A Set of Simplified Scheduling Constraints for Underwater Acoustic MAC Scheduling

Wouter van Kleunen, Nirvana Meratnia, Paul J.M. Havinga
Pervasive Systems, University of Twente
7522 NB Enschede, The Netherlands
(w.a.p.vankleunen, n.meratnia, p.j.m.havinga)@utwente.nl

Abstract—The acoustic propagation speed under water poses significant challenges to the design of underwater sensor networks and their medium access control protocols. Similar to the air, scheduling transmissions under water have significant impacts on throughput, energy consumption, and reliability. Although the conflict scenarios and required scheduling constraints for deriving a collision-free schedule have been identified in the past, applying them in a scheduling algorithm is by no means easy. In this paper, we derive a set of simplified scheduling constraints and propose two scheduling algorithms with relatively low complexity for both known and unknown orders of transmissions. Our experimental results show that scheduling without slots is on average 22% better than scheduling with slots for large packet sizes, while for small packet sizes scheduling without slots is about 40% better. We also compare our "smallest delay first" heuristic algorithm with the "highest transmission load first" heuristic of ST-MAC [1] and show that our heuristic algorithm performs on average 13% better.

I. INTRODUCTION

The unique properties of the underwater acoustic channel pose significant challenges on the design of underwater sensor networks (UWSN). The acoustic propagation speed, which is five orders of magnitude slower than the radio, makes existing radio-based medium access control (MAC) designs unsuitable for underwater sensor networks. Acoustic networks suffer from limited bandwidth, high transmission energy costs, and variations in channel propagation. This imposes new requirements which can not be met by existing radio-frequency based communication solutions.

Examples of existing underwater MAC protocols include T-Lohi [2], Slotted-FAMA [3], and ST-MAC [1]. All these MAC protocols consider data communication only. However, there is a need for reliable network protocols which provide not only data communication but also localization and time synchronization. For instance collaborative beam forming via which signals of different sensors are combined to provide directionality and to increase transmission range and consequently to reduce power consumption greatly relies on existence of time synchronization and localization. We strongly believe that an integrated approach has significant advantages over three separate solutions. Therefore, we aim to develop a collision-free MAC protocol that provides both time synchronization and localization in an energy efficient way and with high throughput. As the first step towards achieving this goal, in this paper, we develop a centralized collision-free scheduling algorithm for fully-connected single-hop underwater sensor networks.

Both [1] and [4] relate to our objectives as they provide a way to schedule the transmissions in underwater communication in such a way that no collision occurs at the receiver. Because of the spatial-temporal uncertainty, exclusive access to the medium is not required for collision free communication. Rather transmission times should be scheduled such that no collision occurs at reception. Figure 1 shows how two packets can be transmitted at the same time but are received without collision at the receiver.

The approach in [1] uses graph-coloring for scheduling, which may be cumbersome and time-slots, which is suboptimal. In addition, the authors do not model the processing time of the packet, which results in reception of the packet spanning through several time-slots. The approach described in [4], on the other hand, cannot guarantee to be collision-free as it only considers one previous scheduled transmission and not all previous scheduled transmissions.

In this paper, after reviewing the related work described in Section II, we present a set of simplified scheduling constraints to enable much simpler scheduling for underwater sensor networks. In Section III we explain how to derive these constraints. We further show the application of this set of constraints in two scheduling algorithms. Given a traffic flow and network topology, the algorithms are able to schedule transmissions such that no collisions occur and the total schedule length is minimized. The first algorithm, described in Section V, will calculate the shortest transmission schedule using a given order of transmissions. While the
second algorithm, described in Section VI, will use a heuristic approach to find the order of transmissions which will yield the shortest schedule length. Performance evaluation of our scheduling algorithms will be presented in Section VII, while Section VIII concludes the paper by drawing conclusions and highlighting future directions.

