IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 1, 2020, accepted March 15, 2020, date of publication March 26, 2020, date of current version April 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983641

Underwater Multirobot Cooperative Intervention MAC Protocol

DIEGO CENTELLES[1], ANTONIO SORIANO[2], JOSÉ V. MARTÍ[1], AND PEDRO J. SANZ[1], (Senior Member, IEEE)

[1]Interactive and Robotic Systems Laboratory, Universitat Jaume I, 12071 Castelló de la Plana, Spain
[2]Departament d'Informàtica, Universitat de València, 46100 Burjassot, Spain

Corresponding author: Diego Centelles (centelld@uji.es)

This work was supported in part by the Spanish Government under Grant BES-2015-073112 and Grant DPI2017-86372-C3 (TWINBOT), in part by the Generalitat Valenciana under Prometeo Program, and in part by the Jaume University I under NEPTUNO Project.

ABSTRACT This work introduces a Medium Access Control (MAC) protocol designed to allow a group of underwater robots that share a wireless communication channel to effectively communicate with each other. The goal of the Underwater Multirobot Cooperative Intervention MAC (UMCI-MAC) protocol presented in this work is to minimize the end to end delay and the jitter. The access to the medium in UMCI-MAC follows a Time Division Multiple Access (TDMA) strategy which is arbitrated by a master, which also has the capability to prioritize the transmission of some nodes over the rest of the network. Two experiments have been carried out with a team of four Autonomous Underwater Vehicles (AUV) in order to compare this protocol with Aloha-CS and S-FAMA MAC protocols used in Underwater Wireless Sensor Networks (UWSN). In the first experiment, the communications and the AUVs have been simulated using UWSim-NET. The objective of this experiment was to evaluate all three protocols in terms of delay, jitter, efficiency, collisions and throughput depending on the size of the data packet and the rate of packet delivery in the application layer for each robot. The results of this experiment proved that UMCI-MAC successfully avoids packet collisions and outperforms the other two protocols in terms of delay, jitter and efficiency. The second experiment consisted of a Hardware In The Loop (HIL) teleoperation of a team of four robots. One of the AUVs was a real BlueROV in a water tank, while the remaining AUVs and the communications were simulated with UWSim-NET. It demonstrates the impact of the MAC protocols in underwater acoustic links. Of the three MAC protocols evaluated in this work, UMCI-MAC was the only one which succeeded in the proposed teleoperation experiment. Thus demonstrating its suitability as a communications protocol in underwater cooperative robotics.

INDEX TERMS Underwater communications, wireless networks, access protocols, remotely operated vehicles, telerobotics.

I. INTRODUCTION

The range of applications of Underwater Wireless Networks (UWN) has been increasing in recent years. Data collection, assisted navigation, prevention of natural disasters or maritime surveillance are among their most common applications [1]–[3]. Unlike terrestrial wireless networks, the suppression of wires in UWN applications is not straightforward. UWN are usually based on acoustic signals, since they are the only technology that enables devices under water and separated by several kilometres, to communicate with each other. However, acoustic modems suffer from high sensitivity to ambient acoustic noise; extremely limited bandwidth; large delays due to the low propagation speed of sound in water ($\approx 0.6\ ms/m$); high variability in propagation time and interference due to multi-path [4], [5]. Commercial acoustic modems typically have a transmission rate of a few kbps [6]. When high data rates are needed, as is the case of the transmission of visual information in real time, communication links based on Visible Light Communication (VLC) are also considered. But, VLC is limited to a few tens of meters in case there is low ambient light and good visibility between the transceivers [7]. Radio Frequency (RF) links are another alternative to acoustic links for UWN applications [8]. By employing large antennas and greatly reducing the transmission speed, devices like the S100 from Wireless

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

for Subsea [9] allow implementing robust links of up to 12 meters, but the strong attenuation of RF signals in water constraints the range of possible applications.

Because of the aforementioned limitations of underwater wireless communications, there are still situations in underwater robotic applications which require the use of umbilicals. In the field of cooperative robotics a team of Remotely Operated Vehicles (ROV), or Intervention AUVs (I-AUV), work together in order to fulfill a given task such as grasping or transporting objects. Some examples of these applications are maintenance tasks in the oil and the gas industries [10], underwater rescue or archaeology [11], [12]. In these applications the exchange of messages among the robots, and also with the supervisor, has to be in real time and also in a robust way. When sending commands from robot to robot, or from robot to operator, it is necessary to have as little delay as possible and a very small fluctuation of the delay (jitter) [13]. The main source of jitter in wired networks is router congestion. Transport layer protocols in terrestrial networks are therefore designed to adapt the packet injection rate in order to avoid this congestion, while trying to reduce the jitter [14], [15]. But this strategy is not appropriate for UWN because the source of the fluctuation of the delay is not only in the routing layer, but also in the access to the medium and physical layers.

The low bandwidth and the large and variable propagation time of acoustic links in UWN are the reasons why packet re-transmissions strongly contribute to introduce large delay and jitter, as is the case of Medium Access Control (MAC) protocols that do not completely avoid packet collisions. The impairment to performance caused by these factors and the high probability of errors in underwater communications justify the development of MAC protocols specific for these environments [5]. Because of the long delay in the communication, pure Carrier-Sense Multiple-Access (CSMA) approaches are not the most appropriate in scenarios where distant nodes can start to transmit at different times, because the space-time uncertainty might lead to the generation of many collisions [16].

Underwater Wireless Sensor Networks (UWSN) applications have boosted the design of protocols specific for UWN in recent years [1], [5], [6], [17]. These protocols are focused on minimizing the likelihood of collisions in acoustic channels, while trying to maximize their efficiency as it is related to the power consumption required for the communications. Despite the wide variety of strategies followed in UWN protocols, they can be categorized as contention-based or contention-free.

In contention-based protocols devices compete to access the channel. Some protocols are improvements on Aloha, such as Aloha with carrier sensing (Aloha-CS), with Collision Avoidance (Aloha-CA) or with Advanced Notification (Aloha-AN) [18]. As with the original Aloha, these protocols do not make a prior reserve of the channel, but try to avoid collisions by listening to the channel or to the messages from their neighbors. Other strategies consist of using reservation and synchronization mechanisms like the Slotted Floor Acquisition Multiple Access (S-FAMA) [19]. S-FAMA divides the time into slots whose duration depends on the maximum propagation time plus the transmission time of a control message. Two control messages are considered in S-FAMA, Request To Send (RTS) messages, which are used to ask for permission to access the channel, and Clear To Send (CTS) messages, which are used to give permission to access the channel. S-FAMA prevents the collision of data packets but does not avoid the collision of control packets. Unlike S-FAMA, the Reservation-MAC protocol (R-MAC) [20] does not require the nodes to be synchronized, instead it requires the nodes to estimate their time differences by measuring the propagation time of the messages exchanged with their peers. The Tone Lohi protocol [21] proposes a strategy which avoids the exchange of RTS and CTS messages. Instead, it divides the time in a Contention Round into a Reservation Period (RP) and a Data Period. A node interested in transmitting a message sends a pure tone during the RP. If it does not receive a tone from another node during the RP then it starts the transmission, otherwise it waits for a random number of RPs.

