tx_Data state until all forward flows have received.

2nd scenario: the Tx anchor node S waits to retry data packets (RDP) if it misses a Multi-flow data packets for any of the continuous flow of data packets with in the UAWSNs. The Tx anchor node S shifts to the Tx_Data state after receiving RDP for n number of flows and sends data packets for that specified RDP. This shifting between statuses from Wait_Retry to Tx_Data continues until the Tx anchor node S receives acknowledgements for all constructive flows that highly depends on RDP.

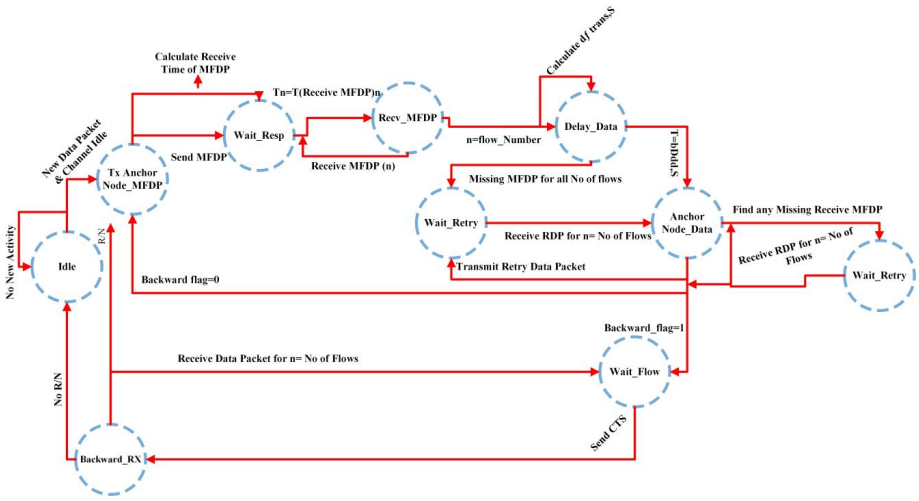


Figure 7. Tx Anchor node sender transition diagram.

Three scenarios can occur in the Tx_Data state.

1st Scenario, if the Tx anchor node S doesn't miss a Multi-flow, then backward_flag is confirmed that a Multi-flow data packet has entered the Wait_Flow state and waits for d_WY,S^n plus the network propagation delay to prevent collisions. The Tx anchor node S then transmits the CTS data packet and shifts to the Backward_Rx state. In this state, it receives the backward communicating data packets of the same flow. It then returns to the Wait_Flow state and waits for d_WY,S^(n+1) plus the network propagation delay before transmitting the CTS for the next flow (n + 1). As soon as the data packet transmission through backward flow across all the f number of backward flows is finished. The Tx anchor node shifts between these two transition states Backward_Rx and Wait_Flow.

2nd Scenario, If the Tx anchor node S misses an MFDP, it enters the Wait_Retry state and waits for d_bDbD,S + T_W. Here, d_bDbD,S is the waiting period defined in the Delay_Data state, which allows the repeater nodes to finish their MFDP exchange, and T_W is an additional guard time for the RDP to arrive from the repeater node. If the RDP is received within this time, S enters the Tx_Data state and sends the data packets of the corresponding flow. The procedure followed by the Tx anchor node S when Multi-flow data packets (MFDP) are missing is summarized in Algorithm 2.

3rd Scenario, if Tx anchor node identifies backward_flag = 0, it knows that no data packets are being transmitted in the backward flow direction, consequently it enters into Idle state.

Algorithm 2: Calculation of Tx anchor node missing MFDP

- **for** n = 1 to f flows **do**
- **if** state = Delay_Data and all MFDP are missing **then**
- **go to** the Wait_Retry state
- wait for d_bDbD,S + T_W for the reception of an RDP
- **if** n < f **then**
- remain in the Wait_Retry state for the next flow (n + 1) **else**
- **go to** the Tx_Data state
- **end if**
- **else if** state = Tx_Data and any MFDP is missing **then**
- **for** each missing MFDP **do**
- **go to** the Wait_Retry state
- wait for d_trans,S^(n-1) + network propagation delay + T_W for the reception of an RDP
- **if** an RDP is received **then**
- send data
- **end if**
- **end for**
- **end if**
- **end for**

3.7. (R/N) State Transition Diagram

The state transition of a repeater/relay node (R/N) in the proposed Q-EEMFTW-MAC protocol is shown In Figure 8. When there is no action to do, a repeater node stays in the idle state. In the idle state, if an R/N wishes to repeat data packets, while the communication channel is idle, it shifts to the TX_Multi-flow data packet state. The requested Multi-flow data packet is then sent across multiple flows to the next R/N $R_{(n+1)}$. Repeater node than enters the Wait_Resp state after transmitting a Multi-flow data packet, and waits for the time T_{RMFDP} to receive a Multi-flow data packet from the next repeater node. The R/N calculates the T_{RMFDP} receiving time of the responded Multi-flow data packet in this state. The repeater node switches to the Rcv_Multi-flow data packet state. Where it receives the confirmation of a Multi-flow data packet, if the sending time is directly proportional to the receiving time of a Multi-flow data packet.

The R/N enters the Wait_Data state after receiving the Multi-flow data packets from the next R/N. R/N calculates the waiting time $d_{WbD,}R_n$ in the Wait_Data state and waits for the duration to get data from $R_{(n-1)}$ or Tx anchor node. There are two possible scenarios in the Wait_Data state.

1st Scenario: if T = $d_{WbD,}R_n + T_W$ and there is no data packet received in T time, the communicating R/N enters to Tx_Retry state, and sends a RDP. After which it then entersin to the Data_Rx state.

2nd Scenario: if T = $d_{WbD,}R_n$ and communicating data packets are delivered at T time, the communicating R/N enters to Data_Rx state and starts connecting the data packets from their related R/N $R_{(n-1)}$ or Tx anchor node.

R/N can move from the Data_Rx state to four other different transition states depending on the following scenarios.

1st Scenario: if the R/N receives a data packet, and finds confirmation of a Multi-flow, then the R/N moves toTx_Data state and starts transmission of the data packets to its next R/N.

2nd Scenario: if the R/N gets data packets but does not receive confirmation of Multi-flow data packets, it goes toWait_Retry state. The R/N then waits for the RDP to receive a R/N transfer acknowledgement from the Tx_Data state of communicating data packet.

3rd Scenario: if backward_flag = 1 and the R/N receives a data packet, then the R/N enters to Backward_Wait_Resp state.

4th Scenario: if the R/N receives data packets and R/N is the final communicating node on the flow with backward_flag = 0, R/N goes to the Idle state. The corresponding procedure for a relay or sensor Rx node that misses an MFDP is given in Algorithm 3.

In the Tx_Data state, a communicating node sends data packets to its R/N. There are three possible scenarios in the Tx_Data state.

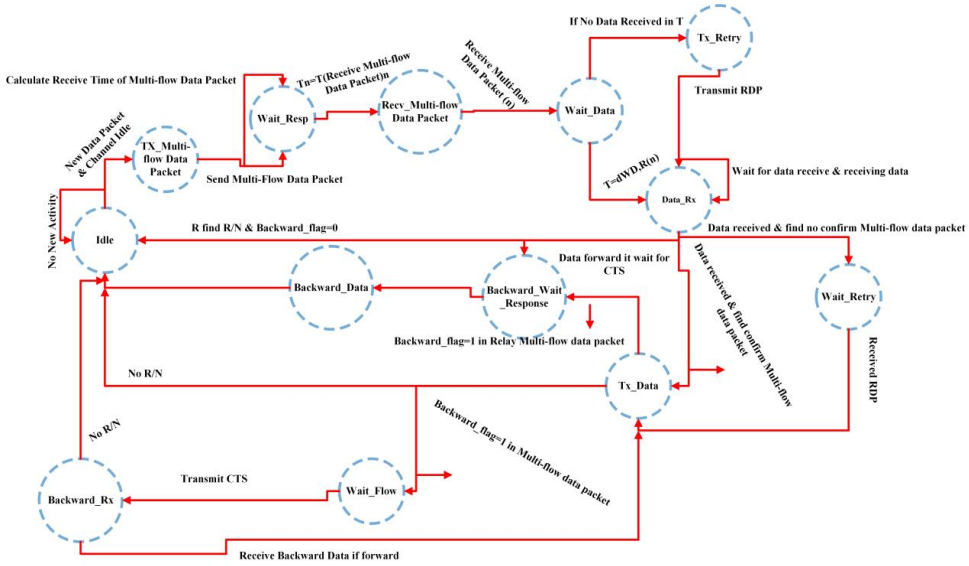


Figure 8. State-transition diagram of a repeater/relay node (R/N) in the proposed Q-EEMFTW-MAC protocol.