II. RELATED WORK

The goal of scheduling is to coordinate the transmissions to avoid conflicts. A valid schedule should therefore follow certain constraints to avoid packet collisions at the receiver. The scheduling problem of underwater transmissions basically boils down to determining an order for transmissions. Once the order of transmission is selected, the transmission times can be calculated to form a collision-free schedule. In case there are \( n \) number of transmissions, the number of possible schedules is \( n! \), because \( n! \) possible orders of \( n \) transmissions are possible. For all these orders of transmission a collision-free schedule exists.

To determine the efficiency of a schedule, an optimization criteria must be selected. A possible criteria for an optimal schedule may be a minimum schedule length, which implies that the maximum bandwidth will be achieved. But this may not always be the most critical criteria. Another criteria which has been chosen by, for example STUMP-WR [5], is to schedule the transmission based on hop-distance from the sink. This causes the routing towards the sink in a multihop network to be faster.

For some applications the only constraint on the schedule might be that it is a valid schedule, no collisions occur at the receivers, and no packets are lost due to collisions. In these cases a random transmission order can be chosen and the scheduling constraints can be applied to calculate a valid schedule.

In both [1] and [4], the scheduling constraints for underwater communication have been identified. They are derived from the four possible conflicts that may occur during communication, namely: TX-TX conflict, TX-RX conflict, RX-RX conflict and RX interference (see Figure 2). As presented by [1] and [4], these constraints can be used to form a mixed integer linear programming model which can in turn be used to calculate the optimal solution to the underwater scheduling problem. In the next section we will show how these constraints can be simplified and how by dropping a single optimization possibility a set of constraints can be derived which allows much easier scheduling.

III. DERIVING A SET OF SIMPLIFIED SCHEDULING CONSTRAINTS

We aim at scheduling transmissions in a fully-connected dense underwater sensor network. We assume that positions of all the nodes are known beforehand and network transmissions are pre-determined and fixed. We adjust the scheduling constraints of [1] and [4] so that they illustrate how one transmission task can be scheduled after another transmission task. In other words, let us consider two transmission tasks denoted as \( \delta_i \) and \( \delta_j \) and let us assume that \( \delta_j \) should be scheduled after \( \delta_i \). We aim to determine what constraints should be applied between the current and all previous transmissions. In case we want to schedule transmission \( \delta_{i+1} \), we should verify the constraints with all transmissions from \( \delta_0 \) up to and including transmission \( \delta_i \).

For each transmission, we need to calculate the transmission starting time \( \delta_{start} \). Every transmission has a certain duration \( \delta_{duration} \), source \( \delta_{src} \) and destination \( \delta_{dst} \). We assume the unspecified function \( T \) will give the transmission delay between two nodes.

We specify the constraints in such a way that transmission \( \delta_j \) is scheduled after or at the same time as transmission \( \delta_i \). In other words:

\[
\delta_j, start \geq \delta_i, start \quad \text{if} \quad j \geq i \tag{1}
\]

We will now discuss the scheduling constraints and show their formulation.

- TX-TX conflict: This case occurs when two transmissions are scheduled from the same source. We assume that the nodes are equipped with a single physical interface and therefore are not capable of transmitting multiple packets at the same time. To prevent occurrence of this conflict, this transmission should be scheduled with a delay so that the first transmission \( (\delta_i) \) is finished when the second transmission \( (\delta_j) \) starts. This can be formulated as:

\[
\delta_i, start + \delta_i, duration \leq \delta_j, start \tag{2}
\]