Contention-free is the other category of protocols. These solutions allow nodes to access the channel at the same time, or without risk of collision. They are subdivided into: Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA) and Time Division Multiple Access (TDMA). Due to the narrow bandwidth of the acoustic links, a solution based solely on FDMA is not feasible for UWSN [4]. The encoding of the message in CDMA reduces the effective transmission rate [6]. However, the combination of CDMA with contention-based methods has proven to be suitable for UWSN [22]. Finally, in TDMA protocols each node is assigned a time slot in which it has exclusive access to the channel. In very large networks the need to wait for each node's time slot leads to large delays. Therefore, TDMA protocols specific for UWSN such as Multidimensional-Scaling-MAC (MDS-MAC) [23] or the one utilized in ACMENet [3] are usually considered for small clusters within a large network.

The Underwater Multirobot Cooperative Intervention MAC protocol (UMCI-MAC) is presented in this work. It has been designed to minimize delay and jitter, thus allowing an efficient communication among a group of AUVs in underwater cooperative interventions. UMCI-MAC proposes a TDMA strategy. One of the AUVs acts as *master* and coordinates the communication process, while the rest of AUVs react to *master* commands, in order to avoid packet collisions. Moreover, the protocol has the potential to prioritize the access to the channel of each *slave* as a service to the upper layers. The protocol is intented to be used in a small team of up to eight AUVs.

The rest of the paper is organized as follows. In Section II the UMCI-MAC protocol is described in detail. In Section III the performance of UMCI-MAC is evaluated and compared with that of Aloha-CS and S-FAMA protocols. In a first

experiment throughput, efficiency, collisions, end to end delay and its fluctuation (jitter) are obtained for several combinations of packet size and application data rate. A second experiment consisting of a Hardware In The Loop (HIL) control of a team of underwater AUVs is also presented. In this experiment results in terms of command delay, jitter and position errors are also obtained. Finally, in Section IV are summarized the main results presented in this manuscript.

II. UMCI-MAC

As mentioned in the previous section, delay and jitter are the parameters of communications that have the highest impact in robotic applications. In this work we introduce UMCI-MAC protocol for underwater acoustic channels the aim of which is to allow the control of up to 8 ROVs participating in a cooperative intervention.

The communication among the nodes in UMCI-MAC is organized by one of the nodes which is identified as *master*. The rest of the nodes are identified as *slaves* and act in response to the *master*'s messages. The communication process is based on rounds which are divided into control and data phases. The control phase is initiated by the *master* through a very short synchronization packet (SYNC) that serves as a reference point for the *slaves*. After the reception of a SYNC, each slave is assigned a time slot to send their request to transmit in a control packet (CTRL). The duration of each time slot equals the transmission time of a CTRL plus the maximum network propagation time. During the data phase the *master* gives each *slave* permission to transmit by means of a CTS packet. On the reception of a CTS, the designated *slave* will transmit its DATA packet. At the end of the data phase, the *master* can optionally send a DATA packet to some designated *slaves*, and will start a new round. The *master* will use the SYNC packet of the next round to acknowledge all the messages received from the *slaves*. The *slaves* will use the next round's CTRL packet to acknowledge the messages they received.

Every transmitted packet has a common structure, with a header (Fig. 1), a body (Fig. 2) and a Cyclic Redundancy Check (CRC-16) hash of the information contained in the header and the body.

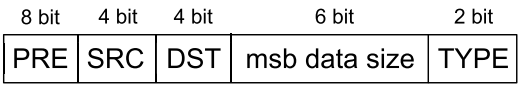


FIGURE 1. Header structure.

The header structure, shown in Fig. 1, contains an 8-bit preamble (PRE) to synchronize the reception of a new packet, two 4-bit fields to identify the transmitter (SRC) and the receiver (DST), a 6-bit field (msb data size) with the 6 most significant bits (MSB) of the message's length and a 2-bit field (TYPE) used to identify the four BODY messages considered in UMCI-MAC. All UMCI-MAC messages have the same header structure but different body structures shown in Fig. 2.

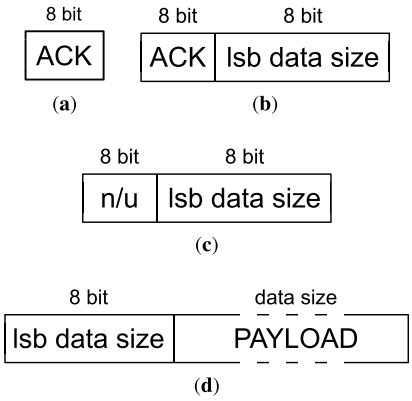


FIGURE 2. Body structure of the four packet types considered in UMCI-MAC. (a) SYNC. (b) CTRL. (c) CTS. (d) DATA.

The SYNC message is sent by the *master* to start each communication round. Its body is composed of a single 8-bit field (ACK), Fig. 2(a), one per AUV. The *master* sets to "1" the bit corresponding to a *slave* in order to acknowledge the reception of a message from this *slave* during the previous round.

Each node sends a CTRL message (Fig. 2(b)) during its time slot of the control phase which indicates the number of bytes that it wants to transmit in the current round. This number is encoded on 14 bits, the 6 MSB are in the header of the message, and the 8 less significant bits (LSB) are in the *lsb data size* field of the CTRL packet. The CTRL also contains an 8-bit ACK field that each node uses to acknowledge any packet received during the previous round, as the *master* uses the ACK field in the SYNC message.

CTS packets, shown in Fig. 2(c), are sent by the *master* during the data phase in order to give permission to each *slave* to transmit the data requested in its CTRL packet. The 8 bit *lsb data size* field contains the 8 LSB of the number of bytes that the corresponding *slave* is allowed to transmit. The remaining 6 MSB bytes are encoded in the HEADER. This packet also contains an 8-bit field labeled as "n/u" which have not been used but are reserved for future applications.

DATA is the only variable length packet, its structure can be seen in Fig. 2(d). The first byte of the DATA packet contains the lower 8 bits (LSB) of the number of bytes in the payload. The remaining (MSB) 6 bits are included in the HEADER.

