A Receiver-initiated MAC Protocol for Underwater Acoustic Sensor Networks

Wen-Hwa Liao
Dept. of Information Management
Tatung University
Taipei, Taiwan
E-mail: whliao@ttu.edu.tw

Yu-Chieh Lin and Ssu-Chi Kuai
Dept. of Information Management
Tatung University
Taipei, Taiwan

Abstract—Due to the long propagation delay and the low data transmission rate, many transmissions of control packets will occupy a lot of channel resources, power consumption and transmission time for underwater acoustic sensor networks (UWSNs). In this paper, we propose a receiver-initiated MAC protocol with packet train for UWSNs called multi-receiver MAC (MR-MAC) protocol. Our protocol will increase of the throughput and reduce of the transmission of control packet for handshaking. Our MR-MAC protocol can make more than two nodes to communicate in one handshake held by a main receiver. By scheduling the packet transmission time, the data packet will be sent in a packet train manner and the receiver can receive data packet without collision. Our simulation results show that our protocol can increase the throughput and decrease the packet delay than the other existing MAC protocols.

Keywords—MAC protocol; receiver-initiated; underwater acoustic sensor networks (UASNs).

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of tiny, low-cost, low powers, multifunctional sensor nodes and communicate with each other in short range. In WSNs, media access control (MAC) protocol is one of the most important research issues. The MAC protocols are used to prevent collision when more than two transmissions occurred. There are several challenges to design an efficient MAC protocol for WSNs, e.g., energy waste caused by packet retransmission, the prevention of collision, overhearing and idle listening. Underwater acoustic sensor networks (UASNs) consist of a variable number of mobile sensors that are deployed to perform collaborative monitoring tasks over a predetermined area. UASNs can be used in the applications of oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation and tactical surveillance [1]. Underwater sensor nodes are very expensive, and the sensing areas of ocean environments are also large, which leads to the sparse network deployments and the widespread using of mobile sensors. In addition, underwater sensors will have corrosion problem and the capacity of battery is also limited. The battery cannot be recharged in simple way because it is hard to access to power sources such as solar in underwater [2].

In terrestrial WSNs, the handshaking-based protocols usually use the control packets (e.g., RTS/CTS) to communicate each other and to prevent the collisions. It suffers from two main drawbacks when these handshaking-based protocols are applied in UASNs. First, it needs at least one full round-trip exchange of control packets prior to sending each data packet, and such procedure will led considerable latency due to the long propagation delay. In existing sender-initiated handshaking based MAC protocols, only one sender is allowed to transmit single or multiple packets during a successful handshake. The other backlogged neighbors cannot transmit in this handshake until they get the channel reservation by another handshake. This leads to the under-utilization of the channel, and the low throughput. Second, the long propagation delay also seriously impacts the ability of the RTS/CTS handshake mechanism to resolve the hidden terminal problem, because it takes much longer for a node to receive RTS and CTS packets from its neighbors and extends the vulnerable period. This leads to higher collision rate and low throughput [4].

The rest of this paper is organized as follows. Section II discusses related work on MAC of UASNs. In Section III, we describe our protocol in details. Session IV describes the simulations that we analyzed. Finally, we conclude our paper in Section V.

