ACTA SCIENTIFIC COMPUTER SCIENCES
Volume 6 Issue 6 June 2024
Research Article

Enhancing Data Collection Efficiency in Underwater Wireless Sensor Networks: A Contention-Free Pipelined Scheduling (CFPS) MAC Protocol

Amwine Louis[1], Abdul Rehman[2] and Ganesh Kumar[3]*

[1]*Faculty of Computing and Informatics, Department of Computer Science, Uganda*
[2]*School of Computer Science and Engineering, Kyungpook National University, South Korea*
[3]*Department of Computer Science and Engineering, College of Engineering Guindy, Anna University Chennai, India*

***Corresponding Author:** Ganesh Kumar, Department of Computer Science and Engineering, College of Engineering Guindy, Anna University Chennai, India.

Received: May 08, 2024
Published: May 14, 2024

Abstract

The inherent features of Underwater Wireless Sensor Networks (UWSNs) such as a high latency, limited band width, high bit error and a long propagation delay associated with underwater acoustic communication enable multiple transmitters at varying distances from the receiver to transmit simultaneously. Many of the existing Medium Access Control (MAC) protocols available utilize these features, most especially the high latency feature to reduce data collection time. However, most of the MAC protocols utilise contention-based mechanisms and this causes transmission collisions as the number of nodes increase. The transmission collisions (control and data packet collisions) at the MAC layer, lead to energy waste. It is infeasible to replace the node battery in the UWSNs. Packet collisions need to be avoided at the MAC layer so as to reduce node energy waste, thereby improving the throughput and fairness of the network. In order to mitigate these challenges, we propose a Contention-Free Pipelined Scheduling (CFPS) MAC protocol based on Energy-efficient Duty cycling that reduces node energy waste and data collection time in a single-hop network. The core idea of CFPS involves developing a collision-free transmission scheduling table that the receiver node uses to collect data packets during its awake time. The CFPS protocol is expected to reduce data collection time and improve network throughput when compared to the well-designed data collection protocols like, RI-MAC, DAP-MAC, RHNE-MAC, and NF-TDMA. Extensive simulation results indicate that the CFPS protocol outperform the existing solutions on various network parameters like latency reduction and node energy conservation.

Keywords: Underwater Wireless Sensor Networks (UWSNs); Media Access Control (MAC); Receiver-Initiated; Duty Cycling; Contention Free Pipeline Scheduling

Introduction

In the recent years, Underwater Wireless Sensor Networks (UWSN) have attracted research from both academia and industry due to their wide range of applications [1-4]. The significant attention given to marine environmental surveys is due to the fact that seventy percent of the earth's surface is covered by oceans. UWSNs have been utilized in applications such as surveillance, intrusion detection, military applications, oil and gas exploration, multimedia, pollution monitoring, etc. Underwater Wireless acoustic Networks consist of multiple, low-power, low-cost, multi-functional sensor nodes that are deployed to collect and provide sensing information over a selected area of interest using underwater acoustic links, as illustrated in Figure 1. Due to attenuation, the acoustic (sound) communication wave is a better choice for UWSNs as compared to radio or light waves. In spite of the technological advances in acoustic communications, the acoustic channel's salient features, like time-dependent propagation, high energy consumption, variable speed of sound, and limited bandwidth, pose delay limitations that need to be addressed [5-7]. The limited bandwidth is further constrained by factors such as path loss, noise, high delay variance, multi-path propagation, and Doppler spread [8].

Research has been carried out to conserve the energy of the nodes; this involves limiting or reducing energy waste. Key among them are idle listening to the channel, overhearing, control packet overhead, and packet collisions.

Underwater wireless sensor nodes rely on finite battery power and can't be easily recharged. Research has been carried out to conserve the node energy, this is done by reducing and controlling var-

Citation: Ganesh Kumar., *et al.* "Enhancing Data Collection Efficiency in Underwater Wireless Sensor Networks: A Contention-Free Pipelined Scheduling (CFPS) MAC Protocol". *Acta Scientific Computer Sciences* 6.6 (2024): 20-28.

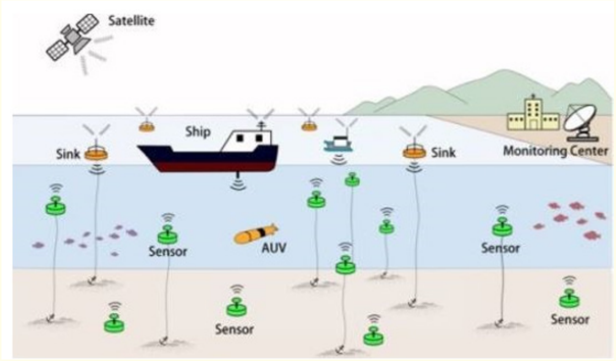

Figure 1: UWSN architecture.

ious sources of energy consumption and waste. Sources of energy consumption and waste include but are not limited to idle listening, overhearing, collisions (that require packet re-transmission), and control packet overhead. Collisions happen when multiple nodes transmit their packets simultaneously, re-transmission of the collided packets consumes extra energy. Idle listening [9] occurs when a node turns its radio on for the purpose of receiving potential packets even though no traffic has been sent. In overhearing, a node receives packets destined for other nodes. Lastly control information adds to the payload, so it is advantageous to have fewer control packets that are used in data transmission. Due to the UWSN channel characteristics coupled with the previous sources of node energy waste, Medium Access Control (MAC) protocols that utilize duty cycle scheduling designed for the terrestrial network can't be directly utilized in the underwater environment.