A high level view of the *master*'s workflow is shown in Fig 3. The left side of the diagram corresponds to the control phase of each round, while the right side corresponds to the data phase of the round. The *master* starts each round broadcasting a SYNC packet and waits for the CTRL of the *slaves*. If the CTRL contains an ACK for the *master*, it dequeues such packet from its transmission queue. If the CTRL contains an RTS, the *master* places the request into the Transmission Request Priority Queue (TRPQ). During the data phase the *master* sends CTS messages of each transmission request in the TRPQ and waits for the maximum propagation time of the network plus the time the *slave* requires to transmit the amount of data requested in the

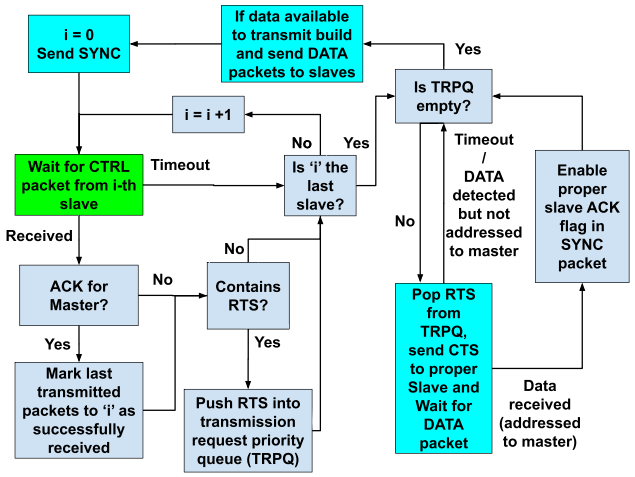


FIGURE 3. Workflow of actions carried by the *master*.

control phase. After processing all transmission requests in TRPQ the *master* checks if it has DATA available to transmit to any slave and sends it.

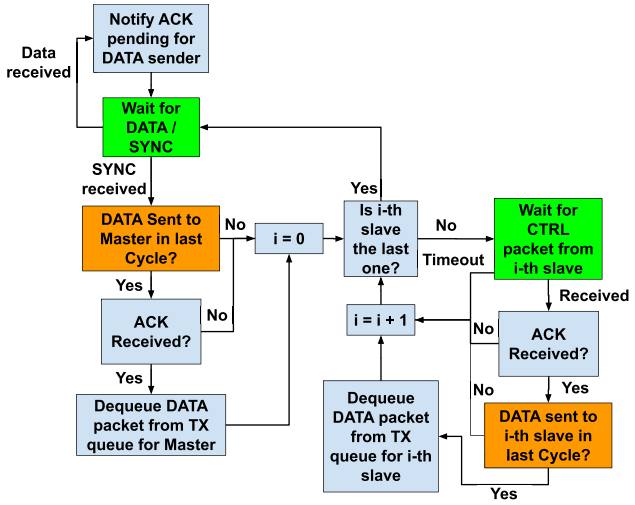


FIGURE 4. Workflow of the actions conducted by the *slave* to process the received messages.

In the case of the *slaves* the algorithm has been divided in two workflows which are executed concurrently, the one in Fig 4 focuses on the process of received messages, while the one in Fig. 5 prepares the messages to be sent by the *slave*. The workflow in Fig. 4 is in charge of processing received DATA packets, scheduling ACKs and dequeueing acknowledged DATA packets from the transmission queue. The actions described in this diagram are mainly related to the control phase. After the reception of the SYNC packet, the *slave* checks if the data sent to the *master* in the last round has been acknowledged. If so, it dequeues the data packet from the transmission queue. The *slave* applies the same procedure to the CTRL packets sent by the rest of *slaves*. After the control phase, this process will listen to the channel for DATA packets until it receives a new SYNC packet from

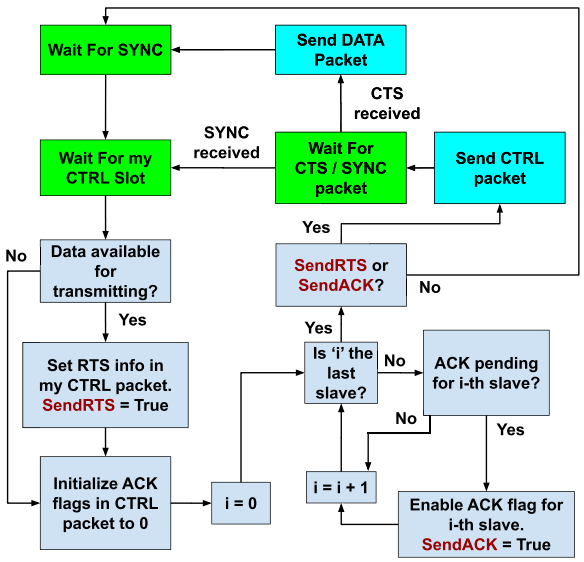


FIGURE 5. Workflow of the *slave* process dedicated to build and send DATA and CTRL packets.

the *master*. For each data packet received, the *slave* will schedule an ACK which will be sent in the next round's control phase.

The second process of the *slave* is dedicated to building and sending DATA and CTRL packets and it is shown in Fig. 5. After the reception of the SYNC packet the *slave* will compose a CTRL packet with the pending acknowledgements and the size of the DATA available to be transmitted. The *slave* will use its slot in the control phase to send a CTRL packet if needed. After the control phase, if the *slave* has not sent a RTS in a CTRL packet it will directly wait for the next SYNC packet. Otherwise it will wait for the CTS from the *master*. On the reception of the CTS packet, the *slave* will send its data and will wait for the next SYNC packet.

III. RESULTS

A. PERFORMANCE EVALUATION OF UMCI-MAC

The performance of the UMCI-MAC in terms of throughput, efficiency, end to end delay, jitter and data collisions is evaluated in this section. The end to end delay results have been measured in the layer above the MAC. They correspond to the time elapsed since a given packet is placed into the MAC layer of the transmitter node until it is recovered from the MAC in the receiver node. UWSim-NET [24], [25] was used to model the communications of four AUVs sending packets to a fifth one which acts as a sink. The sink AUV acted as the *master*, while the transmitting AUVs were the slaves of the UMCI-MAC protocol. All vehicles were equipped with an acoustic modem which has a transmission rate of 1800 bps and a maximum range of 100 meters. The transmission rate for the simulation of the modems in UWSim-NET has been configured considering the specification of the S2CR 18/34 acoustic modems [26], [27] and our previous experience using them in scenarios with severe multipath interferences [25].

The experiment consisted in the transmission of 200 packets from each AUV. All MAC protocols were forced to re-transmit every packet as many times as required in order for the transmission to be successful. The same test was repeated for several combinations of data packet sizes, application data rate, and was conducted with Aloha-CS and S-FAMA in order to compare their performance against UMCI-MAC. Additionally, the impact of the relative position of the AUVs in the performance of the MAC protocols was evaluated by repeating the previous experiment considering two different topologies. A linear topology, in which the four *slaves* were arranged in straight line, being the *master* in one of the edges. In the square topology, the four AUVs were placed forming a square and the *master* was placed in its center.

1) LINEAR TOPOLOGY

In this first topology, the four transmitter AUVs were placed in straight line, 20 meters from each other. All of them at the same depth, and holding their positions during the experiment. The sink (*master*) was located at a depth of eight meters above the first transmitter.