- TX-RX conflict: A node can not receive a packet when it is transmitting a packet. So the second transmission \( (\delta_j) \) should start after the first transmission \( (\delta_i) \) has been received or the second transmission should be finished before the first transmission has reached the first links destination. These two conditions are formulated as:
RX-RX conflict: In this type of conflict a single node is
RX-Interference: To avoid that two transmissions in-
4
11
5
3
If we model transmission as a node receiving its own
In the case of RX-RX conflict, because
For all the rules except rule (11)
Both equation (6) and equation (7) are applied at the
• Both equation (6) and equation (7) are applied at the
• For all the rules except rule (4) we can choose any
• In the case of RX-RX conflict, because δ_i,dst = δ_j,dst,
• If we model transmission as a node receiving its own
Because we want to schedule the transmissions in a given
However when scheduling a transmission, all relations to all transmissions before the “to
For our algorithm we aim to move the starting time to any
possible starting time greater than the minimum starting time, which will make the complexity of scheduling using these
rules much lower. Therefore we choose to drop rule (4). As a
result, when an algorithm uses this set of rules, it may give a slightly suboptimal solution in some cases.

Summing up all the assumptions we made, we basically only have to follow two scheduling constraints, namely equation (2) and equation (8):

- if transmission $\delta_i$ is from the same source as transmission $\delta_j$ ($\delta_i, src = \delta_j, src$), we can calculate the minimum transmission time using equation (2).
- if the transmission is sent from different nodes $\delta_i, src \neq \delta_j, src$, we should apply equation (8).

Because we have dropped equation (4), the starting times following from these equation are a minimum starting time. In other words, we can select any starting transmission time in the future as long as it is after the minimum starting time (see equation (9)).

The resulting set of simplified scheduling constraints are summarized in Figure 3.

V. A QUADRATIC COMPLEXITY ALGORITHM FOR SCHEDULING A FIXED ORDER OF TRANSMISSIONS USING THE SIMPLIFIED SCHEDULING CONSTRAINTS

We will now describe a scheduling algorithm with $O(n^2)$ complexity for scheduling a given transmission order. This scheduling algorithm uses the simplified scheduling constraints described in equation (3) to find the minimum transmission times which give no packet collisions at the receivers.

What the algorithm presented in Figure 5 does is scheduling the transmissions one by one and calculating a delay for every other transmission if this will be the next transmission to be scheduled. Initially all delays are 0 and the first transmission to be scheduled will be sent at time 0. Then all delays compared to the first transmission will be calculated. After that we sequentially go over all other transmissions to be scheduled.

![Image of scheduling algorithm](image_url)

Fig. 5. Scheduling algorithm for a fixed order of transmissions

$$V \leftarrow \text{transmissions}$$
$$c \leftarrow [N \times N] = 0$$
$$\delta_0, \text{start} \leftarrow 0$$

for $i \in \text{range}(1, \text{Length}(V))$ do

end for

$\delta_0, \text{start} \leftarrow 0$

for $i \in \text{range}(1, \text{Length}(V))$ do

end for

$\delta_{\text{min}} \leftarrow \infty$

for $i \in \text{range}(1, \text{Length}(V))$ do

end for

$\delta_{\text{min}} \leftarrow 0$

for $i \in \text{range}(0, \text{Length}(V))$ do

end for

$S \leftarrow \text{time} \leftarrow 0$

for $i \in \text{range}(1, \text{Length}(V))$ do

end for

$S \leftarrow \text{time} \leftarrow \text{time}_{\text{end}}$

for $i \in \text{range}(0, \text{Length}(V))$ do

end for

$\text{time}_{\text{end}} \leftarrow \max(\text{time}_{\text{end}}, \delta_i, \text{start} + T(\delta_i, src, \delta_i, dst) + \delta_i, \text{duration})$