In Wireless Sensor Networks, Media Access Control (MAC) is core, and it has attracted increasing attention due to its potentially crucial impact on the overall performance of the network. A MAC protocol manages nodes' access to the shared common broadcast channel. The major task of MAC is to prevent packet collisions when more than two transmissions occur or resolve packet collisions and also guarantee energy efficiency, low channel access delays, and fairness among the nodes in a network [9,10].

MAC protocols that employ duty cycles have been proposed. The advantage accrued from duty cycling is that it conserves node energy by alternating between sleep and awake states [11]. This technique enables nodes to operate in a low-duty cycle mode, whereby the sensor nodes are able to switch to a sleeping state and periodically wake up for packet or data reception and transmission. This working mechanism ensures that node energy is conserved, and this, in turn, extends the working life of the node. Generally, Media Access Control protocols that utilize duty cycle working mechanisms are classified into two categories, i.e., synchronous and asynchronous. Synchronous duty cycle MAC protocols, like slotted ALOHA (S-ALOHA) [12], ALOHA with carrier sense (ALOHA-CS) [13], UWAN-MAC [14] and tone-Lohi [15] permit nodes to synchronize their wake-up and sleep schedules. Therefore, a node that has data to transmit knows when its intended receiver will be awake for data reception. However, this approach employs multi-hop synchronization that introduces the exchange of multiple packets, consequently leading to higher overhead. The overhead caused by packet exchange exceeds the expected benefit. Additionally, having a fixed sleep and listening schedule is inefficient for handling traffic with varying rates [16]. On the contrary, asynchronous duty-cycling Media Access Control protocols do not require multi-hop synchronization and employ dynamic duty cycle schedules. Depending on which end triggers communication, existing Asynchronous duty cycle protocols are roughly categorized into sender-initiated, e.g., B-MAC [17], X-MAC [18], WiseMAC [19] or receiver-initiated, e.g., RI-MAC [20].

For techniques/protocols that utilize the sender-initiation mechanism, communication is triggered by the node that intends to communicate or send data to the receiver. WiseMAC fixes node wake-up schedules, thereby enabling the sending node to deduce future receiver node wake-up times and send a short wake-up preamble just before the receiver node wakes up. Whereas in receiver-initiated protocols, the receiver node transmits its beacon to notifying the sender nodes that it's ready for packet reception. Due to the inherent features of Underwater Water Sensor Networks, receiver-initiated MAC protocols usually offer better performance as compared to other solutions since they fully utilize the bandwidth of the underwater channel [21]. In contrast, receiver-initiated approaches incur collisions in scenarios where more than one sender node transmits their packets simultaneously. Moreover, the sender nodes involved in the collision have to re-transmit their packets, and this adversely affects the duty cycle of nodes as well as the transmission delay, throughput, etc. If the duty cycle of the node is high, the energy consumption goes high and node efficiency reduces. This ultimately shortens the lifespan of the entire network.

In this paper, we present CFPS-MAC. In CFPS-MAC, a node that intends to transmit data (sender), wakes up and broadcasts a beacon. We have named this beacon a ”Hello Packet (HP)”. This beacon is broadcast while the while receiver node is switched to sleep mode. After a specific waiting time that we have termed as ”Reply Waiting Time (RWT)” if the sending node doesn't hear/get any response to its HP, it takes on the role of Responsible wake-up Node (RWN) and switches to the listening mode. The function of the Responsible Wake-up Node is to build, update, and keep a contention-free transmission scheduling table of the sender nodes' next wakeup time. In the mean time, another sender node that intends to send its data also wakes up, and broadcasts its beacon, ”Hello Packet”. The node that woke up first, the Responsible Wakeup Node that switched to listening mode, hears and receives this beacon. The RWN then compares the level of residual energy indicated in the Hello Packet that it received to its own residual energy. After this comparison, the RWN will send reply a reply. We have termed this as ”Hello Packet Reply”.

The Hello Packet Reply, among other fields, indicates a node's next wake-up time. It's this next-up time that is used to build the contention-free transmission scheduling table. The node that has more residual energy assumes the role of Responsible Wake-up Node (RWN), while the node that has less residual energy switches to sleep mode and will subsequently wake up in its next wake-up time for the purpose of transmitting its data. Meanwhile the receiving node ”receiver node” wakes up. The receiver broadcasts its Hi Packet (HiP). The Responsible Wake up Node that is listening to the channel hears and receives the receiver nodes Hi Packet(HiP). The RWN spontaneously transmits its data packet piggybacked with the Schedule Table. The receiving node, ” receiver node,” will then collect the data from the remaining sending nodes, ”sender nodes,” based on the wake-up time information that is included in the Schedule Table. CFPS-MAC, therefore, addresses the challenge of node energy waste that arises due to idle channel listening by enabling the sender nodes to power on and take turns listening to the channel depending on the amount of node residual energy.