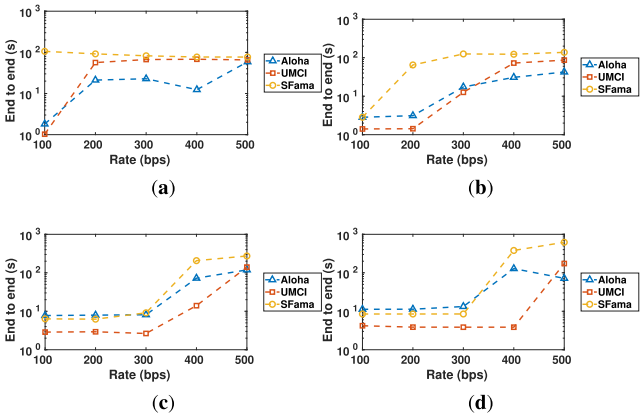


FIGURE 6. Variation of end to end delay for several packet sizes. (a) 20 bytes. (b) 100 bytes. (c) 400 bytes. (d) 600 bytes.

The average end to end delay measured during the experiment with the three MAC protocols is shown in Fig. 6. As it becomes apparent in Figs. 6(b)-6(d), the end to end delay slightly changes at low data rates. This corresponds to a normal working regime, in which the nodes are able to send data packets as they are generated. At a given data rate, it is not possible to transmit all data packets so that they are accumulated in the transmission queue, which causes a rapid increase of the end to end delay. As it can be noticed in Fig. 6, the maximum data rate for which the nodes are in normal regime increases with the packet size. The same behavior is appreciated in Fig. 6(a), but this behavior is not clearly seen because the normal regime is achieved with a 100 bps data rate, and only for the UMCI-MAC and Aloha-CS protocols. As expected, the end to end delay of all three protocols increases with the packet size, because of the larger transmit and receive times for each packet. However, independently of the packet size the UMCI-MAC exhibits the lowest end to end delay of all three protocols in normal regime.

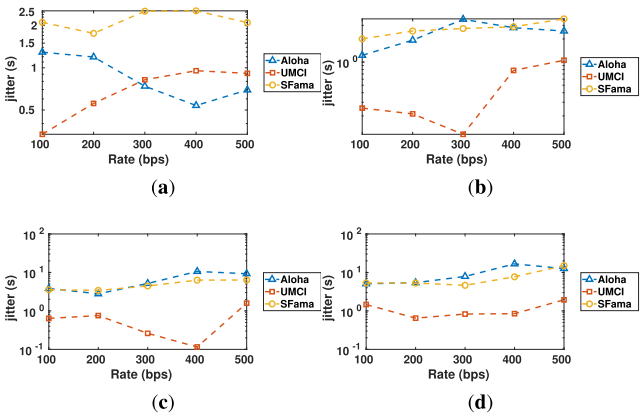


FIGURE 7. Variation of jitter for several packet sizes. (a) 20 bytes. (b) 100 bytes. (c) 400 bytes. (d) 600 bytes.

Fig. 7 shows the average jitter measured during each experiment. Except for the results in Fig. 7(a), where 20-byte long packets are considered, the UMCI-MAC achieved the smallest jitter of all three MAC protocols. The low jitter of UMCI-MAC is an indicator of its ability in delivering control messages at regular periods of time. When considering 20-byte long packets, Aloha-CS exhibits the lowest jitter at high data rates, while UMCI-MAC achieves the best performance at low data rates, see Fig. 7(a).

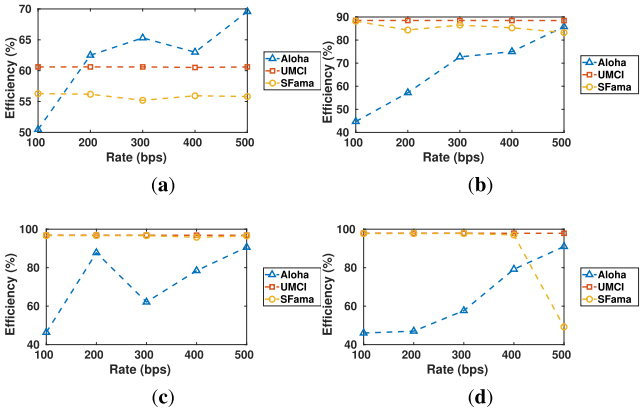


FIGURE 8. Variation of efficiency for several packet sizes. (a) 20 bytes. (b) 100 bytes. (c) 400 bytes. (d) 600 bytes.

The efficiency is defined as the ratio between the number of bits in the message to the total amount of bits required to transmit the message (including re-transmissions). UMCI-MAC and S-FAMA exhibit a similar behavior in terms of efficiency, Fig. 8. Their efficiency degrades when transmitting 20-byte long data packets because of the protocol overload of RTS and CTS messages (see Fig. 8(a). However, they reach an efficiency close to 100 % when data packets larger than 100 bytes are considered, as seen in Figs. 8(b)-8(d). Aloha-CS exhibits a different behavior from UMCI-MAC and S-FAMA in terms

of efficiency. Aloha-CS has a poor performance (≈ 40 %) at low data rates, which increases with data rate. The likelihood of detecting the channel occupied when trying to transmit a packet increases with the data rate. This strongly reduces the likelihood that two nodes start their transmissions at the same time. It also favours the fact that the nodes detect the channel occupied and wait until it is freed in order to start their transmission, thus making a more efficient use of the channel.

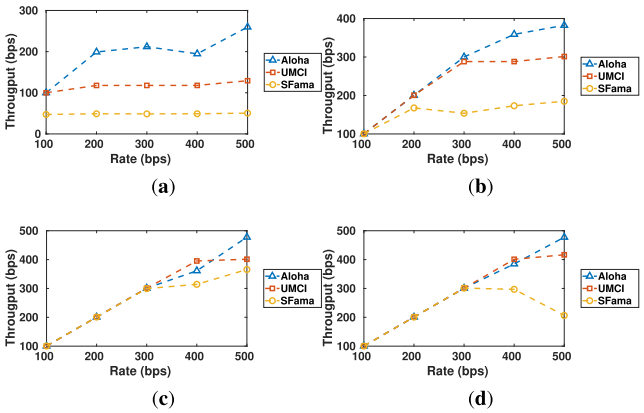


FIGURE 9. Variation of throughput for several packet sizes. (a) 20 bytes. (b) 100 bytes. (c) 400 bytes. (d) 600 bytes.

The comparison of the throughput achieved with all three MAC protocols is shown in Fig. 9. It is noticed that the measured throughput coincides with the data rate as long as the protocols work in normal regime. In all three protocols it is observed that at a given data rate the throughput saturates, and does not increase with the data rate anymore. A flat variation of throughput with data rate is seen in Fig. 9 when the modems are out of their normal regime. For all four packet sizes considered in this experiment it is noticed that the lowest throughput is measured with the S-FAMA MAC protocol while Aloha-CS achieves the highest one. Except for the case shown in Fig. 9(a), considering 20-byte long packets, the results in Figs. 9(b)-9(d) demonstrate that the performance of UMCI-MAC in terms of throughput is close to that exhibited by Aloha-CS.