1st Scenario: if the R/N forwards a communicating data packet that acknowledge a backward_flag = 1 in requested MFDP, it shifts to the Wait_Flow state. In the Wait_Flow state, the R/N calculates the time $d_{WY,}R_n$ and waits for the time $d_{WbD,}R_n$ to avoid a collision. The R/N shifts to the Backward_Rx state after delivering CTS data packets to the interrelated R/N. The R/N receives communicating data packet in backward flow, when it is in the Backward_Rx state. If data packets are mandatory to be forwarded, the R/N shifts to the Tx_Data transition state. If R/N have no data packets after transmitting data packets in Tx_Data transition state, the R/N enters to idle state.

2nd Scenario: the node enters in the Backward_Wait_Resp state while forwarding data packets, and if a backward_flag = 1 is detected in response to Multi-flow data packets. The R/N measures the time $d_{Backward,}R_n$ and then again waits for the time $d_{WbD,}R_n + T_{CTS}$ to prevent a collision in this transition state. The communicating node enters to Backward_Data state when the communicating node receive response data packet CTS from the specified R/N. In the Backward_Data transition state, the nodes communicate backward data packets, if no R/N is present, then the R/N enters to Idle state.

3rd Scenario: if communicating data packets are not available in R/N, the R/N enters to the Idle state.

Algorithm 3: Calculation of Relay or sensor Rx node missing MFDP

for n = 1 to f data flows **do**

if T = d_WbD,R_n + T_W and no data packet is received **then**

go to the Tx_Retry state and send an RDP

end if

if T = d_WbD,R_n + T_W + T_CTS + network propagation delay **then**

go to the backward data state (send CTS and receive backward data)

update d_WY,R_n and d_backward,R_n

else if state = Tx_Data and no other MFDP is received **then**

go to the Wait_Retry state

receive RDP and transmit data packet to R_(n+1)

update d_WY,R_n and d_backward,R_n if backward_flag = 1 in RDP **then**

return to the first step for the n data flows **else**

go to the Idle state

end if

end if

end for

3.8. Simulation Setup and Performance Evaluation

Figure 9 depicts the topology of multi-hop static communicating nodes arranged in a 6000 × 6000 m^2 region for our analysis. The grid spacing between two adjacent nodes is 1000 m, and the communication range of each node is 1.5 times the grid spacing (1500 m). Every communicating node in the network is assumed to have the same communication power. The red dotted circle in Figure 9 indicates that all of the communicating nodes in the network have approximately eight neighboring nodes, within their range. The average transmission and receiving power of the network communicating nodes is 2 W and the acoustic transmitter and receiver is 35 mW. It is considered that the acoustic communication channel is prone to errors.

I evaluated our proposed protocol with different network and traffic situations. To ensure a fair comparison between MAC protocols, I kept the network setup, communication features, traffic patterns, and physical-layer settings the same for every test. Table 4 shows the main simulation configuration.

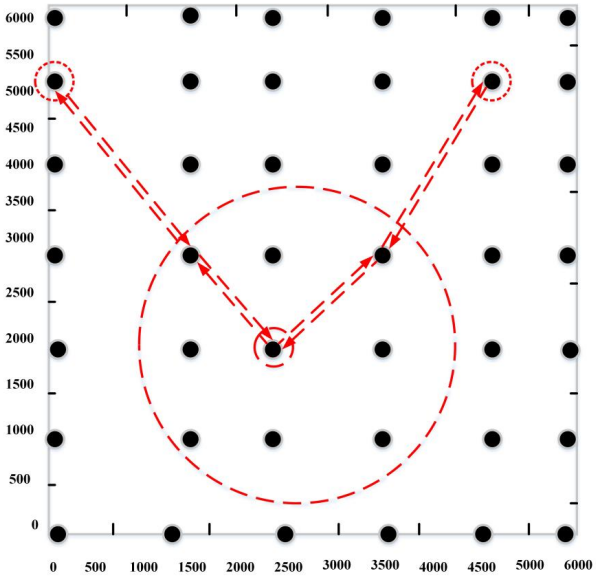


Figure 9. Proposed Network topology for Q-EEMFTW-MAC.

Table 4. Parameters for the proposed Q-EEMFTW-MAC protocol.

Parameter	Value
Network topology	Static multi-hop grid
Network area	6000 × 6000 m^2
Grid spacing	1000 m
Communication range	1500 m
Number of nodes	5–40
Approx. neighbouring nodes	8
Tx–Rx distance	1–8 km
Number of hops	1–10
Number of flows	1–4
Number of backward packets	1–6
Offered traffic load	0.2–3 packets/s/node
BER evaluation	0–0.1 (reference value 10^{-3})
Reference BER	10^{-3}
Acoustic propagation speed	1500 m/s
Acoustic carrier frequency	1 kHz
Data packet size	1024 bits
Control packet size	128 bits
Transmission power	2 W
Receiving power	2 W
Idle power	35 mW
Simulation duration	1000 s
Q-learning algorithm	Model-free Q-learning
Learning rate, (β)	0.05
Comparison protocols	CMRT-MAC and DR-ALOHA-Q

The proposed EEMFTW-S is a variation of the Q-EEMFTW-MAC protocol that only considers multi-flow data packet communication, while the Q-EEMFTW-R/N, considers multi-flow backward data packet transmission. I evaluated Q-EEMFTW-MAC, CMRT-MAC, and DR-ALOHA-Q under the same simulation setup. I used the same parameter setup for each protocol. In each experiment, I changed only the MAC operation for the protocol being tested. The Q-EEMFTW-S represents the standard multi-flow transmission case, in which data packets are transmitted through the regular forward communication flows. In contrast, Q-EEMFTW-R/N represents the relay/sensor receiver-node case, in which the protocol additionally supports backward data packet transmission through relay or receiving nodes and both are evaluated separately because backward-flow communication introduces additional transmission opportunities and different channel-reservation and forwarding behavior.

The power parameters were selected based on the relative energy costs of the primary operational modes of underwater acoustic communication; however, actual power consumption varies between commercial acoustic modems and operational scenarios. Therefore, an important future step for experimental validation is the evaluation based on specific commercial modem power profiles.

The proposed Q-EEMFTW-MAC protocol's effectiveness in terms of network throughput, network end-to-end latency, and power consumption is simulated using MATLAB. The backward-flow transmission procedure of the Tx anchor node is described in Algorithm 4.