In addition, CFPS-MAC addresses data packet collision by carefully scheduling the sender node's data transmissions based on their wake-up times. As long as the contention free transmission scheduling table is correctly received by the receiver node, there will be no packet collision since every sender node knows its wake-up time for data transmission. Finally, CFPS-MAC scales well in dense and large networks due to less message overhead. In CFPS-MAC, as long as the sender node's next wakeup time is correctly received by the Responsible Wakeup Node (RWN), it will be able to transmit its data packet without any collisions when the receiver node wakes up.

The contribution of this paper is as follows;

- The proposed CFPS-MAC adopts a receiver-initiated approach, which is advantageous in two ways. First, it reduces the time the receiver none is awake thereby reducing idle listening time and overhearing. Secondly, as a result of reduced awake time, CFPS-MAC improves node energy efficiency.
- We present an efficient data transmission scheduling mechanism that avoids packet collisions. The proposed CFPS-MAC avoids packet collisions by carefully scheduling data packet transmissions. This matches well with the primary target of our work. That is to avoid packet collisions, maximize energy efficiency, and improve network throughput.
- We carry out a performance comparison of the proposed CFPS-MAC with previously proposed solutions, for example, ES-MAC and RI-MAC, and present the results based on NS2 simulations to evaluate the performance of CFPS-MAC. CFPS-MAC outperforms previously proposed schemes like ES-MAC and RI-MAC. Simulation results confirm that our work scores high on parameters like throughput and energy efficiency while also reducing packet delay and the number of packet collisions.

The remainder of this paper is structured as follows. A review of state of the art related works on MAC is presented in Section II. Our proposed solution CFPS-MAC protocol is described in detail in Section III. CFPS-MAC performance evaluation is presented in Section IV. Finally, a conclusion is presented in Section V.

Related works

In the USNWs, multiple nodes contend to share the broadcast channel. This should be done efficiently and with fairness, thus designing a MAC mechanism is crucial to the working mechanism of Underwater Wirelesses Sensor Networks. With a need to achieve energy efficiency and a high throughput, Media Access control protocols that seek to solve the challenges associated with the long propagation delay have been considered and are ideal [22-24]. Recently, emphasis and attention have been devoted to analyzing and designing suitable solutions that employ the receiver-initiated working mechanism that can access the Underwater Medium. These solutions can be broadly categorized into two types. That is sender-initiated and receiver-initiated MAC protocols. Starting with the Sender-initiated MAC schemes, the nodes that have an intention of sending data agree on when to start the communication for data transmission purposes. For instance, in [25-28], the authors do propose and advance several sender-initiated protocols that attempt to mitigate packet collisions amongst handshaking, propagation delay tolerant, collision avoidance, and achieve high channel utilization. However, the biggest drawback that the majority of these solutions face is that they are susceptible to high overhead (exchange of too many control packets) and energy consumption. We have expanded our previous research [29] by introducing a well-defined problem statement and optimizing the nearest neighbor forwarding schedule.

In Contrast, duty cycle receiver-initiated MAC protocols, which are a central focus of this research, reduce the high overhead (packet payload) and energy consumption by enabling the receiving node to initiate handshaking and data communication. Furthermore, Duty cycle MAC protocols can be classified according to synchronization requirements. These are Synchronous and Asynchronous Duty cycle Media Access Control protocols. Asynchronous duty cycle solutions are advantageous due to the fact that they enable full bandwidth channel utilization of the underwater acoustic channel and are usually preferred [30]. In spite of that, as earlier mentioned, the challenge of idle listening adversely affects the performance of the receiver-initiated protocols. Idle listening leads node energy waste and therefore, different duty cycle solutions have been advanced and these mechanisms seek to conserve node energy.

Similarly, in the synchronous duty cycle MAC categorization, the authors in [31] propose a variant of traditional TDMA protocol that seeks to achieve a higher channel utilization as opposed to the maximum utilization of contemporary TDMA approaches. An alternative solution, ST-MAC [32], employs the use of a Spatial Conflict Graph (STCG) to avoid delays that are caused by the exposed node problem. In STUMP [33], the authors designed another TDMA that seeks to take advantage of the propagation delay information to prioritize the conflicting packet transmissions. TDMA scheduling constraints and distributed and centralized algorithms are introduced to solve the scheduling problem. The order of transmission among conflicting nodes is then ascertained. The BellmanFord algorithm is then employed to resolve the scheduling conflict, that's after ascertaining the transmission order. And in [34], the same authors advance their solution by incorporating routing capabilities. Centralized management and lack of fault tolerance are limitations of TDMA in underwater sensor networks. While distributed scheduling algorithms address this issue, they introduce additional network traffic overhead [35].

The authors in [36] proposed RIPT a technique that enables two or more sender nodes to communicate with a receiver using only one handshake and in [37] the nodes that intend to send data packets to the receiver node reserve the channel first in the channel reservation phase there after which an order list is generated. Once the receiver creates a transmission order based on reserved channels, data packets are sent according to this list. This approach leverages propagation delay information to schedule control packets efficiently, minimizing collisions and maximizing channel utilization.

CFPS-MAC aims to improve channel utilization, network throughput, and network lifetime by employing a receiver initiated handshake like some of the aforementioned MAC protocols. Sending nodes employ a dynamic duty cycle and manage their wakeup times based on residual energy levels. They then form a contention-free transmission scheduling table that the receiver bases on to collect the data packets, therefore reducing overhead packet load and alleviating packet collisions.