Finally, in Fig. 10 the total amount of collisions during each experiment is illustrated. Only the collisions of the S-FAMA and Aloha-CS are shown in Fig. 10 because no collisions were measured with the UMCI-MAC protocol. The fact that the access to the medium is arbitrated by the *master* prevents that two nodes start to transmit at the same time in the UMCI-MAC protocol. Another effect shown in Fig. 10 is the fact that the amount of collided bytes in the Aloha-CS protocol decrease with the data rate. When the channel occupancy is high it is less likely that two nodes detect the channel free and start to transmit at the same time.

2) SQUARE TOPOLOGY

In the second topology considered in this work, the four transmitter AUVs were placed forming a square, all of them

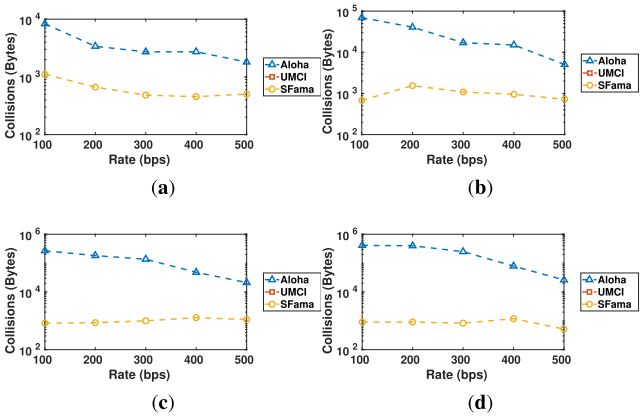


FIGURE 10. Variation of the amount of bytes collided bytes for several packet sizes. (a) 20 bytes. (b) 100 bytes. (c) 400 bytes. (d) 600 bytes.

at the same depth. The *master* was placed in the center of the square, at a depth of eight meters above the rest of the AUVs. All the nodes hold their positions during the whole experiment. In order to evaluate the impact of the distance between the AUVs in the performance of the communication link, two different squares of 20 m and 60 m were considered in this work. In the remaining pages of the manuscript they are identified as (20 × 20) and (60 × 60), respectively.

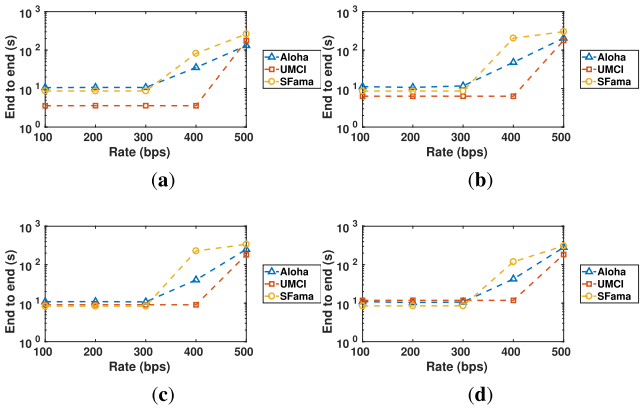


FIGURE 11. Variation of the end to end delay with the priority assigned to each AUV. All the measurements correspond to a 20 m square topology when sending 600 bytes long packets. (a) AUV 1. (b) AUV 2. (c) AUV 3. (d) AUV 4.

The results obtained with the square topology, either with the 20 × 20 or with the 60 × 60 squares, were similar to those obtained previously with the linear topology. One of the consequences of assigning a transmission turn to each AUV in UMCI-MAC is that the end to end delay of the transmissions from the AUV which receives the first token is smaller than those measured with the rest of AUVs, as it is shown in Fig. 11. This is explained by the fact that the *master* sends a CTS packet to each AUV in priority order, assigning the AUV 1 the highest priority and AUV 4 having the lowest one. In UMCI-MAC an AUV has to wait until all the AUVs with higher priority finish their transmission.

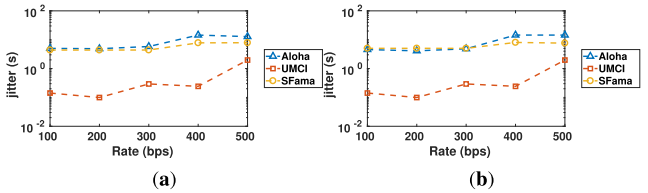


FIGURE 12. Variation of the jitter with the priority assigned to the AUV. All the measurements correspond to a 20 m square topology when sending 600 bytes long packets. (a) AUV 1. (b) AUV 4.

This effect is more noticeable the larger the data packets are, as is the case of the transmission of 600 bytes long packets shown in Fig. 11. When considering UMCI-MAC the end to end delay measurement of the AUV 4 is the highest of all AUVs. However, its value is comparable to that obtained with Aloha-CS and S-FAMA which achieve similar end to end delay values in all AUVs. In this way, the UMCI-MAC permits to selectively improve the end to end delay of one of the AUVs by assigning it the first turn in each transmission round. The results shown in Fig. 11 correspond to a square topology in which the AUVs formed a 20 × 20 m square. Similar results were also appreciated with the 60 × 60 square and the linear topologies, however we have chosen these results because all the AUVs were at the same distance from the *master* .

Despite the variation in the end to end delay measurements of each AUV shown in Fig. 11, the jitter measurements were not affected by the different priorities assigned to each AUV, as it is shown in Fig. 12. The results shown in Fig. 12 are comparable to those in Fig. 7(d) corresponding to the linear topology. In both cases it is noticed how UMCI-MAC achieved the lowest jitter of all the three protocols considered in this study. This proves the predictability of UMCI-MAC in the delivery of the messages.

The fact that all the AUVs are at the same distance in the square topology permits to analyze its impact on the performance of the communication links. The comparison of the results between the case in which the AUVs form a 20 × 20 *m* square, and the 60 × 60 *m* case has shown the decrease in the efficiency achieved with the Aloha-CS as the AUVs are more and more separated, see Fig. 13. The efficiency achieved by Aloha-CS when sending 20 bytes and 600 bytes data packets in the (20 × 20) square, shown in Fig. 13(a) and 13(b) respectively, decreases when the (60 × 60) square is considered, shown in Fig. 13(c) and 13(d). On the contrary, the efficiencies measured with the S-FAMA and the UMCI-MAC are not affected by the separation among the AUVs, and exhibit similar values to those obtained in the linear topology Fig. 8. The degradation of the efficiency in the case of the Aloha-CS might be explained by the fact that the more distant the AUVs are, the more time is required to detect that another AUV is transmitting. Thus increasing the number of collisions and re-transmissions. This does not occur with the S-FAMA and the UMCI-MAC because the permission to transmit is arbitrated by the RTS and CTS signals.