end for

\begin{align*}
V & \leftarrow \text{transmissions} \\
c & \leftarrow [N \times N] = 0 \\
\delta_0, \text{start} & \leftarrow 0 \\
\text{for } j \in \text{range}(0, \text{Length}(V)) \text{ do} \\
\text{c}[0][j] & = \text{constraint}(V[0], V[j]) \\
\text{end for} \\
\text{time} & \leftarrow 0 \\
\text{for } i \in \text{range}(1, \text{Length}(V)) \text{ do} \\
\text{delay} & \leftarrow \max(c[i - 1][S[j]], \text{constraint}(V[i], V[S[j]]) \text{)} \\
\text{if } S[j]! = i \text{ and delay} < \delta_{\text{min}} \text{ then} \\
\delta_{\text{min}} & \leftarrow \text{delay} \\
\text{link}_{\text{min}} & \leftarrow S[j] \\
\text{end if} \\
\text{end for} \\
T[i] & \leftarrow S[\text{link}_{\text{min}}] \\
\delta_{\text{S[link}_{\text{min}}].start} & \leftarrow \text{time} + \text{delay} \\
\text{for } j \in \text{range}(0, \text{Length}(V)) \text{ do} \\
\text{c}[i][j] & \leftarrow \max(c[i - 1][j], \text{constraint}(V[i], V[j])) - \text{delay} \\
\text{end for} \\
\{\text{Remove scheduled transmission from transmissions to be scheduled}\} \\
S & \leftarrow S - \text{link}_{\text{min}} \\
\text{end for} \\
\{\text{Calculate the end-time of the schedule}\} \\
\text{time}_{\text{end}} & \leftarrow 0 \\
\text{for } i \in \text{range}(0, \text{Length}(V)) \text{ do} \\
\text{time}_{\text{end}} & \leftarrow \max(\text{time}_{\text{end}}, \delta_i, \text{start} + T(\delta_i, src, \delta_i, dst) + \delta_i, \text{duration}) \\
\text{end for} \\
\{\text{Store the optimal schedule (min time)}\} \\
\text{if } \text{time}_{\text{end}} < \text{time}_{\text{min}} \text{ then} \\
\text{time}_{\text{min}} & \leftarrow \text{time}_{\text{end}} \\
\text{schedule}_{\text{min}} & \leftarrow T \\
\text{end if} \\
\text{end for} \\
\}

Fig. 5. Heuristic algorithm for finding transmission order with minimum schedule time

![Image of scheduling algorithm](image_url)

Fig. 4. Calculating the new delay for transmission $\delta_3$ after scheduling $\delta_2$ after $\delta_1$
VI. A HEURISTIC ALGORITHM FOR FINDING TRANSMISSION ORDER WHICH YIELDS MINIMUM SCHEDULE TIME

In this section we will describe a heuristic algorithm for finding a minimum schedule time from all possible orders of transmissions. In other words, this algorithm will find the order of transmissions in such a way that the length of the schedule is minimum. To do this, we will take a similar approach as described in Section V. However this time the order of transmission is not given. We will use a greedy approach and every time a new transmission is to be scheduled we will take the transmission with the minimal delay. We will evaluate every transmission as first transmission and take the minimum schedule from all evaluated schedules. This algorithm therefore has \( O(n^3) \) complexity.

The outer loop of the algorithm presented in Figure 6 goes through all possible transmissions and starts the inner loop with every possible transmission as the first transmission. Then all the delays to the other nodes are calculated in the same way as has been done in the algorithm presented in Figure 5. We now have to schedule all the other transmissions, but because the order is not given, we will have to select a transmission to be scheduled next. To do this, we search through the set of calculated delays and schedule the link with the minimal delay first. After we have scheduled the transmission we update the delays in the same way as in the algorithm in Section V.

We will calculate the end-time of the schedule as we did before and check if this is the minimum schedule time. Then we will calculate \( n \) schedules and from all these schedules we select the schedule which has the minimal length.

\[
\text{throughput} = \frac{n \times \text{packet size}}{\text{schedule time}}
\]

We performed this experiment for different number of nodes/transmissions using the following setups: 16 nodes with 16 transmissions, 32 nodes with 32 transmissions, and 64 nodes with 64 transmissions.

For every setup we repeated the experiments 1000 times, every run has a different random deployment of nodes and different random transmissions. For every experiment we store the schedule time and after the 1000 runs we calculate the average, min and max. From the results shown in Figure 9(a) and Figure 8(a), we can see that when more transmissions are to be scheduled, the algorithm will calculate more efficient schedules.