Receiver-initiated mac based on energy-efficient duty cycling

The design of the proposed state-of-the-art CFPS-MAC is presented in this section. Section III A presents the initial phase. An illustration of the data transmission phase (The receiving node is awake) is presented in Section III B. Section III C handles the collision cases, and Section III D illustrates an unscheduled sender waking up in the data transmission phase.

Initial phase (The receiver is in sleep mode)

The initial phase starts when a sender node wakes up asynchronously to find if the receiver node is awake with the intention to forward its data. Every sender node broadcasts a Hello Packet (HP) when it wakes up. The HP fields are; wakeup time, node ID, Receiver ID and Residual Energy. After broadcasting HP, the sender goes into listening mode while it awaits for a Hello Packet Reply (HPR) for a defined amount of time named Reply Waiting Time (RWT). If Reply Waiting Time expires and the sender node receives no Hello Packet Reply, it will assume the role of Responsible Wakeup Node.

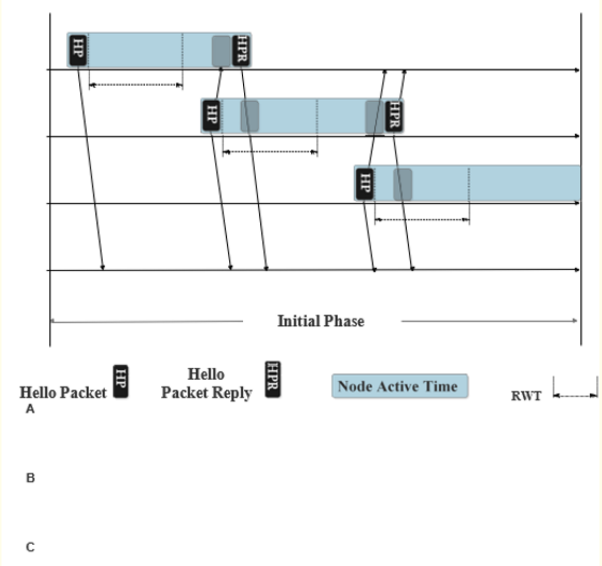

Figure 2: CFPS-MAC Initial Phase Timing Diagram.

Hello Packet Reply fields are; Node ID, Receiver ID, Residual Energy level, Next wake-up Time and a Flag field. Next wakeup time indicates the time the node will wake up next to send its data packet and its this time that is used to build and maintain the Schedule Table. The Flag field indicates if a node is the Responsible Wake-up Node or not. Reply Waiting.

Time calculation is shown below;

$RWT = 2* HP_{Tx} + 2* HPR_{Tx} + 2* MPD + 2* gt$ Where; MPD is for the Maximum Propagation Delay, gt is for guard time, HP_{Tx}, is for Hello Packet transmission, HPR_{Tx} is for Hello Packet Reply transmission.

In the event that the sender node hears a Hello Packet Reply before the (RWT) Reply Waiting Time expires, the sender node compares its residual energy level with the level indicated in the received Hello Packet Reply. The node that has a lower residual energy level switches to sleep mode. If the sending node (sender node) doesn't receive or hear any reply (HPR) to its Hello Packet after Reply Waiting Time (RWT) has expired, the node will take on the role of Responsible Wakeup Node (RWN) as indicated in Algorithm 1.

Algorithm 1 responsible wake-up node

```
for Sender node that wake – up do

broadcast HP and wait for RWT.
    node goes into listening mode and sense channel.
    if RWT elapse and no HPR received then
```

```
5:  Take on role of RWN.
6:  RWN goesto listening mode.
7:  second node wakesup and then follows steps1−3.
8:  if RWN receives this HP compares level of residual energy and send HPR then
9:      node with less energy goes to sleep.
10: end if
11: else
12:     node with more energy takes on role of RWN and repeat 6−10
13: end if
14: end for
```

The role of the Responsible Wake-up Node is to build, maintain, and update a contention-free transmission scheduling table that contains the sender's next wake-up times, reply to other sender nodes that wake up by sending Hello Packet Reply, and forward the contention-free transmission scheduling table to the receiver node when it wakes up. Sender nodes interchangeably take on the role of Responsible Wake-up node based on node residual energy level. The responsible Wake-up Node remains in listening mode, replies to the (hello packets) HPs of other sender nodes, and then forwards its data packet along with the contention-free transmission scheduling table to the receiver node when it wakes up.

Figure 2 depicts the initial phase of CFPS-MAC, where A, B, and C are sender nodes while R denotes the receiver (receiving node). The receiver node is in a sleep state. Sender node A wakes up independently and broadcasts its Hello Packet (HP), then it switches to the listening mode. After Reply Waiting Time (RWT) elapses, and node A, which is in listening mode, hears no Hello Packet Reply (HPR), Node A will then take on the role of Responsible Wake-up Node (RWN). Likewise, node B, which is also a sending node, wakes up and then broadcasts its Hello Packet. Responsible Wake-up (RWN) Node A receives this HP and compares its own level of residual energy with the level indicated in B's HP. Node A has less residual energy compared to node B. Node A subsequently nominates B as the responsible wake-up node by broadcasting a Hello Packet Reply that includes its next wake-up time and then A goes to sleep mode. After B receives node A's HPR, it creates a contention-free transmission scheduling table and stores A's next wake-up time in the Schedule Table.