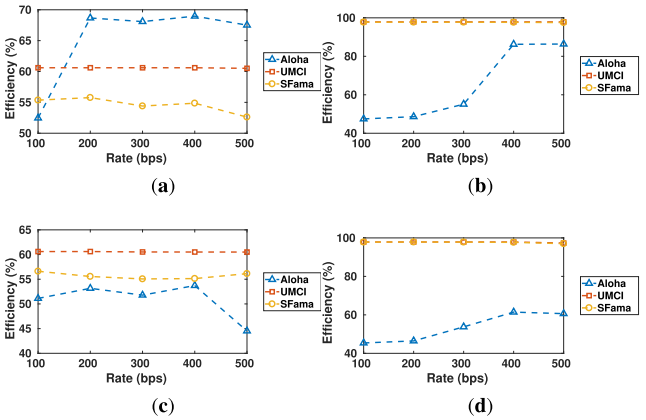


FIGURE 13. Variation of the protocol efficiency with the distance between the AUVs, when sending packets of 20 and 600 bytes in a 20 × 20 m and 60 × 60 m square topologies. (a) 20 bytes data packets in a 20 × 20 m square topology. (b) 600 bytes data packets in a 20 × 20 m square topology. (c) 20 bytes data packets in a 60 × 60 m square topology. (d) 600 bytes data packets in a 60 × 60 m square topology.

B. UMCI-MAC REMOTE CONTROL OF A TEAM OF AUVs IN HIL

In this section a HIL experiment consisting of the remote control of a team of four AUVs in a water tank is presented. One of the AUVs is a real one based on the BlueROV platform while the remaining three are simulated using the UWSim-NET simulator. The three simulated AUVs try to keep a relative position to the real AUV, which acts as the leader, forming a square of 2.5 by 2.5 meters. As in the previous experiment, all the communication devices have a transmission rate of 1800 bps and a maximum range of 100 m.

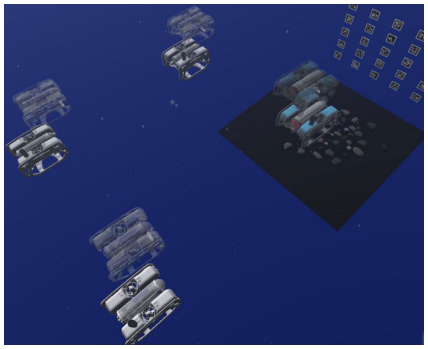
TABLE 1. Configuration of data flows from all the devices during the HIL expriment. Rate indicates the data rate (in bps) generated by the application. Size indicates the length of the DATA packet (in Bytes). Dest and Information indicate the receiver and the information contained in the data packet, respectively.

	Rate	Size	Dest.	Information
Buoy	100	50	BlueROV	Position commands
BlueROV	100	50	Broadcast	Own position (estimated)
AUVs	300	600	Buoy	Own position and images

Movement commands are transmitted from a simulated buoy on the surface located on the vertical axis of the initial position of the formation of AUVs, which are 54.5 m deep. The commands sent from the buoy are the target position and orientation which the leader is expected to reach. The real BlueROV, acting as leader of the formation, broadcasts its position while the simulated AUVs transmit their positions and visual information of the seafloor. The data flows sent by each device during the experiment are detailed in Tab. 1 in terms of application data rate, packet size, destination and the information contained in DATA packets.

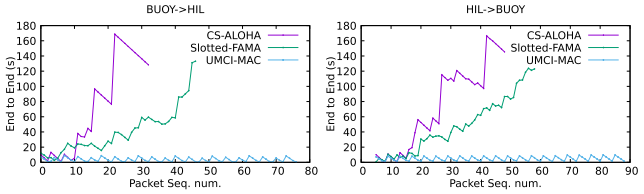
The position of the BlueROV is estimated by vision and using some ArUco markers [28] which have been placed on

(a)



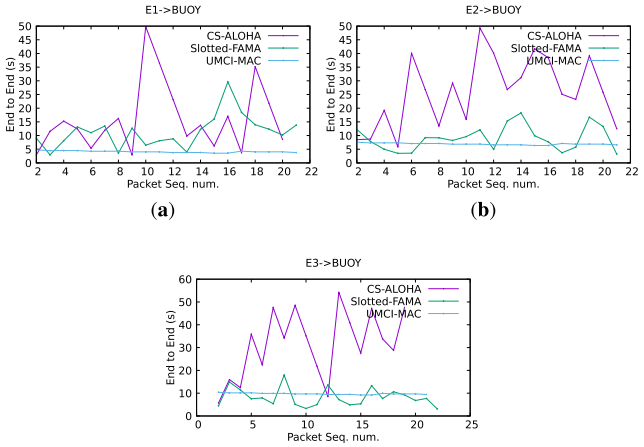
(b)

FIGURE 14. Detail of the HIL experiment: (a) Image of the real BlueROV inside the water tank during the experiment. (b) UWSim-NET virtual reality interface with an image of the whole experiment with the real and simulated AUVs.



(a) (b)

FIGURE 15. End to end delay (in s) of the transmitted packets. (a) From the buoy to the real BlueROV. (b) From the real BlueROV to the buoy.



(a) (b) (c)

FIGURE 16. End to end delay (in s) of the transmitted packets from the AUVs to the buoy. (a) AUV 1, (b) AUV 2, (c) AUV 3.

one of the walls of the water tank as shown in Fig. 14(a). Due to the space constraints of the water tank, movement commands sent to the leader do not have a distance greater than two meters. As can be seen in Fig. 14(b), the floor of the water tank and the markers have been replicated in UWSim-NET in order to create a virtual reality environment for the experiment. Also, the actual positions of the four vehicles are represented by opaque 3D models of BlueROV. In the case of the simulated vehicles their 3D model is white, while the model used for the leader has realistic colours. In addition, the desired position of each vehicle is represented using a transparent version of its 3D model. The desired position of the leader corresponds to the last command sent from the buoy. The desired position of each simulated AUV is where the vehicle should be in order to form a perfect square considering the actual position broadcasted by the leader.

The performance of all three protocols in this experiment is first evaluated in terms of end to end delay. The end to end delay of the communication between the Buoy and the leader is analyzed in Fig. 15. The delay measured with Aloha-CS and S-FAMA increases during the experiment in both the transmission from the buoy to the BlueROV Fig. 15(a) and the transmission from the BlueROV to the buoy Fig. 15(b). The rise of the delay is caused by the filling of the transmission queue. Packet collisions force the re-transmission of packets and avoid the proper emptying of the transmission queue. Contrary to Aloha-CS and S-FAMA, the UMCI-MAC protocol is able to send all the packets within a reasonable and almost constant delay. The end to end delay of the packets sent from the simmulated AUVs to the buoy is shown in Fig. 16. When transmitting large DATA packets Aloha-CS exhibits larger delays than UMCI-MAC and S-FAMA, this behavior was also noticed in the previous experiment. Comparable end to end delay measurements were achieved with UMCI-MAC and S-FAMA. The most remarkable result in this experiment is the increasing delay measured for the three AUVs with the UMCI-MAC protocol. This response is caused by the fact that AUV1 is assigned the highest priority, since it is the first one to receive the CTS from the *master* in every round. This is the reason why the end to end delays measured for AUV1, AUV2 and AUV3 were 5 s, 7 s and 10 s, respectively.