II. RELATED WORK

There is amount of work on terrestrial MAC protocol, they are devoted to reserve energy and increase the system performance by avoiding collision. In [3], MACA and MACAW are well-known schemes for carrier sense multiple access (CSMA). The RTS packet and CTS packet exchange avoids collisions at the receiver, and solves the hidden and expose terminal problem. In [11], S-MAC uses the control packet overhearing to check whether the node is sleepy or not. The nodes have the sleep-listen schedule of their neighbors by broadcasting a SYNC packet. They use RTS/CTS to handshake following the schedule table. In [5], T-MAC decreases the energy consumption by setting adaptive active times to control the sleep and wake-up time. A node will keep listening and potentially transmitting, as long as it is in an active period. An active period ends when no activation event has occurred for a period.
In [6], they proposed a propagation-delay-tolerant collision avoidance protocol (PCAP). The protocol requires clock synchronization between the neighboring nodes in order to improve channel utilization and let a sender to perform other actions during the long wait between the RTS and CTS frames. Although its maximum throughput is 20%, which is higher than what the conventional handshaking protocol can achieve in UASNs, this is merely comparable to the throughput of Aloha protocol. In fact, the transmission of control packets in UASNs will decrease the channel efficiency because of the consideration of long propagation delay. For this reason, many researches focus on how to increase the utilization in one successful handshake. In [4], the main idea of RIPT is that multiple nodes are allowed to transmit data packets to a single receiver with each round of handshake. Upon completing the slot assignment, the receiver then transmits the ORDER packet and order transmission to each neighboring node. If there is any data packet to be sent, it will be transmitted along with ORDER packet. However, the receiver may broadcast RTR (ready-to-receive) incessantly while it’s idle and has no transmitter. A receiver occupying the channel also makes the networks unfair. In [9], they proposed a ROPA protocol, the sender broadcasts RTS packet to inform the receiver about the request to perform primary data transmission. Then the sender polls the potential secondary senders whether they have any data packet to append. After receiving RTS packet, the receiver and the secondary senders will reply a CTS and Request-to-Append (RTA), respectively. RTA-RTA or RTA-CTS collisions can be avoided in the same way with RIPT and MACA-MN. After receiving all the RTA and CTS, the sender broadcasts Clear-to-Append (CTA) to inform the receiver and the secondary sends the schedule and start to send and receive data packet. However, if the secondary sender has no packet to append, it still has to send the RTA packet. The control packet transmission period and data transmission period will occupy the channel by one node and make the network be unfair.

III. Multi-Receiver MAC (MR-MAC) Protocol

We first discuss the space-time uncertainty. The performance of high latency networks, such as UASNs, is affected by both space and time uncertainty. Second, we apply receiver-initiated reservation with 4-way handshake in our multi-receiver MAC (MR-MAC) protocol. Our protocol applies the packet train concept to increase the channel utilization after each successful handshake. Finally, we propose a receiver-initiated MAC protocol for UASNs.

A. Overview

In terrestrial WSNs, these MAC protocols can avoid the collision by preventing the interfering nodes from transmitting at the same time. However, the performance of high latency networks, such as UASNs, is affected by both space and time uncertainty [7][10]. The space uncertainty, on one hand, is caused by the nodes’ locations, which result in different propagation delays. The time uncertainty, on the other hand, is caused by the randomness of packet arrivals. About space-time uncertainty, the distance between the nodes translates into uncertainty of current channel status and a packet may collide even if there are no contenders sending simultaneously. In our protocol, the space-time uncertainty can be solved by carefully scheduling the control and data packets transmissions at the receiver based on the information of inter-node propagation delays.

In [4], unlike other transmitter-initiated protocols that encounter two types of packet collision, namely, “transmit-receive collision” and “receive-receive collision”. The receiver-initiated approach only experiences receive-receive collision. A receiver can know exactly when the packet will be received and provide better collision avoidance channel reservation process by scheduling the transmission.

Although the RTS/CTS mechanism is widely used for alleviating the hidden and exposed terminal problems in terrestrial WSNs, the RTS/CTS mechanism is not suitable when it is applied in UASNs. First, it needs at least one full round-trip time to exchange control packets. With the long propagation delay in UASNs, it causes the low channel utilization and low throughput. Second, the transmitting power is typically 100 times more than receiving power in UASNs. It is too expensive to make handshake with multiple control packets before each data transmission. To reduce the energy consumption, our protocol can increase the channel utilization by sending a train of packets after each successful handshake. Third, with the long propagation delay, the hidden terminal problem cannot be resolved by the RTS/CTS handshake mechanism because it takes a lot of time for a node to receive RTS and CTS packets from its neighbors. In this case, receiver-initiated reservations are better at avoiding collisions in the presence of long propagation delay, since the receiver has accurate information on its own current state.

Among the existing MAC protocols with packet train concept [4], they focus on using packet train to only one receiver. If neighbors have few packets to send to a receiver, the channel reservation will be wasted. None of the other backlogged neighbors can transmit data packet in this handshake, even though the channel has been reserved. Each node can only send data packet to their target node when they content the channel. To solve the problem, we try to put more than one receiver into each successful handshake. The main receiver holds this handshake, collects its neighbors’ target information and schedules the nodes which deserve to become the receiver. In this way, we can use a channel reservation more efficiently and raise the channel utilization.