Owing to the need to avoid packet collision among sender nodes, Responsible Wake-up Node should schedule the transmission of HPR based on the propagation delay between the sender node and itself.

Let $\tau i, j$ be the propagation delay between node i and node j, τ_{max} be the maximum propagation delay, and θ be the transmission time of the control packet. As depicted in Figure 2, the transmission of the Hello Packet Reply(HPR) is by sender nodes {A,B}. To avoid Hello Packet Reply collision, any HPR must be received by the sender node after the successful reception of a Hello Packet. Besides, in order to avoid the influence that is caused by the variation of propagation, we introduce a guard time $2\Delta\tau$ between the Hello Packet(HP) and the Hello Packet Reply Assuming that the RWN sends an HPR at $t_s HPR,_{RWN}$. The receiving time of the HPR from node i can be calculated as;

$$tr\_HPR\_i = tr\ HPR + (\ OHPR, i\text{-}1) * (\theta + 2\Delta\tau) + \Delta, i \epsilon \{A, B\}$$

Where OHPR is the transmission order of the HPR from node i.

When node B receives a Hello Packet from node A, node B checks its next wake-up time. The node that woke up first, (node A) next wake-up time is set as the parameter $t\ aw\ A = T_a wake$. Node B records and stores this wakeup information in the contention-free transmission scheduling table, and node B stays in listening mode, waiting for the next sender. Next, C wakes up and broadcasts its Hello Packet. B hears this and compares its level of residual energy with the level indicated in C's HP. B has less residual energy. B nominates C as the next Responsible Wake-up node by sending a Hello Packet Reply that now contains the updated contention free transmission scheduling table. That is both A and B's next wake-up times. Node B's next wake-up time is then calculated as follows:

$$t\ aw\ B = t\ aw\ A + 2 * T_{HPR} + T_{CCA}$$

Where T_{HPR} is the time that is required to transmit HPR and T_{CCA} is the time required by a node to perform Clear Channel Assessment (CCA) checks. In the same way, the rest of the sender nodes collect the wake-up schedule of the nodes that woke up before them (predecessor) and calculate their wakeup schedule accordingly. Node C remains in listening mode as it awaits the receiver node.

Data transmission phase (When the receiver node is awake)

The receiver broadcasts its Hi packet (HiP) on wake up. The Hi Packet fields are Sender ID (Receiver node ID), Receiver ID, Sender Residual Energy, Wake up time, and Flag. The sender node that is awake and also doubles as the Responsible Wake Node (RWN) hears this packet, checks and confirms the sender ID (Receiver node ID). Granted that the ID that is indicated in the Hi packet tallies with the receiver ID, the Responsible Wake-up Node forwards its data packet, which also contains the contention-free transmission scheduling table. The receiver node acknowledges data packet reception by sending ACK, and the sender node then switches to sleep as soon it receives the acknowledgment packet. The receiver node also switches to sleep and wakes up later to collect the data packet of the next sender node that will wake up since the contention-free transmission scheduling table contains all the next wake-up times of this particular receiver. Whenever the receiver node wakes up to receive a data packet, it first broadcasts a HiP.

Figure 3 is an illustration of the CFPS-MAC data transmission phase. The Receiver R broadcasts a Hi Packet (HiP) on its wake-

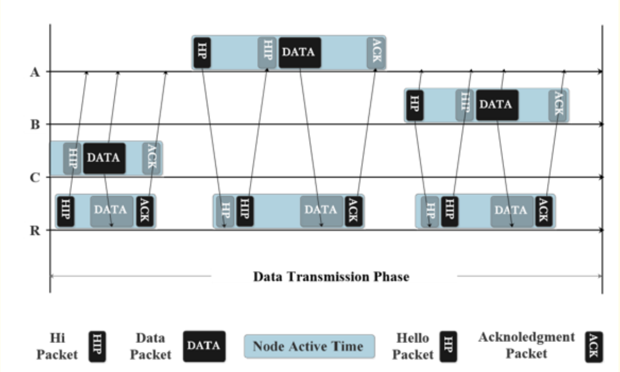

Figure 3: CFPS-MAC Data Transmission Phase Timing Diagram.

up. Sender Node C, which also doubles as the Responsible Wake-up Node, receives this HiP and checks the sender ID (Receiver node ID). If it matches the receiver ID, C sends its data packet R. The data packet C sends is piggybacked with the contention-free transmission scheduling table that has the sender's next wake-up times. The receiver node receives the data packets from the sender nodes using the sender's next wake-up time information, which is included in the Schedule Table. The receiver acknowledges data packet reception from C by transmitting the ACK packet. C switches to sleep mode immediately after it receives ACK from R. In the same way, node R switches to a sleep state due to the fact that it knows the wake-up time of the next sender, that is, node A, and also switching to a sleep state aids R in conserving its energy and reducing its duty cycle schedule. R wakes up later in time to receive sender node A's Hello Packet. R checks and confirms A's ID, and then R sends its Hi Packet. A also checks receiver R's ID, and then it sends its data packet. R acknowledges data packet reception from A by sending an ACK packet to A. Node A powers off its radio and switches to a sleep state after receiving ACK from R. In a similar approach, receiver R collects data packets from all the senders indicated in the schedule table.