Finally, the errors between desired and actual positions of the AUVs are compared in order to evaluate the impact of the MAC protocol on the control of the AUVs. The position error of the leader BlueROV is illustrated in Fig. 17. The times when the position command changes are indicated in vertical red lines. When the BlueROV receives the command it starts to move to the specified position and an exponential decrease of the error is appreciated in all three protocols. However, in the case of the experiment using Aloha-CS and S-FAMA, due to the rise of the delay in command reception the BlueROV will not receive the commands during

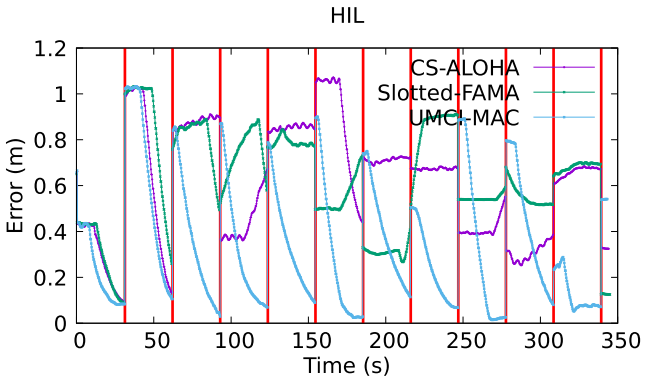


FIGURE 17. Difference between the actual and desired position of the BlueROV.

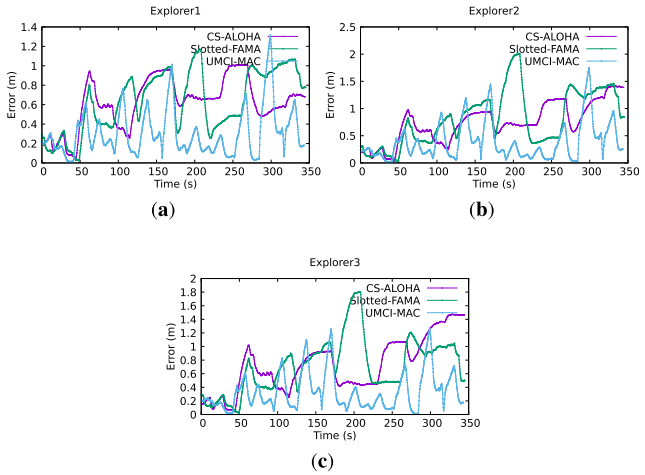


FIGURE 18. Differences between the actual and desired positions of the AUVs. (a) AUV 1, (b) AUV 2, (c) AUV 3.

the corresponding interval. This is the reason why a constant error is observed in Fig. 17 during the whole time interval in the S-FAMA and Aloha-CS curves, especially from $t = 100\ s$ onwards. The lower end to end delay achieved with the UMCI-MAC protocol allows the delivery of all command messages before the position command is changed. Thus, the error decreases in all periods when the UMCI-MAC is considered.

The error between the actual and desired positions of the AUVs is shown in Fig. 18. In all three figures it can be seen how the UMCI-MAC is able to broadcast the position of the leader in time and the AUVs update their positions according to the leader's, thus reducing the position error. However, the long delay of Aloha-CS and S-FAMA makes that the AUVs receive the message with the position of the leader after the actual position of the leader has changed. So they try to move to the previous position of the leader, this is the reason for the increase of the position error of the AUVs during the experiment.

IV. CONCLUSION

The UMCI-MAC protocol specific for wireless communications in cooperative robotics is presented in this work. The purpose of the UMCI-MAC is to make it possible to control of a team of up to 8 AUVs during a cooperative intervention. It was designed in order to minimize delay and jitter in UWN based on acoustic links. It is based on a TDMA strategy with a *master* node in charge of the coordination of the access to the medium of all the nodes, thus avoiding packet collisions. The protocol gives the *master* the capability of prioritize the transmissions of some nodes.

The performance of UMCI-MAC is compared against that of the Aloha-CS and S-FAMA. The performance of all three protocols is evaluated in the first experiment in terms of delay, jitter, throughput and collisions, considering several data rates and packet sizes. The results confirm that in all the scenarios considered in the first experiment the UMCI-MAC protocol achieved the lowest end to end delay and jitter values of all three protocols. The results presented in this manuscript also show how the UMCI-MAC permits to control which AUV will have the lowest end to end delay by assigning it the highest priority. The low delay and jitter measured with UMCI-MAC suggest that it might be considered for the control of a team of AUVs in underwater cooperative interventions. The arbitration of the access to the medium in UMCI-MAC makes it possible to avoid the packet collisions, and yields an efficiency superior to that measured with S-FAMA and Aloha-CS. The comparison of several topologies has shown that the S-FAMA and UMCI-MAC protocols were little affected by the relative positioning of the AUVs, since they exhibited a similar performance in all the topologies considered in this study. On the other hand, it has been observed how the efficiency of the Aloha-CS decreased as the distance among AUVs increased. The second experiment presented in this work consisted of a teleoperation of a team of 4 AUVs. This second experiment has demonstrated that the excellent performance of UMCI-MAC in terms of end to end delay and jitter makes it suitable for use in robotic applications. Of the three protocols compared in this work UMCI-MAC was the only one that was able to deliver the position commands in time so that the simulated AUVs were able to follow the movements of the leader.

REFERENCES

[1] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: Applications, advances and challenges," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 370, no. 1958, pp. 158–175, Jan. 2012. [Online]. Available: http://rsta.royalsocietypublishing.org/content/370/1958/158

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 257–279, May 2005.

[3] G. Acar and A. E. Adams, "ACMENet: An underwater acoustic sensor network protocol for real-time environmental monitoring in coastal areas," *IEE Proc.-Radar, Sonar Navigat.*, vol. 153, no. 4, pp. 365–380, Sep. 2006.

[4] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 84–89, Jan. 2009.

[5] D. Pompili and I. Akyildiz, "Overview of networking protocols for underwater wireless communications," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 97–102, Jan. 2009.

[6] S. Climent, A. Sanchez, J. Capella, N. Meratnia, and J. Serrano, "Underwater acoustic wireless sensor networks: Advances and future trends in physical, MAC and routing layers," *Sensors*, vol. 14, no. 1, pp. 795–833, 2014.