Algorithm 4: Tx anchor node transmission in backward flow

- **if** state = Wait_flow **then**
- **for** i number of flows sends
- **if** T = $d^{n}_{trans,S}$ + network propagation_delay **then**
- Step No.2: transmit CTS to sensor Rx node of n number of flows.
- Come to Backward_Rx state
- **if** T = $d^{n}_{WY,S}$ + network propagation_delay **then**
- Receive Backward data packet from R/N n number of flows
- **If** Backward_flag = 1 in confirm MFDP, where number of flows = n + 1
- **Then** again comes to step No.2
- **else** comes to idle state
- **end if**
- **end for**
- **end if**
- **end for**

3.9. Time Duration Parameters Calculation

The maximum transmission delay between these communicating nodes are considered as τ_{max}. The common communication time for all control data packets is $T_{control}$. Assume that $d^{f1}_{data,P}$ is the batch communicating data packet transmission time for first flow in the network and $d^{f2}_{data,P}$ is the batch communicating data packet transmission time for second flow in the network. Following is The busy time of the Tx anchor node for the second flow in Figure 3 is as follows.

$$d^{f1}_{busy,S}, = 2\tau_{max} + T_{control} + d^{f1}_{bDbD,S} \quad (3)$$

The busy and waiting time of the repeater node Z is:

$$d_{W\,bd,Z} = 2\tau_{max} + T_{control} \quad (4)$$

$$d_{busy,Z} = d_{W\,bD,Z} + d^{f1}_{data,P} \quad (5)$$

Assume that T_{data} represents the duration of a single data packet's transmission and that $B^{data,P}_{size}$ Z represents the size of a batch data packet. Following are the calculations for the repeater node Y busy time duration.

$$d_{busy,Y} = (d_{busy,Z} - T_{control} + B^{data,P}_{size} T^{f1}_{data,P} = 2\tau_{max} + 2B^{data,P}_{size} T^{f1}_{data,P} \quad (6)$$

The first flow busy time duration of repeater node R/N may be generalized as:

$$d_{busy,R_n} = 2 + nB^{data,P}_{size} T^{f1}_{data,P} - (n-2)T_{control} \quad (7)$$

The busy time of Tx anchor node for the second flow in Figure 9 as follows:

$$d_{busy,S}^{f1} = d_{busy,S}^{f1} + d_{data,P}^{f1} \tag{8}$$

The busy and waiting time of repeater node A is:

$$d_{W\,bd,A} = 2\tau_{max} + T_{control} + d_{data,P}^{f1} \tag{9}$$

$$d_{W\,bd,A} = d_{W\,bd,A} + d_{data,P}^{f2} \tag{10}$$

The busy time of node B is.

$$d_{busy,B} = \left(2\tau_{max} + d_{data,P}^{f1}\right) + B_{size}^{data,P} T_{data,P}^{f2} \tag{11}$$

The total number of flows in the network are denoted by f. Therefore, the busy time of the repeater node R/N for the second flow is as follows:

$$d_{busy,R_n} = 2\tau_{max} + d_{data,P}^{f1} + nB_{size}^{data,P} T_{data,P}^{f2} - (n-2) \tag{12}$$

$$d_{W\,Y,S}^{f1} = d_{busy,S}^{f2} + d_{data,P}^{f2} \tag{13}$$

Assume that the transmission time of CTS is TCTS data packets. Therefore, the time of propagation of the backward flow of repeater node E is as follows:

$$d_{backward,Z} = d_{W\,Z,S}^{f1} + T_{CTS} \tag{14}$$

and the time taken by a repeater node Y is as follows:

$$d_{W\,Y,Y} = d_{busy,Y} + d_{data,P}^{f1} \tag{15}$$

4. Results and Discussions

In our architecture, I looked at how the distance between the sensor Rx node and the Tx anchor node affected the networks' throughput performance. In the proposed UAWSNs architecture, the effects of various directional flows and provided loads are tracked and quantified using the bit error rate (BER). The network throughput is calculated as Throughput = (N_DP × N_pDP) / T_R, where N_DP is the number of successfully delivered data packets, N_pDP is the number of bits per data packet, and T_R is the total required time in seconds. The throughput is therefore expressed in bits per second (bps). In Figure 10, when the network offered load is estimated to be 0.9 data packets per second, the effects of distance between Tx anchor node and sensor Rx node of the proposed MAC protocol and the existing MAC protocol with bit error rate 10^{-3} are depicted.Due to the lengthy propagation delay in UAWSNs, all MAC protocols perform worse in terms of throughput as the distance between the Tx anchor node and sensor Rx node increases. The handshaking duration also increases as the distance between the Tx anchor node and the sensor Rx node increases.

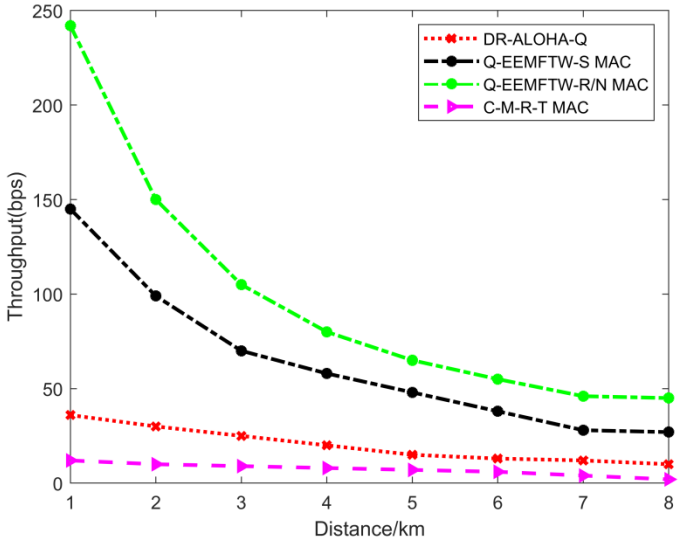


Figure 10. Throughput vs. Distance between Tx anchor node and sensor Rx node.

Therefore, the ability of a node to transmit control data packets is restricted. In multi-flow UAWSNs, it can be demonstrated that, the proposed MAC protocol can achieve better performance than the existing MAC protocol in terms of controlling changeable traffic flows. Furthermore, the proposed Q-EEMFTW-R/N and Q-EEMFTW-S, as shown in Figure 10, may achieve throughput performance that is about 84.3% and 46.7% more than the current MAC protocol, where the distance among the Tx anchor node and sensor Rx node is 1 km. Consequently, the proposed Q-EEMFTW-R/N and Q-EEMFTW-S achieved a throughput performance that was about 28.1% and 19.2% increase, as compared with the current MAC protocol, where the distance among the Tx anchor node and sensor Rx node was 4 km.

Figure 11 contrasts the network throughput performance of the proposed MAC protocol with the current MAC protocol, where the bit error rate is 10^{-3}. By applyingthe MAC protocol, the network can now utilize the whole channel for communication. So, it's different from standard network throughput per node in this regard. It is also noted that, in terms of channel utilization, the performance of the proposed MAC protocol is greater than other existing MAC protocols. The results show that, the existing MAC protocol is unsuccessful, where a Tx anchor node sends communicating data packets across a single flow per round channel reservation. However, in the proposed protocol a single Tx anchor node transmits a communicating data packets to multiple nodes in a flow with each round having a channel reservation. In order to increase channel performance, the proposed protocol has the opportunity to significantly reduce the cost of fast and hard channel reservations.

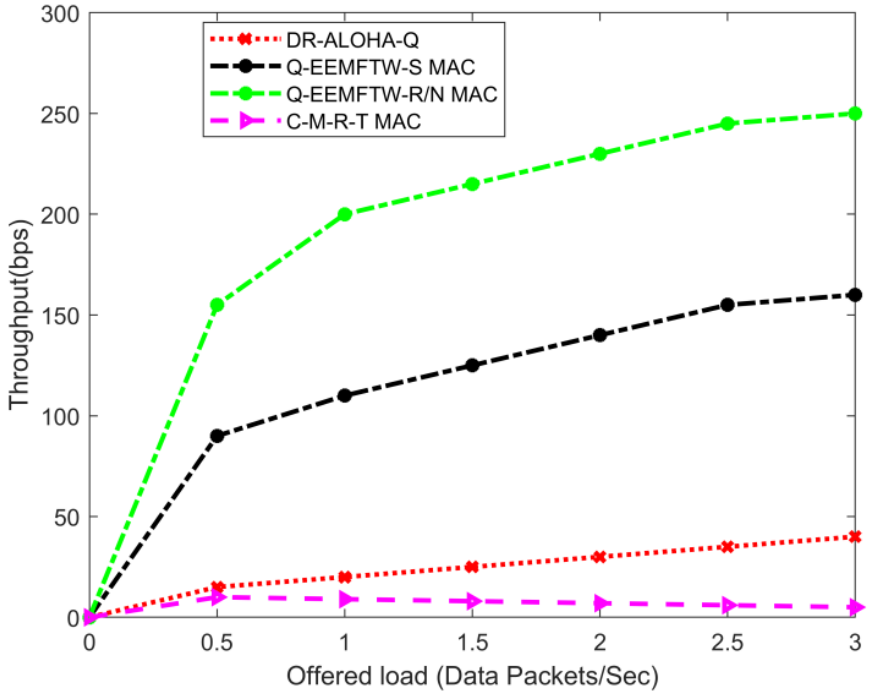


Figure 11. Throughput vs. different offered loads in the network.