In case of Hello Packet (HP) Collision

CFPS-MAC employs a dynamic independent duty cycle schedule. Therefore, there is a possibility that more than one sender node can wake up and broadcast their Hello Packet (HP) at the same time. In such a case, there would be a collision. Also, two sender nodes can broadcast their HP at different times, but due to the difference in propagation delay, their HP can collide at the RWN. When HP collision occurs at the Responsible Wake-up Node, the RWN remains silent in the listening mode because it cannot decode the information in the overlapped (collided) Hello Packets. The Hello Packets that collide at the RWN will have been received by any of the other sender nodes at different time intervals. The sender node that receives a Hello Packet compares the level of residual energy and waits for the Reply Waiting Time (RWT) to elapse. When RWT elapses, the sender node with less residual energy will nominate the sender node with more residual energy as the RWN through the Hello Packet Reply (HPR) transmission, and then it goes to sleep mode. At this point, we have two nodes that have the role of RWN. When the node that first took on the role of RWN receives this HRP, it will then also send its HPR, which contains its next wake-up time, to the newly nominated RWN and then go to sleep. In this way, the Hello Packet collision is resolved.

Figure 4 illustrates the Hello Packet collision. Node A wakes up, broadcasts its HP, and goes into listening mode. RWT elapses, and node A takes on the role of RWN. Sender B wakes up and broadcasts its HP. Likewise, node C wakes up and broadcasts its HP. Due to the difference in the propagation delay, the Hello Packets of Nodes C and B collide at node A, which is the RWN. Node A does not send HPR since it can not decode the information contained in the collided HPs. A remains silent in listening mode. Node C's HP was, however, received by Node B. Node B waits for RWT to elapse and nominates C as the RWN through an HPR since C has more residual energy, and then B goes to sleep. Node A, which is in listening mode, also receives B's HPR. A will then send its HPR to the new RWN, that is C since A has less residual energy compared to C. Node A goes to sleep.

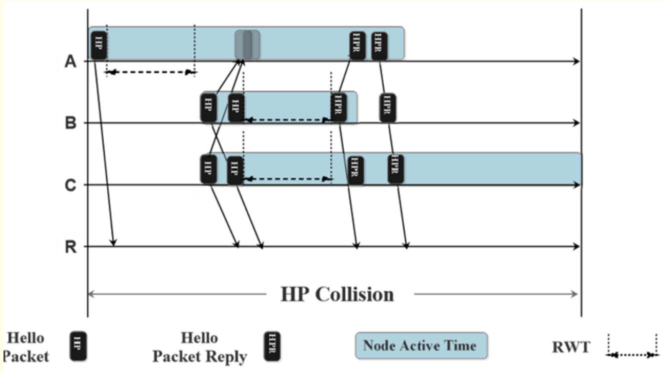

Figure 4: Hello packet collision.

Unscheduled sender waking up in the Data transmission phase

CFPS-MAC employs a dynamic duty cycle. Due to this, there is a possibility of a sender that did not participate in the initial phase waking up in the data transmission phase. Such a sender node is unscheduled because, in CFPS-MAC, the receiver node uses information contained in the contention free transmission scheduling table to collect data packets. The receiver node will have no information about a sender node that is not contained in the Schedule Table.

If an unscheduled sender node wakes up in the data transmission phase, it goes into listening mode. Using Clear Channel Assessments checks, it does not broadcast its HP because its HP will interfere with the pending data transmissions and lead to both control and data packet collision. The unscheduled sender will stay in listening mode and listen to the HiP of the receiver node. From the HiP, the unscheduled sender will know how many sender nodes have so far sent their data packets and the sender nodes that have not yet sent their data packets. This sender will also know at what times the sender nodes that haven't sent their data will wake up to send their data packets. This sender node will then go to sleep and wake up when the last scheduled sender is sending its data packet. It will then transmit its HP immediately after ACK transmission by the receiver node. When the Receiver node receives this HP, it checks the ST and creates a new entry since this sender node is not scheduled. The receiver node then sends a HiP, and the unscheduled sender sends its data packet. The receiver node then sends ACK and goes to sleep. The unscheduled sender node goes to sleep after receiving this ACK packet.

Performance Evaluation and Discussions

This section discusses the simulation analysis of the proposed CFPS-MAC. The performance evaluation of CFPSMAC is done using the NS-2 simulator. In this simulation, we deployed 8 nodes in a grid topology, and the receiver is placed in the center of the grid. The transmission range is set to 500m. We considered a star topology as depicted in Figure 5 and averaged the simulation results over 30 independent runs.

In addition, for the purpose of better understanding, the values were normalized before plotting. For comparison purposes, we modeled the extension of RI-MAC named ES-MAC [38], where the sender's schedule is maintained by only one sender. This permits the receiver node to retrieve data from all the sender nodes without any collisions. CFPS-MAC performance improvement is demonstrated with respect to the following metrics: energy consumption and duty cycle of the senders.

Figure 5 depicts the network topology, and Figure 6, 7, and 8 show the performance of the schemes in terms of energy consumed, average awake time and packet delivery delay against varying data sizes. CFPS-MAC manages the sleep and awake periods of the nodes based on the level of residual energy; thus, energy consumption is minimized with fewer duty cycles. On the other hand, nodes in ES-MAC stay in active mode till successful data delivery, therefore incurring higher energy consumption and awake periods.