B. Our MAC Protocol: Multi-Receiver MAC Protocol (MR-MAC)

We propose a receiver-initiated MAC protocol and its associated adaptations for UASNs. We consider static multi-hop underwater acoustic networks where each node is equipped with a single half-duplex underwater acoustic modem. We assume that each node knows its estimated propagation delay from each of its one-hop neighbors, as well as each of these neighbors’ maximum propagation delay from the latter’s one-hop neighbors. During the initialization, all nodes take turns to broadcast some control packets to its neighbors. According to the round-trip time measurements of control packets, a node can calculate its propagation delay by comparing the timestamp
on the packet with its local clock. The notations showed in Table 1 are used in our protocol.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>Main receiver in each receiver-initiated handshake</td>
</tr>
<tr>
<td>$i$</td>
<td>Secondary receivers in each receiver-initiated handshake which $i \in U_R$</td>
</tr>
<tr>
<td>$U_R$</td>
<td>Set of node $R$'s one-hop neighboring nodes</td>
</tr>
<tr>
<td>$N(U_R)$</td>
<td>The number of set $U_R$</td>
</tr>
<tr>
<td>$N_{sender}$</td>
<td>The number of sender to each main receiver or secondary receiver</td>
</tr>
<tr>
<td>$U'_i$</td>
<td>The number of sender to each secondary receiver $i$</td>
</tr>
<tr>
<td>$N(U'_{sender})$</td>
<td>The number of set $U'_{sender}$</td>
</tr>
<tr>
<td>$T_{ATR}$</td>
<td>Transmission time of each ATR</td>
</tr>
<tr>
<td>$T_{TNI}$</td>
<td>Transmission time of each TNI</td>
</tr>
<tr>
<td>$T_{MSP}$</td>
<td>Transmission time of each MSP</td>
</tr>
<tr>
<td>$T_{SP}$</td>
<td>Transmission time of each SP</td>
</tr>
<tr>
<td>$T_{DATA}$</td>
<td>Transmission time of total data packet</td>
</tr>
<tr>
<td>$T_{CRP}$</td>
<td>The end of the channel reservation phase</td>
</tr>
<tr>
<td>$P(R,a)$</td>
<td>Propagation delay from $R$ to neighboring node $a$</td>
</tr>
<tr>
<td>$S_{R,a}$</td>
<td>Set of the sort of $R$’s neighboring node in ascending order for all $a \in U_R$</td>
</tr>
<tr>
<td>$S_{i,a}$</td>
<td>Set of the sort of $i$’s senders in ascending order for all $a \in U_R$</td>
</tr>
<tr>
<td>$P(S_{R,a})$</td>
<td>Propagation delay from $R$ to neighboring node $a$ in $S_{R,a}$ by the propagation delay</td>
</tr>
<tr>
<td>$T_{end}(S_{R,a})$</td>
<td>The finish time of packet transmission for node $a$ in $S_{R,a}$ by the propagation delay</td>
</tr>
<tr>
<td>$d_{TNI}(a)$</td>
<td>Delay time when neighboring node $a$ transmits TNI</td>
</tr>
<tr>
<td>$d_{DATA}(a)$</td>
<td>Delay time when sender node $a$ transmits data</td>
</tr>
</tbody>
</table>

1) **Channel Reservation Phase:**

Instead of the typical 3-way (RTS/CTS/Data) handshake in protocols such as MACA, our MR-MAC protocol is a receiver-initiated 4-way (ATR/TNI/MSP or SP/Data) handshake. As shown in Fig. 1, when an idle node try to act as a main receiver (MR), it will broadcast an Ask-To-Receive (ATR) packet to start a 4-way handshake. ATR packet informs neighboring about the main receiver’s ID and the target nodes to which MR wants to transmit. When a neighboring node receives ATR, it will reply a TNI packet if it has data to send.