Figure 11 shows that, for high traffic loads with 3 data packets per second, the proposed Q-EEMFTW-R/N and Q-EEMFTW-S can achieve different offered loads performance of approximately 69.5% and 40.2% greater in comparison with the existing MAC protocol. For low traffic loads with 1 data packet per second. In terms of throughput with high and low traffic loads, the proposed Q-EEMFTW-R/N and Q-EEMFTW-S can outperform the various supplied loads with performances that are around 47% and 22% more than those of the present MAC protocol.

As a result, a communicating node has a crucial chance to send and receive two-way data packets across many flows in less time, which is necessary. Figure 12 shows that, when the number of backward data packet is 1. The proposed Q-EEMFTW-R/N and Q-EEMFTW-S can achieve the performance increase of approximately 41.3% and 19.1% in comparison to the existing MAC protocol in terms of number of backward data packets throughput. Furthermore, when the numbers of backward data packets are increased to 6 the proposed Q-EEMFTW-R/N and Q-EEMFTW-S can achieve an increase in performance of approximately 79% and 47.4% in comparison with existing MAC protocol.

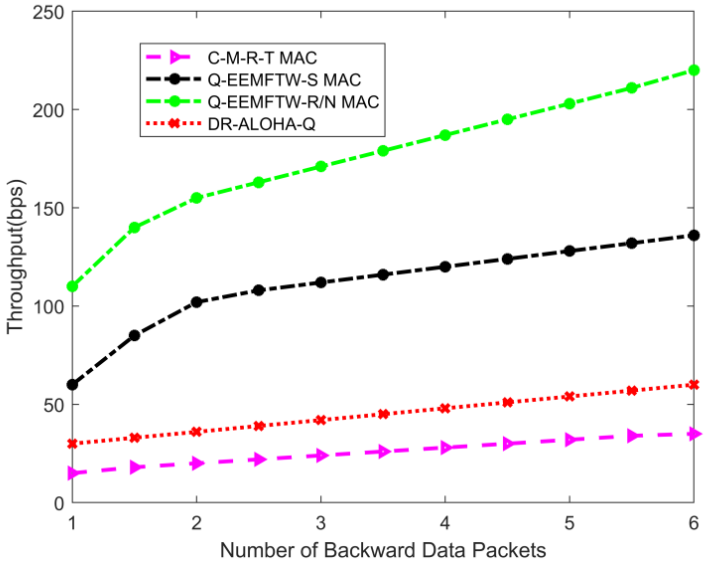


Figure 12. Throughput vs. Number of backward data packets.

Figure 13 shows the network throughput performance of the proposed MAC protocol with effects of different number of nodes, compared to the existing MAC protocol, while the data packets transmission are 0.4 data packets per second. Furthermore, when the number of nodes is 5 in the network, the proposed Q-EEMFTW-S MAC protocol achieves 6.2% greater performance, as compared to the present MAC protocol in terms of network throughput.

Moreover, when number of nodes are increased from 5 to 40, the proposed Q-EEMFTW-S MAC protocol achieves 39.4% greater performance. For Q-EEMFTW-R/N MAC protocol we achieved 75.3% greater performance, as compared to the existing MAC protocol in terms of network throughput.

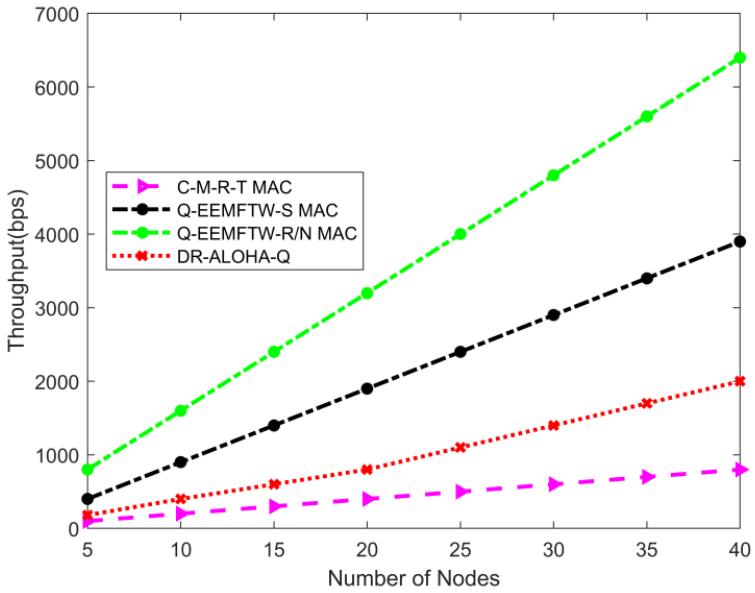


Figure 13. Throughput vs. number of nodes in the network.

For different BER values, the network data throughput is demonstrated in Figure 14 with a load of 0.8 packets per second. The results that are obtained from multiple procedures are shown in separate lines. When BER increases from 0 to 1, the performance of the network throughput decreases. It is shown that, the multi-flow Two-Way communicating data packet communication mechanism of the proposed Q-EEMFTW-MAC protocol maintains higher network throughput under increasing bit-error-rate conditions. Table 5 shows the abbreviations and parameter definitions used in this study.

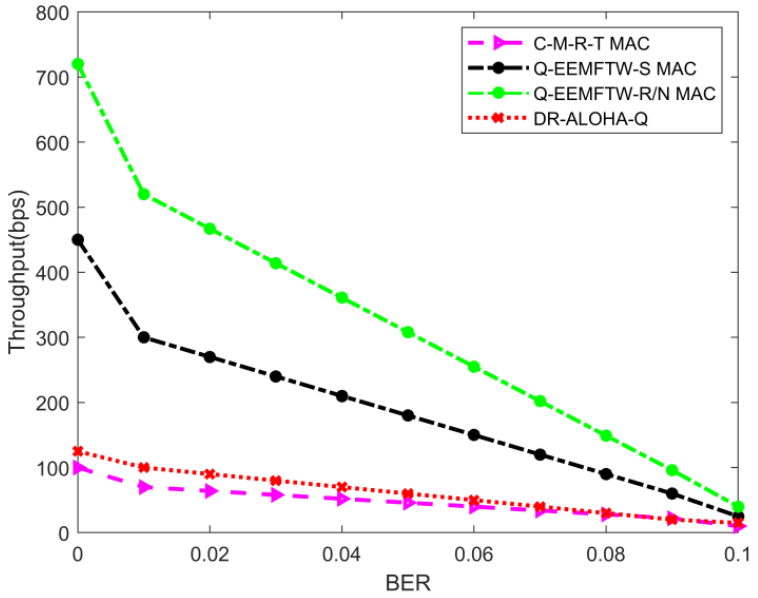


Figure 14. Throughput vs. BER.

Table 5. Notations and parameter definitions.