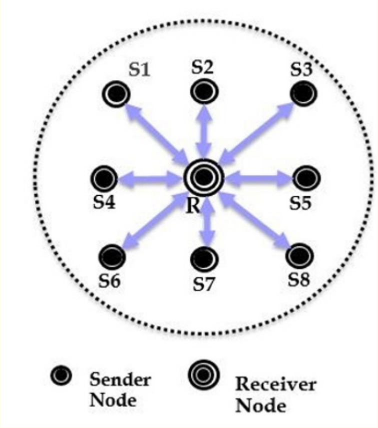

Figure 5: Network topology

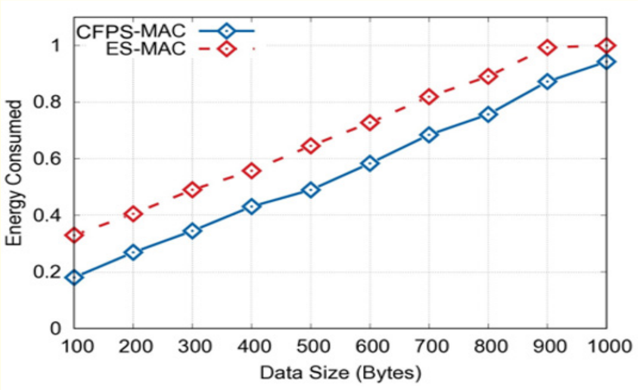

Figure 6: Energy consumed.

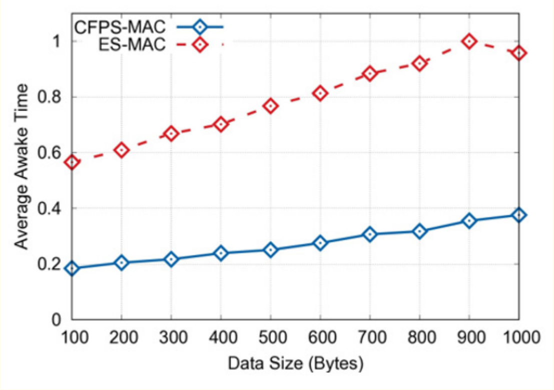

Figure 7: Energy consumed.

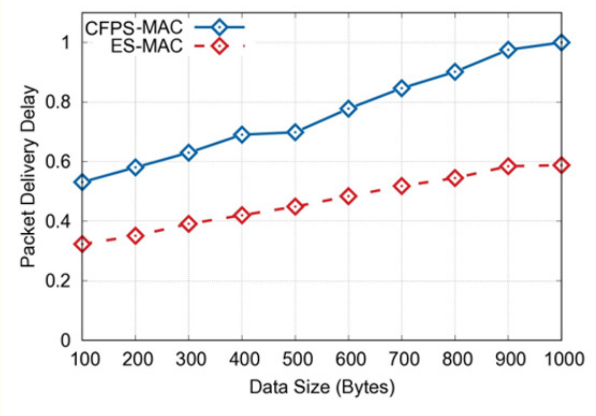

Figure 8: Packet Delivery Delay

Conclusion

In this paper, we proposed a Contention-Free Pipelined Scheduling (CFPS) MAC protocol based on Energy-efficient Duty cycling that reduces node energy waste and data collection time in a single-hop network. CFPS also reduces packet collision idle listening and improves energy efficiency and network throughput. Sender nodes in CFPS-MAC wake up before the receiver node takes turns listening to the channel based on their residual energy and then form a contention-free transmission scheduling table for their next wake-up times. When the receiver node wakes up, it uses this contention free transmission scheduling table to collect the data packets without any collisions. In addition, We have also provided a mechanism that caters to the sender nodes that are not included in the Schedule Table. Through simulation results, we observed that CFPS-MAC reduces the overhead cost packet collisions, reduces the overall duty cycle, and thus improves.