If more than two neighboring nodes relay TNI at the same time when they receive an ATR, the receiver node will calculate the time of transmission for each sending node to avoid an collision occurred. We assumed that each node know the propagation delay from the one-hop neighbors, therefore our protocol can avoid the collision by calculate $d_{TNI}(a)$. The collision-free TNI packet transmission algorithms is shown as Fig. 2.

1. Sort neighboring node in ascending order $S_R[a]$ by the propagation delay for all $a < N_R$
2. $T_{end}(S_R[1]) = T_{ATR} + 2P(S_R[1])$
3. $d_{TNI}(S_R[1]) = 0$
4. **for** $k = 2$ to $|N_R|$
5. **if** $T_{ATR} + 2P(S_R[k]) > T_{end}(S_R[k - 1])$
6. $T_{end}(S_R[k]) = T_{ATR} + T_{TNi} + 2P(S_R[k])$
7. $d_{TNI}(S_R[k]) = 0$
8. **else**
9. $d_{TNI}(S_R[k]) = T_{end}(S_R[k - 1]) - 2P(S_R[k]) - T_{ATR}$
10. $T_{end}(S_R[k]) = T_{end}(S_R[k - 1]) + T_{TNi}$
11. **end if**
12. **end for**

First, the algorithm sorts the neighboring nodes in ascending order base on the propagation delay. Then, it uses the receiving time of first TNI packet to determine the transmission time of other TNI packets. In TNI packet, there are two types of information for neighboring nodes, as the target node ID and the amount of data packet to each target node. Assume that data packets can be classified according to the priority. The neighboring nodes choose the target nodes which have the data packets with the highest priority and inform the main receiver by responding the TNI packet. When the main receiver gets the neighboring nodes’ target nodes information by receiving TNI packet, it will choose the proper neighboring nodes to be the receiver. If the chosen neighboring node has senders which want to send data packets to it, they will become secondary receivers (SR) in data transmission phase. An example of target node information is shown in Fig. 3 when the main receiver receives TNI packets.

The neighboring node $C$ is not a secondary receiver because there is no sender to send data to it. If a neighboring node receives an ATR packet but has no packet to send out, it
will keep silent in the duration $d_{\text{wMSP},a}^{\text{TR}}$ and wait for the main schedule packet (MSP). If the node overhears a TNI but not an ATR, the node still has to wait for a silent duration $d_{\text{wMSP},a}^{\text{TNI}}$ to avoid MSP collision. The silent durations of ATR packet and TNI packet are as equation (1) and equation (2):

$$d_{\text{wMSP},a}^{\text{TR}} = T_{\text{CRP}} + P(R,a) + T_{\text{MSP}} - T_{\text{ATR}} - P(R,a)$$

$$d_{\text{wMSP},a}^{\text{TNI}} = T_{\text{CRP}} + T_{\text{MSP}} - T_{\text{ATR}}$$

$$d_{\text{wMSP},a}^{\text{TR}} = T_{\text{CRP}} + P_{\text{max}} + T_{\text{MSP}} - [T_{\text{ATR}} + P(S_{R_i}) + T_{\text{TNI}} + P(S_{R_i,a}) + d_{\text{TNI},a}]$$

(2)

2) Channel Reservation Phase:

In channel reservation phase, after the transmission of TNI packets between the main receiver and neighboring nodes, they immediately switch their roles to data transmission. At the beginning of data transmission phase, as shown in Fig. 1, the main receiver transmits MSP to the secondary receivers which are chosen in channel reservation phase. In MSP, the main receiver specifies three types of information, schedule of senders, sequence of all receivers, and all receivers’ sender list. Schedules of senders are calculated in Fig. 4 to avoid data-data collision. Our scheme uses the end time of receiving data packet to calculate the next data packet transmission time. The MR will send data packet immediately after it broadcasts the MSP packet to decrease the idle time of channel. The receiver will use the first end time of MR to consider the MSP and additional data packet, and the last sender’s ID in the schedule of senders to determine the schedule packet (SP) transmission time by next secondary receiver. The sequence of all receivers will be known by each secondary receiver node when it becomes a receiver. According to the last sender’s ID and this sequence, the next secondary receiver can send its SP after it receives the last data packet. The sender list in MSP for all receivers’ contains all senders to each receiver.