Symbol	Meaning
(T_{control})	Transmission duration of a control packet
(\tau_{max})	Maximum propagation delay between communicating nodes
(Time_{MFDP})	Time required for one MFDP control-packet exchange, Eq. (16)
(Time_{DATA})	Time required to transmit one data packet
(d_{data})	Data-packet transmission duration
(H)	Number of hops between source and destination
(b)	Number of backward-flow packets
(P_e)	Bit-error probability
(P_c)	Data transmission probability
(cw)	Contention-window duration
(T_{CTS})	Transmission duration of the CTS control packet
(R/T)	Total required transmission time
(P_f)	Frame-error probability
(d_s)	Data-packet size in bits
(d_{soh})	Data-packet overhead size

To transmit only one MFDP the required time will be calculated as:

$$\text{Time}_{\text{MFDP}} = 2(T_{control} + \tau_{max}) \quad (16)$$

To calculate time for data when only one data packet is transmitted:

$$\text{Time}_{\text{DATA}} = (d_{data} + \tau_{max}) \quad (17)$$

To calculate required time to transmit data packets from R/N to Tx anchor node will be:

$$Time_1 = \text{Time}_{\text{MFDP}} + \text{Time}_{\text{DATA}} \quad (18)$$

The required time to transmit data packet from communicating node which is away from Tx anchor node at multiple hopes, will be:

$$Time_2 = \text{H} \times \text{Time}_{\text{MFDP}} + \text{H} \times \text{Time}_{\text{DATA}} \quad (19)$$

Time required for n number of flows in opted MFTW-S will be:

$$Time_3 = (1 - P_e)\ l_{+loh} \times \text{n} \times Time_2 \quad (20)$$

Now, required time for backward flow will be:

$$Time_4 = (1 - P_e)\ l_{+loh} \times \text{n} \times Time_1 \quad (21)$$

The total required time R/T in MFTW-MF will be:

$$\text{R/T} = Time_3 + Time_4 \tag{22}$$

The required time for MFTW-R/N MAC to transmit data packets in backward direction will be:

$$Time_5 = (1 - P_e)\ l_{+loh} \times \text{n} \times T_{CTS} + \tau_{max} + \text{Time}_{\text{DATA}} \tag{23}$$

Furthermore, the final required time (R/T) for MFTW-R/N will be

$$\text{R/T} = Time_3 + Time_5 \times \text{H} \tag{24}$$

Here MFDP denotes the Multi-Flow Data Packet, and MFTW-S represents the standard Multi-Flow Two-Way transmission case. The relay/receiver-node-assisted Multi-Flow Two-Way case is represented by MFTW-R/N and R/N denotes a relay/receiver node. Moreover, CTS denotes Clear-to-Send, and BER denotes Bit Error Rate.

The performance of the proposed Q-EEMFTW-MAC protocol in terms of end-to-end transmission latency is also evaluated. The performance is evaluated using different observation distances, multiple amounts of hops, multiple amounts of flows, and bit error rate (BER).

The equation for total delay to transmit only one data packet is shown below

$$\text{latency_MFDP} = \text{Time_MFDP} = 2(\text{T_control} + \tau\text{_max}) \tag{25}$$

Figure 15 shows the performance comparison of the proposed MAC protocol and the existing MAC protocol in terms of effects of different number of hops, while the bit error rate is 10^{-3}. As seen in the Figure 15, the end-to-end delay increases as the number of hops increases from one to ten. However, while using the existing MAC protocol technique, the communicating data packets suffer from extended latency as the number of hops increases. Furthermore, in comparison to the current MAC protocol, the suggested Q-EEMFTW-MAC protocol requires 27.3% less data packet latency, while the number of hops in the network are five.

Furthermore, with an average of five hops, the proposed Q-EEMFTW-R/N MAC protocol achieves 36.3% lower data packet transmission delay, than the existing MAC protocol. In addition, when the number of hops is increased from five to ten, stability decreases by approximately 73%. However, the proposed technique is capable of transmitting multiple communicating data packets across multiple flows in the same time duration. Furthermore, when the time spent by the communicating nodes was increased, the proposed MAC protocol can still transmit data packet in backward flow in a comprehensive way. As a result, compared with the existing MAC protocols, the proposed MAC protocol achieves better end-to-end delay performance for different numbers of hops. Figure 16 shows the end-to-end delay of the proposed and existing MAC protocols for different distances between the Tx anchor node and the sensor Rx node, while the bit error rate is 10^{-3}. As the distance increases, the total network time and the data-packet transmission time from the Tx anchor node to the sensor Rx node also increase because of the long propagation delay in UAWSNs. Consequently, the Wait_Flow state time increases as the overall transmission time increases. The proposed protocols keep the delay lower than the existing protocols at all tested distances.

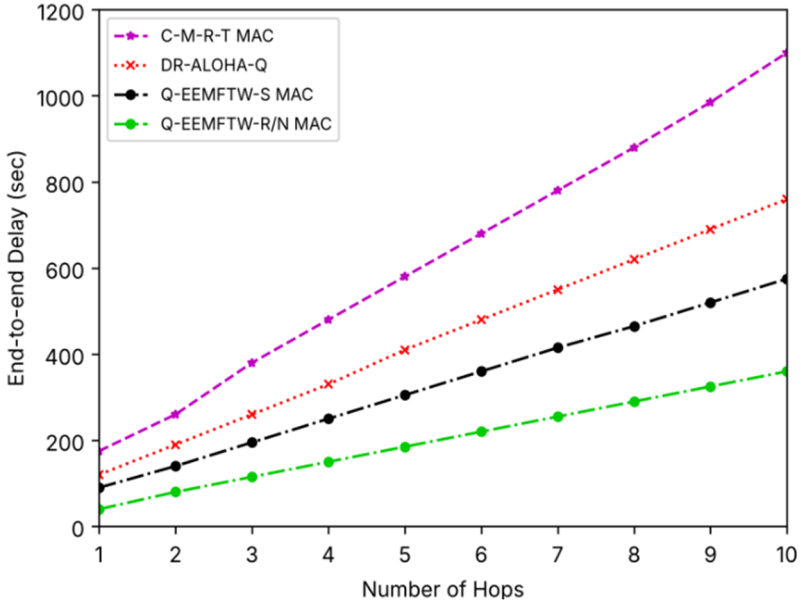


Figure 15. End-to-end delay vs. number of nodes.

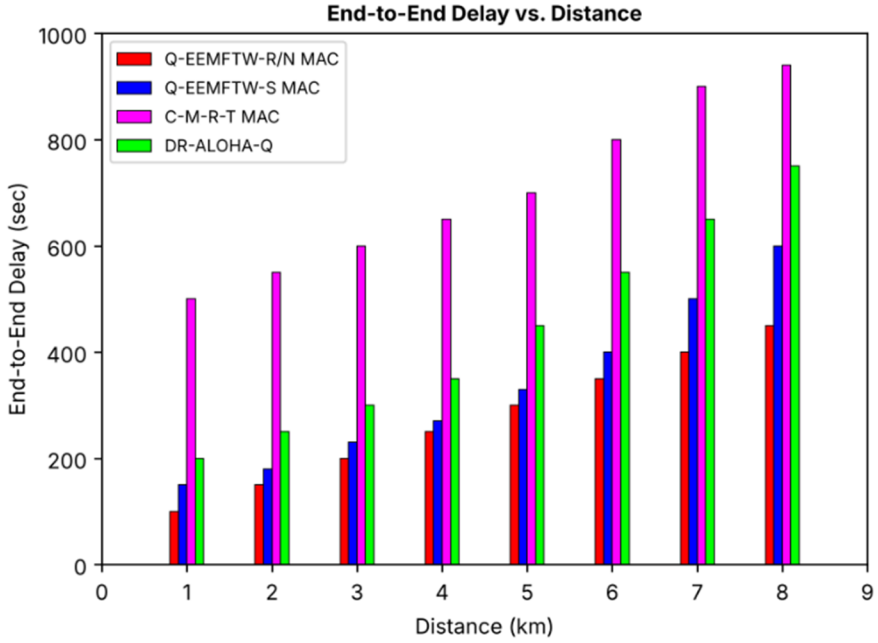


Figure 16. End-to-end delay vs. distance.

Figure 17 shows the end-to-end delay of the proposed and existing MAC protocols for different numbers of flows, while the bit error rate is 10^{-3}. The end-to-end delay increases as more communication flows occur simultaneously, because additional flows require more packet scheduling, channel access and forwarding. Nevertheless, Q-EEMFTW-S and Q-EEMFTW-R/N keep the end-to-end delay lower than CMRT-MAC and DR-ALOHA-Q for all tested numbers of flows. Q-EEMFTW-R/N achieves the lowest delay, which highlights the benefit of the proposed backward-flow and multi-flow transmission methods as the traffic increases.