Bibliography

1. T He., *et al*. "VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance". *ACM Trans. Sensor Networks* 2.1 (2006): 1-38.
2. G Werner-Allen., *et al*. "Fidelity and Yield in a Volcano Monitoring Sensor Network". Proc. Seventh Symp. Operating Systems Design and Implementation (OSDI), (2006).
3. G Wittenburg., *et al*. "Fence Monitoring—Experimental Evaluation of a Use Case for Wireless Sensor Networks". Proc. Fourth European Conf. Wireless Sensor Networks (EWSN), (2007).
4. J Koo., *et al*. "A Tale of Two Synchronizing Clocks". Proc. Seventh ACM Conf. Embedded Networked Sensor Systems (SenSys), (2009).
5. M Garcia., *et al*. "Underwater wireless ad-hoc networks: A survey". *Mobile Ad Hoc Networks: Current Status and Future Trends* (2011): 379-411.
6. S H Ahmed., *et al*. "EENC-energy efficient nested clustering in UASN". In Proceedings of the 29th Annual ACM Symposium on Applied Computing (2014): 706-710.
7. S Rani., *et al*. "Energy efficient chain based routing protocol for underwater wireless sensor networks". *Journal of Network and Computer Applications* 92 (2017): 42-50.
8. J Heidemann., *et al*. "Research challenges and applications for underwater sensor networking". In Wireless communications and networking conference (2006).
9. W Ye., *et al*. "An energy-efficient MAC protocol for wireless sensor networks". In Proceedings of IEEE INFOCOM (2002): 1567-1576.
10. Rehman S., *et al*. "An algorithm for alleviating the effect of hotspot on throughput in wireless sensor networks". in 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops) (2017): 170-174.
11. M Buettner., *et al*. "XMAC: A short preamble MAC protocol for duty-cycled wireless sensor networks". In SenSys (2006): 307-320.
12. LFM Vieira., *et al*. "Analysis of ALOHA Protocols for Underwater Acoustic Sensor Networks". In the Proceedings of first ACM International Workshop Underwater Networks (WUWNet) (2006).
13. N Chirdchoo., *et al*. "ALOHA-based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks". In Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM), (2007): 2271-2275, Anchorage, AK, USA.
14. MK Park and V Rodoplu. "UWAN-MAC: An energy-efficient mac protocol for underwater acoustic wireless sensor networks". *IEEE Journal of Oceanic Engineering* 32 (2007): 710-720.v
15. AA Syed., *et al*. "T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks". In Proceedings of the 27th Conference on Computer Communications (INFOCOM), Phoenix, AZ, USA, 13-18 (2008): 789-797.
16. Wei Ye., *et al*. "An energy-efficient MAC protocol for wireless sensor networks". In Proceedings of IEEE INFOCOM (2002): 1567-1576.
17. J Polastre., *et al*. "low power media access for wireless sensor networks". In SenSys '04 (2004): 95-107.
18. M Buettner., *et al*. "XMAC:A short preamble MAC protocol for duty-cycled wireless sensor networks". In SenSys (2006): 307-320.
19. El-Hoiydi and Jean-Dominique Decotignie. "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks". In ALGOSENSORS (2004): 18-31.
20. Y Sun., *et al*. "RI-MAC: A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks". In SenSys (2008).
21. P Casari and M Zorzi. "Protocol design issues in underwater acoustic networks". In Computer Communications 17 (2011): 2013-2025.
22. M Stojanovic. "On the Relationship between Capacity and Distance in an Underwater Acoustic Communication Channel". In Proceedings of the 1st ACM International Workshop on Underwater Networks, Los Angeles, CA, USA (2010): 41-47.
23. F Batool., *et al*. "A Query-Based Greedy Approach for Authentic Influencer Discovery in SIoT". *Computers, Materials and Continua* 74.3 (2023).
24. X Lurton. "An Introduction to Underwater Acoustics: Principles and Applications". Springer-Praxis: London, UK (2002).
25. M Molins and M Stojanovic. "Slotted FAMA: A MAC Protocol for Underwater Acoustic Networks". In the Proceedings of the IEEE OCEANS (2006): 1-7.
26. X Guo., *et al*. "Design of a Propagation-DelayTolerant MAC Protocol for Underwater Acoustic Sensor Networks". *IEEE Journal of Oceanic Engineering* (2009): 170-180.

27. HH Ng., *et al*. "ROPA: A MAC protocol for underwater acoustic networks with reverse opportunistic packet appending". In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC) (2010): 1-6.

28. Y Noh., *et al*. "DOTS: A Propagation Delay-aware Opportunistic MAC Protocol for Underwater Sensor Networks". In Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP), Kyoto, Japan (2010): 183-192.

29. Amwine L., *et al*. "RIED-MAC: Receiver-initiated MAC based on energy-efficient duty cycling for UWSNs". In 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO) (2018): 1-5.

30. P Casari and M Zorzi. "Protocol design issues in underwater acoustic networks". In Computer Communications 34.17 (2011): 2013-2025.

31. J Yackoski and CC Shen. "UW-FLASHR: Achieving High Channel Utilization in a Time-based Acoustic MAC Protocol". In Proceedings of the 3rd ACM international workshop on Underwater Networks (2008): 59-66.

32. C Hsu., *et al*. "ST-MAC: Spatial-Temporal MAC Scheduling for Underwater Sensor Networks". in the Proceedings of 27th Conference on Computer Communications (INFOCOM), Rio de Janeiro (2009): 1827-1835.

33. K Kredo., *et al*. "STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol". in the Proceedings of 28th Conference on Computer Communications (INFOCOM), Rio de Janeiro (2009): 2961-2965.

34. K Kredo., *et al*. "Distributed Scheduling and Routing in Underwater Wireless Networks". Proceedings of the IEEE Global Telecommunications Conference GLOBECOM 2010, Miami, FL, USA, (2010): 1-6.

35. WH Liao., *et al*. "A Receiver-Initiated MAC Protocol with Packet Train Design for Underwater Acoustic Sensor Networks". in Wireless Personal Communications 82 (2015): 2155-2170.

36. WH Liao., *et al*. "A receiver-initiated MAC protocol for underwater acoustic sensor networks". in Proceedings of the 28th International Conference on Information Networking (ICOIN 14) (2014): 1-6.

37. Z Liao., *et al*. "A handshake based ordered scheduling MAC protocol for underwater acoustic local area networks". *International Journal of Distributed Sensor Networks* (2015): 15.

38. I Choi and D Kim. "ES-MAC: Energy-Driven Scheduling Scheme in Asynchronous Duty Cycle MAC Protocol for Wireless Sensor Networks". IIISC 2012, Chiamg Mai, Thailand (2012).