1. Sort senders in ascending order $S[a]$ by the propagation delay for all $a < N_{\text{sender}}$
2. $S[0] = i$
3. if $S[0] = \text{Main receiver} R$
4. $T = T_{\text{MSP}}$
5. else
6. $T = T_{\text{SP}}$
7. end if
8. $T_{\text{end}}(S[0]) = T + T_{\text{DATA}}$
9. for $k = 1$ to $N_{\text{sender}}$
10. if $2P(S[k]) + T > T_{\text{end}}(S[k-1])$
11. $d_{\text{DATA}}(S[k]) = 0$
12. $T_{\text{end}}(S[k]) = 2P(S[k]) + T + T_{\text{DATA}}$
13. else
14. $d_{\text{DATA}}(S[k]) = T_{\text{end}}(S[k-1]) - 2P(S[k]) - T$
15. $T_{\text{end}}(S[k]) = T_{\text{end}}(S[k-1]) + T_{\text{DATA}}$
16. end if
17. end for

In Fig. 5, the nodes $R, B$ and $A$ are receivers in this handshake and the corresponding senders in all receivers’ sender list. Each secondary receiver can get the information of its senders and the amount of packets by receiving the MSP, and then it starts to transmit data packet following the schedule in the MSP. As the algorithm in Fig. 4, the data packets will be collision free and will form a packet train at the receiver. Different from the main receiver, the secondary receiver sends SP to inform its senders’ schedule of data packet transmission. The format of the packet is showed in Fig. 6. The SP contains the schedule of senders. However, when a node receives a MSP and it has no packet to send to this receiver, it will keep silent to avoid interference. The neighboring node gets the information of secondary receivers by the sequence of all senders. On one hand, when the next secondary receiver is located in transmission range, this neighboring node will wait a duration $d_{\text{silent},a}^{\text{MSP}}$ for the next SP. The calculate of equation $d_{\text{MSP},a}^{\text{silent}}$ as equation (3):

$$d_{\text{MSP},a}^{\text{silent}} = T_{\text{end}}(S_{\text{TNI}}) - T_{\text{CRP}} + 2P_{\text{max}}, \text{where } k = N(U_{\text{sender}})$$

Figure 4. Algorithm for collision-free schedule of data packet $d_{\text{DATA}}(a)$.

If the neighboring node only receives a data packet, it will get the information of remaining senders in data packet. To avoid the interference of next SP transmission, we compute the silent duration $d_{\text{DATA}}^{\text{silent}}$, as follows:

$$d_{\text{DATA}}^{\text{silent}} = T_{\text{end}}(S_{R_{a}}) + 2P_{\text{max}} - [T_{\text{end}}(S_{R_{a}}) - P(S_{R_{a}}) + P(S_{R_{a}}, a)]$$

(4)

In MR-MAC protocol, a node needs to become as a receiver and to initiate the handshake by broadcasting an ATR packet. In traditional receiver-initiated MAC protocols, the
nodes decide to initiate the channel by checking the packets queue size. If the queue size is larger than a threshold, the node cannot be a receiver due to the avoidance of data flow blocking. In our MR-MAC protocol, it does not take this problem into consideration for even when a node becomes the main receiver it still has the opportunity to send its data packet.

IV. Simulation Results

In this section, we describe the simulations performed to evaluate the performance of our MR-MAC protocol. We compare our protocol with two previously proposed MAC protocols, RIPT and slotted FAMA by evaluating the important media access metrics such as throughput and packet delay.

A. Simulation Model

In our network environment, there are 36 static sensor nodes arranged in a grid spacing of 3,500m, and each node can deviate from the grid intersection point by a maximum of 10% of the grid space. Due to the nodes placement which is usually non-equidistantly, this deviation is introduced here in order to simulate the real scenario. The maximum transmission range of each node is 1.75 times the grid space, and we can make sure that the eight one-hop neighbors within its transmission range. The signal of acoustic propagation speed in underwater environments is assumed to be fixed at 1,500m/s. Each node is equipped with half-duplex and the data rate of the acoustic modem is 2,400 bps. The packet length of ATR, TNI and SP is 100 bits, MSP is 200 bits and DATA is 2,400 bits. For RIPT, the packet length of all control packets (i.e., RTR, SIZE, and ORDER) is the same length of 100 bits. Also, the length of the RTS, CTS and ACK are 64 bits. Two buffers are maintained by each node for relayed and new packets. In order to maintain each node with exactly eight neighbors, we adopt the wraparound setting of network environment. That is, even a node located on the boundaries, it still has eight one-hop neighbors. Our simulation network topology is shown in Fig. 7.