Figure 18 shows that end-to-end delay increases as more communication flows occur simultaneously. This is because extra flows need more packet scheduling, channel access, and forwarding. Still, Q-EEMFTW-S and Q-EEMFTW-R/N keep end-to-end delay lower than CMRT-MAC and DR-ALOHA-Q for all tested flow numbers. Q-EEMFTW-R/N has the lowest delay, highlighting the benefit of the proposed backward-flow and multi-flow transmission methods as traffic increases.

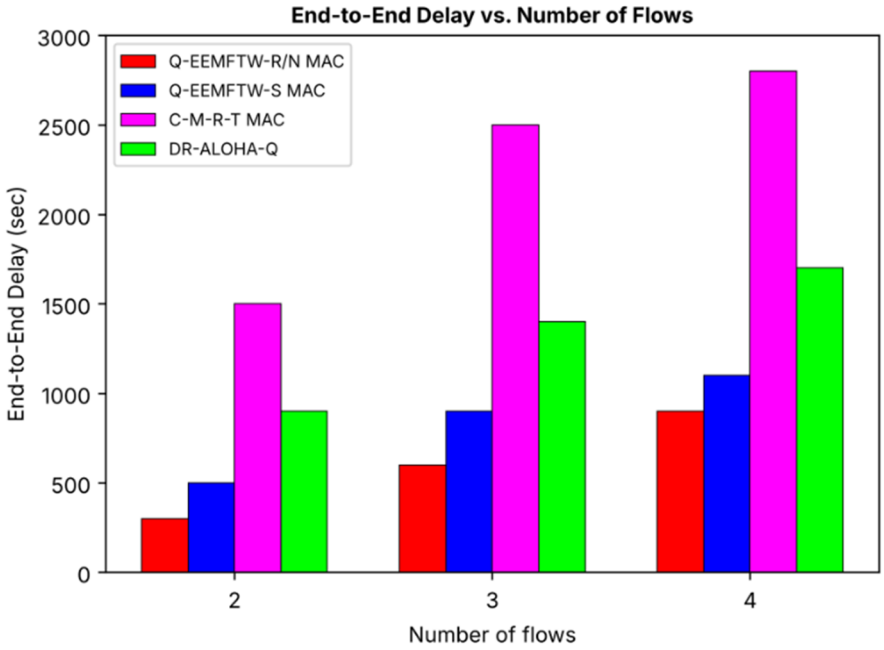


Figure 17. End-to-end delay vs. number of flows.

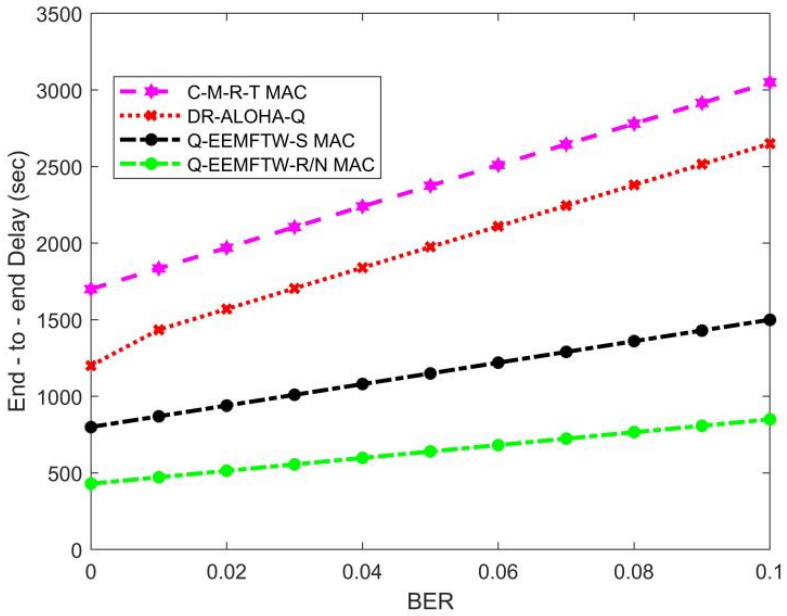


Figure 18. End-to-end delay vs. BER.

In multi-hop setup, the latency of the opted protocol to transmit data packets from Tx anchor node to H hop R/N with a probability P_c is shown below, while cw denotes the contention window duration and H is the number of hopes between the Tx anchor node and the relay/receiver node. Similarly, LatencyMFDP represents the delay associated with MFDP control exchange, and LatencyDATA represents the transmission delay of one data packet.

$$latency_1 = \text{cw} + d_{bDbD,s} + \text{H}\times \text{ latency }_{\text{MFDP}} + \text{H} \times \text{ latency }_{\text{DATA}} \quad (26)$$

The delay for n number of data packets in probability pc is:

$$\text{latency}_2 = \text{cw} + \text{d_bDbD,S} + \text{H} \times \text{latency_MFDP} + (\text{n} \times \text{H} \times \text{latency_DATA}) / \text{P_c} \quad (27)$$

In backward flow, delay of backward number of data packets b flows will be:

$$\text{latency}_3 = \text{cw} + \text{d_bDbD,R,S} + \text{H} \times \text{latency_MFDP} + (\text{b} \times \text{H} \times \text{latency_DATA}) / \text{P_c} \quad (28)$$

The overall latency of the opted MFTW-S MAC protocol will be:

$$\text{latency} = latency^2 + latency^3 \quad (29)$$

The latency of the proposed EEMFTW-R/N MAC protocol for control data packets will be

$$latency_4 = (T_{CTS} + \tau_{max}) + (s_{WY,}R_n - 2 \times T_{control} - 2\tau_{max}) \quad (30)$$

The latency of the backward EEMFTW-R/N will be:

$$latency_5 = (d_{data} + \tau_{max}) + (d_{backward,}R_n - 2 \times T_{control} - 2\tau_{max}) \quad (31)$$

The total latency will be:

$$\text{latency} = latency_2 + latency_4 + latency_5 \times \text{H} \quad (32)$$

In this section I calculated about the energy consumption based on the effects of distance between Tx anchor node and sensor Rx node, different flows, different offered loads and Packet time control. The model used to achieve the performance in terms of energy consumption is shown below.

$$P_f = 1 - (1 - P_e)\ d_s + ds_{oh} \quad (33)$$

Here, Pf denotes the probability of frame error, Pe is the bit-error probability, ds is the payload size in bits, and dsoh is the protocol-overhead size in bits. Figure 19 shows that, when data is transmitted across two different numbers of flows, the proposed MAC protocol Q-EEMFTW-S uses 19.4% less power than the existing MAC protocols. When data is transmitted across four different numbers of flows, the proposed MAC protocol Q-EEMFTW-S uses 51.2% less power than the existing MAC protocols. Moreover, when data is transmitted across two different numbers of flows, our proposed

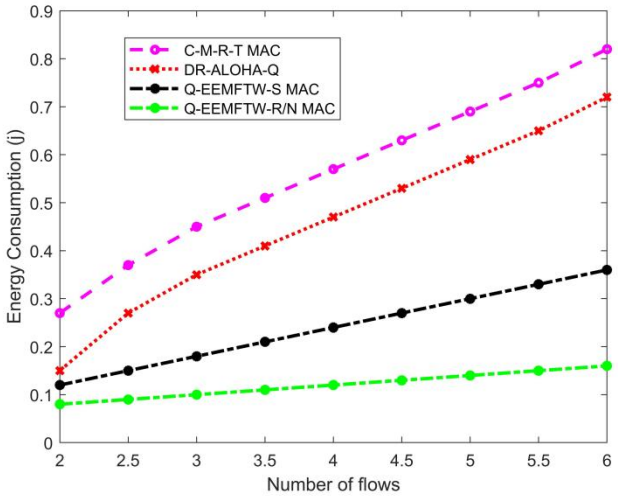


Figure 19. Energy consumption vs. number of flows.

MAC protocol Q-EEMFTW-R/N uses 24.7% less power than the existing MAC protocols. When data is transmitted across four different numbers of flows, our proposed MAC protocol Q-EEMFTW-R/N uses 79.4% less power than the existing MAC protocols.