[7] H. Kaushal and G. Kaddoum, "Underwater optical wireless communication," *IEEE Access*, vol. 4, pp. 1518–1547, 2016.
[8] J. Dea, D. Radosevic, N. Tran, J. Chavez, and B. Neuner III, "Land and undersea field testing of very low frequency RF antennas and loop transceivers," SSC Pacific, San Diego, CA, USA, Tech. Rep., 2017.
[9] W. F. Subsea. *Datasheet of Seatooth s100-l Wireless Subsea Controller*. Accessed: Feb. 27, 2020. [Online]. Available: http://www.wfs-tech.com/wp-content/uploads/2017/11/Seatooth-S100-L-Exte nded-Range-17.11.1.pdf
[10] A. Shukla and H. Karki, "Application of robotics in offshore oil and gas industry—A review part II," *Robot. Auto. Syst.*, vol. 75, pp. 508–524, Jan. 2016.
[11] IRSLab. (2018). *TWINBOT Research Project*. [Online]. Available: http://www.irs.uji.es/twinbot/
[12] (2015). *MERBOTS Research Project*. [Online]. Available: http://www.irs.uji.es/merbots/
[13] T. Slama, A. Trevisani, D. Aubry, R. Oboe, and F. Kratz, "Experimental analysis of an Internet-based bilateral teleoperation system with motion and force scaling using a model predictive controller," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3290–3299, Sep. 2008.
[14] R. Wirz, R. Marin, M. Ferre, J. Barrio, J. M. Claver, and J. Ortego, "Bidirectional transport protocol for teleoperated robots," *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3772–3781, Sep. 2009.
[15] P. X. Liu, M. Q.-H. Meng, P. R. Liu, and S. X. Yang, "An End-to-End transmission architecture for the remote control of robots over IP networks," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 5, pp. 560–570, Oct. 2005.
[16] A. A. Syed, W. Ye, J. Heidemann, and B. Krishnamachari, "Understanding spatio-temporal uncertainty in medium access with ALOHA protocols," in *Proc. 2nd workshop Underwater Netw. (WuWNet)*, New York, NY, USA, 2007, pp. 41–48, doi: 10.1145/1287812.1287822.
[17] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, "A survey on MAC protocols for underwater wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1433–1447, 3rd Quart., 2014.
[18] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "Aloha-based MAC protocols with collision avoidance for underwater acoustic networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2007, pp. 2271–2275.
[19] M. Molins and M. Stojanovic, "Slotted FAMA: A MAC protocol for underwater acoustic networks," in *Proc. OCEANS Asia–Pacific*, May 2006, pp. 1–7.
[20] P. Xie and J.-H. Cui, "R-MAC: An energy-efficient MAC protocol for underwater sensor networks," in *Proc. Int. Conf. Wireless Algorithms, Syst. Appl. (WASA)*, Aug. 2007, pp. 187–198.
[21] A. A. Syed, W. Ye, and J. Heidemann, "T-lohi: A new class of MAC protocols for underwater acoustic sensor networks," in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Apr. 2008, pp. 231–235.
[22] D. Pompili, T. Melodia, and I. F. Akyildiz, "A CDMA-based medium access control for UnderWater acoustic sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1899–1909, Apr. 2009.
[23] W. Van Kleunen, N. Meratnia, and P. J. M. Havinga, "MDS-mac: A scheduled MAC for localization, time-synchronisation and communication in underwater acoustic networks," in *Proc. IEEE 15th Int. Conf. Comput. Sci. Eng.*, Dec. 2012, pp. 666–672.
[24] D. Centelles, A. Soriano-Asensi, J. V. Martí, R. Marín, and P. J. Sanz, "Underwater wireless communications for cooperative robotics with UWSim-NET," *Appl. Sci.*, vol. 9, no. 17, p. 3526, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/17/3526
[25] D. Centelles, A. Soriano, J. V. Marti, R. Marin, and P. J. Sanz, "UWSim-NET: An open-source framework for experimentation in communications for underwater robotics," in *Proc. OCEANS*, Jun. 2019, pp. 1–8.
[26] Evologics. *Specifications of s2cr 18/34 underwater acoustic modem*. Accessed: Feb. 27, 2020. [Online]. Available: https://evologics.de/acoustic-modem/18-34
[27] O. Kebkal, M. Komar, K. Kebkal, and R. Bannasch, "D-MAC: Media access control architecture for underwater acoustic sensor networks," in *Proc. OCEANS IEEE*, Jun. 2011, pp. 1–8.
[28] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320314000235

DIEGO CENTELLES was born in Castellón de la Plana, (Castellón), Spain, in 1991. He received the computer science engineering degree and the M.Sc. degree in intelligent systems from Universitat Jaume I, Castelló, in 2014 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Interactive and Robotic Systems Lab. He has participated in several research projects, such as TRITON, MERBOTS, and TWINBOT. His research interests include computer networks and robotics.

ANTONIO SORIANO was born in Benaguasil, Valencia, Spain, in 1978. He received the Licenciado degree in physics and the Ph.D. degree from the University of Valencia, Valencia, in 2001 and 2007, respectively. He has worked as a Postdoctoral Research at the Spanish National Researh Council (CSIC), the Universitat Politecnica de Valencia, and the Universitat Jaume I. Since 2017, he has been an Associate Professor with the Department of Informatics, University of Valencia. His research interests include computer networks and robotics.

JOSÉ V. MARTÍ received the Licenciado degree in physics from the University of Valencia, in 1990, and the Ph.D. degree in computer engineering from the Universitat Jaume I (UJI), Spain, in 2012. Since 2015, he has been a Lecturer with the Computer Science and Engineering Department, UJI. His research interests include robotics, localization, and communications.

PEDRO J. SANZ (Senior Member, IEEE) received the B.Sc. degree in physics from the University of Valencia (UV), the M.Sc. degree in engineering (CAD/CAM) from the Technical University of Valencia (UPV), and the Ph.D. degree in computer engineering from Universitat Jaume I (UJI), Spain.

He is currently a Full Professor with the Computer Science and Engineering Department, UJI, and the Head of the Interaction and Robotic Systems (IRS) Laboratory. His main research interests include devoted to multisensory-based grasping and dexterous manipulation, telerobotics and human–robot interaction (HRI), all of them applied to real life scenarios, including assistive and underwater robotics.

Dr. Sanz was a member of the Advisory Committee of the IEEE Systems Council, from 2008 to 2012. He was a recipient of the Best Thesis in Electronics and Computer Science Domains, in 1996, the National Research Prize from the Artigas Spanish Foundation (UPM), Madrid, Spain. Since 1990, he has been active in Research and Development within several projects on Advanced Robotics, such as the Coordinator of European FP7-TRIDENT Project, from 2010 to 2013, and the Coordinator of Spanish TWINBOT Project for the period of 2018–2020. He was appointed as a Visiting Scientist at different Universities, like TUM, Germany, in 2000 and 2016, Blaise Pascal, France, in 2002, and Bologna, Italy, in 2008. He was the Humanoids Competition Chair of the 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, and the Chair of several tutorials and workshops within outstanding International Conferences on Robotics (IROS, IFAC, and ICMA. He has served as an Associate Editor of some outstanding journals, such as the IEEE RAM and the IEEE SMC—C. He has been the Coordinator of the Spanish Robotics Network (CEA-IFAC), from 2012 to 2016.