![Fig. 7. Simulation parameters](image)

![Fig. 8. Plot of scheduling results for our scheduling algorithm](image)

![Fig. 9. Throughput of our scheduling algorithm in different setups](image)

We also repeated the experiment but this time scheduled 256 byte packets to be sent. The results of this test can be seen in Figure 9(b) and Figure 8(b). From the results it is clear that larger transmissions are easier to schedule and the results from the small number of transmissions are close to the results of the larger number of transmissions.

To compare our scheduling approach with slotted scheduling approaches such as the one used in ST-MAC [1], we looked at the impact of using slots for scheduling. To do so, we took the schedules we found during the experiments and moved the starting times to a slot boundary of 0.3s, which is the smallest slot size mentioned in [1]. The effects of using slots can be seen in Figure 11 and Figure 10(a). One can observe that in case of having smaller packet size (or higher bandwidth, which also results in smaller packet processing time), the use of slots...
A heuristic algorithm with (b) Min. delay vs max. load heuristics

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16 nodes, 16 transmissions</td>
<td>641.91 bps</td>
<td>880.78 bps</td>
<td>37 %</td>
</tr>
<tr>
<td>32 nodes, 32 transmissions</td>
<td>679.90 bps</td>
<td>953.25 bps</td>
<td>40 %</td>
</tr>
<tr>
<td>64 nodes, 64 transmissions</td>
<td>701.30 bps</td>
<td>1020.41 bps</td>
<td>45 %</td>
</tr>
</tbody>
</table>

Average diff.: 41 %

(b) Setup with nodes scheduled to send large packets (256 bytes)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16 nodes, 16 transmissions</td>
<td>948.05 bps</td>
<td>987.75 bps</td>
<td>4 %</td>
</tr>
<tr>
<td>32 nodes, 32 transmissions</td>
<td>961.04 bps</td>
<td>999.09 bps</td>
<td>4 %</td>
</tr>
<tr>
<td>64 nodes, 64 transmissions</td>
<td>968.16 bps</td>
<td>1008.13 bps</td>
<td>4 %</td>
</tr>
</tbody>
</table>

Average diff.: 4 %

Fig. 11. Comparison of a slotted and unslotted scheduling approach

Fig. 12. Comparison of smallest delay versus maximum load heuristics

VIII. CONCLUSION

Scheduling the transmission in an underwater acoustic communication network can be beneficial in terms of lowering down packet loss, energy-consumption and latency. However, scheduling the transmissions is not easy. In this paper we have shown how the scheduling constraints for underwater acoustic communication can be reduced and re-defined into a set of simplified scheduling constraints.

Our set of simplified scheduling constraints will still yield a schedule that is free of collisions at the receiver and is significantly easier to schedule. To show this, we propose two scheduling algorithms:

1. An algorithm with $O(n^2)$ complexity to find the minimum schedule time given a fixed order of transmissions.
2. A heuristic algorithm with $O(n^3)$ complexity to find the optimal order of transmissions which yields the minimum schedule time from all possible orders of transmission.

We compared our results against slotted approaches and different heuristics. We have shown that using slotted scheduling can result in significantly worse schedules for small packet size. We have also shown that the heuristic used by ST-MAC [1], which schedules high-load transmissions first, results in larger schedule times on average.

From our results, we can conclude that the difference between scheduling methods becomes smaller when the ratio between packet processing time propagation delay becomes larger. So for networks with smaller packet sizes or higher data-rates, the impact of the scheduling approach becomes more important.

In the future we will look at different algorithms which exploit the simplified scheduling rules. We will also investigate whether it is still possible to further optimize the proposed algorithms and obtain a scheduling algorithm with a lower complexity. Another area of future research is investigating implementation of scheduling in a distributed manner.

ACKNOWLEDGMENT

This work is supported by the SeaSTAR project funded by the Dutch Technology Foundation (STW).

REFERENCES