Figure 20 shows the energy consumption of the proposed MAC protocol and the existing protocol with increase in load of the proposed architecture. When there are various data packets to transmit, with multiple number of flows, the Q-EEMFTW-MAC decreases transmission time by allowing multiple communicating nodes to transmit data at the same time. Under all traffic load conditions, the proposed MAC protocol needs very low energy than the existing MAC protocol. The proposed MAC protocol needs low control data packets during transmission of data packets. Furthermore, during transmitting data packets across both regular and backward flow, the proposed MAC protocol uses six different states. AS a result, in low traffic loads of 0.2 data packet per second, the performance of the proposed MAC protocol is round about 29.6% increased from existing MAC protocol.

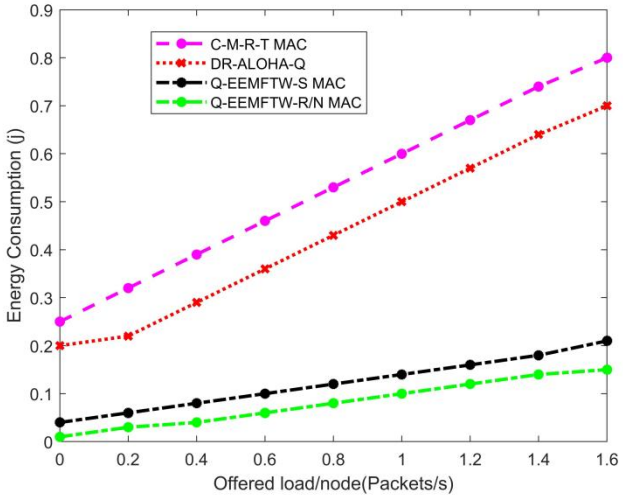


Figure 20. Energy consumption vs. different offered loads per node.

Figure 21 shows the performance of the proposed Q-EEMFTW-MAC protocol in terms of energy consumption with the effects of different number of nodes. When the number of nodes is 20 in the network, the proposed Q-EEMFTW-S MAC protocol outperforms by 38.3%, while Q-EEMFTW-R/N MAC protocol outperforms by 50.2%, as compared to the existing MAC protocol in terms of energy consumption. Furthermore, when the number of nodes is increased from 20 to 40, Q-EEMFTW-S MAC protocol outperforms by 76.2%, when compared to the existing MAC protocol, while the Q-EEMFTW-R/N MAC protocol outperforms by 88.6%. The network bit error rate P_e is calculated by dividing the total number of bit errors by the total number of transmitted bits. The frame-error probability P_f is related to P_e through Eq. (33), where d_s is the payload size and d_soh is

the protocol-overhead size of a data packet, both in bits. The probability that a frame is received without error is the complement of P_f, and is denoted by P_c:

$$P_c = 1 - P_f = (1 - P_e)^{(d_s + d_soh)} \quad (34)$$

Figure 22 shows the control-overhead ratio of the proposed and existing MAC protocols for different offered loads, while the bit error rate is 10^{-3}. The control-overhead ratio is defined as the percentage of the total transmission time that is spent on control packets (MFDP, CTS and RDP). The ratio decreases as the offered load increases for all protocols, since more data packets share the same control exchanges. The simulation results show that the proposed Q-EEMFTW-S and Q-EEMFTW-R/N protocols have a lower control-overhead ratio than CMRT-MAC and DR-ALOHA-Q at all offered loads.

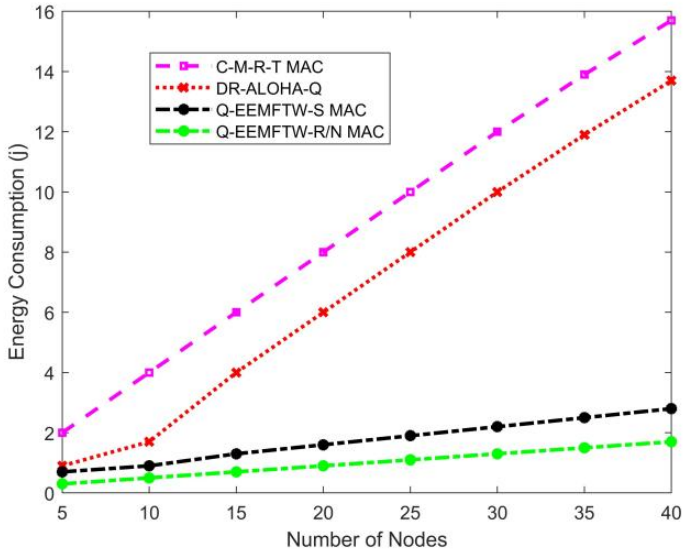


Figure 21. Energy consumption vs. number of nodes.

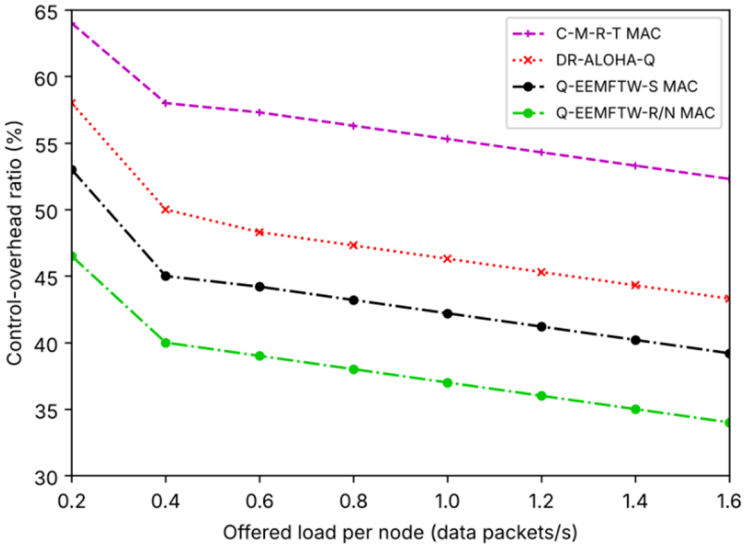


Figure 22. Control-overhead ratio (%) vs. offered load per node.

To quantify the contribution of the proposed two-way communication mechanism, Q-EEMFTW-R/N and Q-EEMFTW-S are compared under the same network and traffic conditions. Q-EEMFTW-S supports normal forward multi-flow transmission. Q-EEMFTW-R/N supports backward packet transmission through relay/receiver nodes. The comparison shows that the two-way mechanism increases the channel utilization by utilizing the available backward transmission opportunities instead of a dedicated communication cycle. Therefore, fewer additional channel-reservation operations are needed. This can reduce the control overhead and energy consumption and improve the throughput and end-to-end delay. Table 6 summarizes the quantitative comparison.

Table 6. Quantitative Comparisons of Two-Way Communication mechanism.

Performance Parameters	Q-EEMFTW-S (Forward Only)	Q-EEMFTW-R/N (Two-Way)	Improvement in Two-Way
Throughput, normalized index	145.5	183.7	26.3% higher
End-to-end delay, normalized index	71.6	64.1	10.5% lower
Energy consumption, normalized index	47.4	19.7	58.4% lower
Control-overhead ratio	≈38	≈33	≈13.1% lower

The above table illustrates the contribution of the two-way transmission mechanism. Compared with the standard forward-flow Q-EEMFTW-S configuration, Q-EEMFTW-R/N provides higher normalized throughput, lower end-to-end delay, lower normalized energy consumption, and lower control overhead under the selected operating conditions. Moreover, the scalability of the Q-EEMFTW-MAC was investigated with various numbers of sensor nodes and a number of traffic flows at the same time. The results show that the protocol preserves its performance advantage as the network size and simultaneous traffic flows. This shows the robustness and scalability of Q-EEMFTW-MAC in highly network and communication environments.

A sensitivity analysis was conducted to examine the effect of the key Q-learning parameters on the Q-EEMFTW-MAC performance. The learning rate affects the pace of Q-value updates, the discount factor affects the importance of future rewards, and the exploration probability affects the trade-off between exploration and exploitation. Each parameter was modified, holding all other simulation settings constant. The obtained performance variations were evaluated in terms of throughput, end-to-end delay, and energy consumption.