![Figure 7. Our simulation network topology.](image)

B. Simulation Result

1) Throughput:

We compare our MR-MAC’s throughput with two handshaking based protocols, RIPT [4] and slotted FAMA [8]. Here, we adopt the definition of throughput from [4], and define “throughput per node” as the average throughput over 36 nodes as follows:

\[
\begin{array}{c|c|c}
\text{No. of Packets Received/Simulation Time} & \text{Data Rate/Packet Length} \\
\hline
1 & 36 \\
\end{array}
\]

In Fig. 8, we can observe that throughput of three MAC protocols increases along with the increase of the offered load when the offered load is small, and they increase smoothly after offered load reach a values. The value of MR-MAC, RIPT and slotted FAMA is 0.08, 0.06 and 0.03, respectively. As expected, slotted FAMA is very inefficient because of setting time slot in maximum propagation delay, and it sends each packet at the beginning of time slot. It will waste much time in waiting for the control packet and data packet. Furthermore, the receiver in slotted FAMA can only receive data packets from only one neighbor for each round of handshake. Moreover, we can observe that our MR-MAC’s maximum throughput outperforms RIPT and slotted FAMA. The reason is that our protocol can permit more than one node to become the receiver in one handshake. After the control packet transmission between the main receiver and its neighbors, they can send more than one packet train to each receiver compared with RIPT.

![Figure 8. Comparison the throughput of MR-MAC, RIPT and slotted FAMA.](image)

2) Packet delay:

In Fig. 9, we compare the average data packet delay of three protocols. At very low offered load such as 0.01 or 0.02, our protocol’s packet delay is the highest in three protocols. In our MR-MAC protocol, it has to handle more than one pair nodes communication. For this reason, we use more complicate control packet transmission to reserve channel. If there is low offered load in the network environment, our protocol will spend a lot of time in control packet transmission and less data packet to send compared with other protocol. Therefore, the packet delay will be higher. However, with the increase of offered load, slotted FAMA has the highest packet delay because of the time slot setting of maximum propagation delay. Each pair of nodes has to spend a lot of time to make channel reservation, and it will lead to the low channel utility.
In the higher offered load environment, our MR-MAC can use the reserved channel more efficiently by sending more data in each handshaking. For this reason, our packet delay will decrease when the offered load becomes higher. Our protocol’s packet delay begins to rise when the offered load increases. However, our protocol has the lowest packet delay in among three protocols.

3) Number of packet collisions:

Fig. 10 shows the comparison of the number of packet collisions. We can observe that the number of packet collisions of our MR-MAC is less than RIPT. In our MR-MAC, after the channel reservation, the protocol uses the control packet before the packet train transmission. The header of data packet forwards the transmission information to the nodes which could cause the interference. Therefore, our protocol can reduce the packet collisions and increase the channel efficiency. Slotted FAMA has the lowest packet collisions among three protocols because it makes the channel reservation and uses two full time slots, which are two maximum propagation delays. Therefore, it can make sure the data transmission to avoid collision. However, it wastes a lot of time in waiting the control packet transmission and causes the low channel efficiency.

V. CONCLUSION

It is necessary to consider the long propagation and energy consumption in UASNs. Hence, we proposed a receiver-initiated MAC protocol with packet train in UASNs. In each successful four-way handshake, the main receiver and its neighboring nodes transmit and receive data packet to their target nodes in the main receiver’s transmission range. The main receiver will schedule the secondary receiver order and calculate the propagation delay to deal with the data packet train. Therefore, our protocol can decrease the time of channel reservation and end-to-end delay, and increase the efficiency of the channel. The simulation results show that our protocol can achieve higher throughput and lower data packet delay than two other MAC protocols.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Council, Taiwan, under grant NSC102-2221-E-036-012, and Tatung University, under grant B101-N02-051.

REFERENCES