5. Conclusions

The Underwater Acoustic Wireless Sensor Network (UAWSN) have presented a unique set of challenges that have been addressed with notable progress in the design and modelling of a MAC protocol based on Q-learning. In this article, I propose a Q-EEMFTW-MAC protocol for UAWSNs. The main objective of our opted MAC protocol is to increase the network throughput, reduce end-to-end delay of the network and reduce the energy consumption in UAWSNs. Through the incorporation of reinforcement learning concepts, particularly Q-learning, our protocol has successfully optimized energy consumption, hence augmenting the overall performance and lifespan of UAWSNs. Our opted Q-EEMFTW-MAC protocol achieves better performance in terms of network throughput, end-to-end delay of the network and energy efficiency, when compared with existing CMRT-MAC and DR-ALOHA-Q protocols for UAWSNs. The proposed approach achieves up to 84.3% higher network throughput and up to 88.6% lower energy consumption. In the five-hop latency evaluation, Q-EEMFTW-MAC reduces end-to-end delay by 27.3%, while the Q-EEMFTW-R/N configuration achieves a reduction of 36.3%. These improvements demonstrate the effectiveness of multi-flow two-way communication and adaptive Q-learning-based MAC operation in UAWSNs. For future work I will carry out the implementation of a number of more recent Q-learning-based underwater MAC protocols within the same simulation environment and assess them under the same topology, traffic, channel, energy, and learning conditions [21,22].

Funding: The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (2026/03/44457).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study will be available on request from the corresponding author.

Acknowledgments: Thanks to Prince Sattam bin Abdulaziz University for supporting this work.

Conflicts of Interest: The author declares no conflict of interest.

Use of AI and AI-Assisted Technologies: During the preparation of this work, the author used Grammarly and Quilt bolt to improve the English writing.

References

1. Khan, Z.; Gang, Q.; Muhammad, A.; et al. RETRACTED: A Comprehensive Survey of Energy-Efficient MAC and Routing Protocols for Underwater Wireless Sensor Networks. *Electronics* **2022**, *11*, 3015. https://doi.org/10.3390/electronics11193015.
2. Khan, Z.U.; Aman, M.; Rahman, W.U.; et al. Machine Learning-based Multi-path Reliable and Energy-efficient Routing Protocol for Underwater Wireless Sensor Networks. In Proceedings of the 2023 International Conference on Frontiers of

Information Technology (FIT), Islamabad, Pakistan, 11–12 December 2023; pp. 316–321. https://doi.org/10.1109/fit60620.2023.00064.

3. Narla, V.L.; Kachhoria, R.; Arun, M.; et al. IoT based energy efficient multipath power control for underwater sensor network. *Int. J. Syst. Assur. Eng. Manag.* **2022**, 1–10.
4. Mian, S.; Gang, Q.; Aman, M.; et al. A Visible Light Geometric Multilateration Based Positioning System to Localize Nodes in an UWSNs. In Proceedings of the 2023 IEEE 11th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 21–22 October 2023; pp. 392–396. https://doi.org/10.1109/iccsnt58790.2023.10334585.
5. Alfouzan, F.A. Energy-Efficient Collision Avoidance MAC Protocols for Underwater Sensor Networks: Survey and Challenges. *J. Mar. Sci. Eng.* **2021**, *9*, 741. https://doi.org/10.3390/jmse9070741.
6. Wu, Y.; Geng, X. Overview of the MAC Protocols for Underwater Acoustic Networks. *World Sci. Res. J.* **2021**, *7*, 33–45.
7. Aman, M.; Gang, Q.; Mian, S.; et al. RSSI based Trilateration Technique to Localize Nodes in Underwater Wireless Sensor Networks through Optical Communication. In Proceedings of the 2021 16th International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 22–23 December 2021; pp. 1–5. https://doi.org/10.1109/icet54505.2021.9689902.
8. Fattah, S.; Gani, A.; Ahmedy, I.; et al. A Survey on Underwater Wireless Sensor Networks: Requirements, Taxonomy, Recent Advances, and Open Research Challenges. *Sensors* **2020**, *20*, 5393. https://doi.org/10.3390/s20185393.
9. Chen, K.; Ma, M.; Cheng, E.; et al. A Survey on MAC Protocols for Underwater Wireless Sensor Networks. *IEEE Commun. Surv. Tutorials* **2014**, *16*, 1433–1447. https://doi.org/10.1109/surv.2014.013014.00032.
10. Heidemann, J.; Ye, W.; Wills, J.; et al. Research challenges and applications for underwater sensor networking. In Proceedings of the IEEE Wireless Communications and Networking Conference, 2006, Las Vegas, NV, USA, 3–6 April 2006; pp. 228–235. https://doi.org/10.1109/wcnc.2006.1683469.
11. Chirdchoo, N.; Soh, W.S.; Chua, K.C. RIPT: A Receiver-Initiated Reservation-Based Protocol for Underwater Acoustic Networks. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 1744–1753. https://doi.org/10.1109/jsac.2008.081213.
12. Ng, H.H.; Soh, W.S.; Motani, M. An underwater acoustic MAC protocol using reverse opportunistic packet appending. *Comput. Netw.* **2013**, *57*, 2733–2751. https://doi.org/10.1016/j.comnet.2013.06.007.
13. Lin, W.; Chen, K. MHM: A Multiple Handshaking MAC Protocol for Underwater Acoustic Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 9798075. https://doi.org/10.1155/2016/9798075.
14. Lee, J.W.; Cho, H.S. A Hybrid Sender- and Receiver-Initiated Protocol Scheme in Underwater Acoustic Sensor Networks. *Sensors* **2015**, *15*, 28052–28069. https://doi.org/10.3390/s151128052.
15. Gazi, F.; Ahmed, N.; Misra, S.; et al. Reinforcement Learning-Based MAC Protocol for Underwater Multimedia Sensor Networks. *ACM Trans. Sens. Netw.* **2022**, *18*, 1–25. https://doi.org/10.1145/3484201.
16. Alhassan, I.B.; Mitchell, P.D. Packet Flow Based Reinforcement Learning MAC Protocol for Underwater Acoustic Sensor Networks. *Sensors* **2021**, *21*, 2284. https://doi.org/10.3390/s21072284.
17. Jin, L.; Huang, D.D. A slotted CSMA based reinforcement learning approach for extending the lifetime of underwater acoustic wireless sensor networks. *Comput. Commun.* **2013**, *36*, 1094–1099. https://doi.org/10.1016/j.comcom.2012.10.007.
18. Molins, M.; Stojanovic, M. Slotted FAMA: A MAC protocol for underwater acoustic networks. In Proceedings of the OCEANS 2006—Asia Pacific, Singapore, 16–19 May 2007; pp. 1–7. https://doi.org/10.1109/oceansap.2006.4393832.
19. Ng, H.H.; Soh, W.S.; Motani, M. A Bidirectional-Concurrent MAC Protocol With Packet Bursting for Underwater Acoustic Networks. *IEEE J. Ocean. Eng.* **2013**, *38*, 547–565. https://doi.org/10.1109/joe.2012.2227553.
20. Lee, J.W.; Cho, H.S. Cascading Multi-Hop Reservation and Transmission in Underwater Acoustic Sensor Networks. *Sensors* **2014**, *14*, 18390–18409. https://doi.org/10.3390/s141018390.
21. Gang, Q.; Rahman, W.U.; Zhou, F.; et al. A Q-Learning-Based Approach to Design an Energy-Efficient MAC Protocol for UWSNs Through Collision Avoidance. *Electronics* **2024**, *13*, 4388. https://doi.org/10.3390/electronics13224388.
22. Zuberi, H.H.; Liu, S.; Bilal, M.; et al. Quadrature phase shift keying Sine chirp spread Spectrum under-water acoustic communication based on VTRM. In Proceedings of the 2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 16–20 August 2022; pp. 884–888. https://doi.org/10.1109/ibcast54850.2022.9990102.