MAC SCHEDULING IN LARGE-SCALE UNDERWATER ACOUSTIC NETWORKS

Wouter van Kleunen, Nirvana Meratnia and Paul J. M. Havinga
Pervasive Systems, University of Twente, 7522 ND, Enschede, The Netherlands

Keywords: MAC Scheduling, Large-scale underwater acoustic networks, Spatio-temporal scheduling.

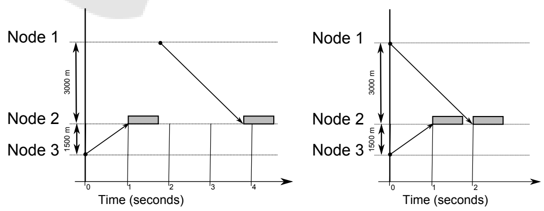
Abstract: The acoustic propagation speed under water poses significant challenges to the design of underwater sensor networks and their medium access control protocols. Scheduling allows reducing the effects of long propagation delay of the acoustic signal and has significant impacts on throughput, energy consumption, and reliability. In this paper we propose two approaches for scheduling large-scale underwater networks. One approach is a centralized scheduling approach, which yields the most efficient schedules but has significant communication and computational overhead. Our second approach uses clustering to split up the network in independent clusters. This approach provides significant benefits in terms of communication and computation, but yields less efficient schedules than the centralized approach.

We evaluate both approaches in terms of efficiency, communication overhead and computation overhead of the resulting schedule. We show that the centralized approach yield the best schedule at the cost of significant communication and computation overhead. The distributed scheduling approach yield less efficient schedules but offers significant communication and computational complexity advantages.

1 INTRODUCTION

Design of an efficient medium access control protocol for underwater acoustic networks is difficult. Underwater communication suffers from the slow acoustic propagation speed. Several underwater MAC protocols exist, examples include T-Lohi (Syed et al., 2007) and Slotted-Fama (Molins, 2006). However these protocols do not provide very high bandwidth, while there is an interest in scheduling transmissions to provide higher bandwidths. Examples of scheduled MAC protocols include ST-MAC (Hsu et al., 2009), STUMP (Kredo II and Mohapatra, 2010). In our previous paper (van Kleunen et al., 2011) we have proposed a MAC scheduling approach which is far much simpler than existing approaches. We have achieved this simplicity by first deriving a set of simplified scheduling constraints and using these constraints in design of a simple scheduling algorithm for underwater communication.

Scheduling communication brings significant advantages to underwater acoustic networks in that it exploits spatio-temporal uncertainty and reduces the impact of long propagation delays. Figure 1 illustrates how scheduling can exploit the long propagation delays to provide higher throughputs.



(a) Exclusive access (b) Scheduled

Figure 1: Exploiting spatio-temporal uncertainty of underwater communication.

Centralized MAC scheduling approaches are able to calculate efficient schedules but this comes with significant costs. To run a centralized algorithm all link and node information should be collected at a central place, which introduces a significant communication overhead. If the network is large or dynamic, it may not be feasible or desired to collect all link information at a central place. The computational complexity of a centralized algorithm can also be high. To this end, the ability to run the algorithm distributedly on several nodes can reduce the overall computation time. Using a distributed approach will come at the cost of suboptimal schedules, which should be reduced as much as possible.

In this paper we present how the complexity of

van Kleunen W., Meratnia N. and J. M. Havinga P..
MAC SCHEDULING IN LARGE-SCALE UNDERWATER ACOUSTIC NETWORKS.
DOI: 10.5220/0003438600270034
In *Proceedings of the International Conference on Wireless Information Networks and Systems* (WINSYS-2011), pages 27-34
ISBN: 978-989-8425-73-7

$$
\text{given } j \text{ for all } i < j, \begin{cases} \delta_j.start \geq \delta_i.start + \delta_i.duration & \text{if } \delta_i.src = \delta_j.src \\ \delta_j.start \geq \delta_i.start + \delta_i.duration + max(& \\ \quad T(\delta_i.src, \delta_i.dst) - T(\delta_j.src, \delta_i.dst), & \text{if } \delta_i.src \neq \delta_j.src \\ \quad T(\delta_i.src, \delta_j.dst) - T(\delta_j.src, \delta_j.dst)) & \end{cases}
$$

Figure 2: Set of simplified scheduling constraints.

the algorithm proposed in (van Kleunen et al., 2011), which allows exploiting the long propagation delays in underwater acoustic communication, can be reduced to allow scheduling of large-scale underwater networks. This reduced complexity centralized approach can be used in scenarios in which the communication overhead is not an issue and the aim is to achieve the best possible schedule.

We will also present how a clustered approach can significantly reduce the computational and communication overhead of scheduling large-scale underwater networks. A large part of the communication of our distributed clustered approach is done locally (single hop) and the computational communication is low enough to be implemented on embedded processors. The distributed clustered approach calculates less efficient schedule but the communication and computational overhead scales much beter to larger-scale underwater networks.

The outline of the paper is as follows: in Section 2 the related work will be discussed. An extended set of simplified scheduling constraints for scheduling multi-hop networks will be presented in Section 3. In Section 4 a centralized algorithm with reduced complexity is proposed. Section 5 describes our distributed scheduling technique, which uses a clustering concept to split up the scheduling problem. Evaluation of the communication and computational complexity of different approaches is presented in Section 6, while performance evaluations of the different approaches will be presented in Section 7. In Section 8 conclusions are drawn and future directions are highlighted.

2 RELATED WORK

Scheduling approaches for underwater acoustic transmissions, which allow mitigating the effects of the propagation delay, already exist in literature. These approaches range from centralized to distributed approaches. ST-MAC (Hsu et al., 2009) is a centralized scheduling approach, which uses timeslots to form a collision free schedule. In (van Kleunen et al., 2011) we have shown that the slotted approach used by ST-MAC leads to suboptimal results.

STUMP-WR (Kredo II and Mohapatra, 2010) provides a fully distributed approach to schedule underwater communication. It also uses timeslots, similar to the approach of ST-MAC, which has been shown to lead to suboptimal results. STUMP-WR derives a node schedule from local interference patterns and link schedules from neighbouring nodes. Nodes broadcast their route as well as link schedule updates during control frames until the network converges. This approach is quite interesting because it is fully distributed. Nodes are able to schedule their transmissions based on their own information and schedules of neighbours. This may, however, require a significant amount of communication between nodes and special attention should be paid to ensure that the nodes converge to a schedule in networks facing packet loss.

Because of the way STUMP-WR works, the possibilities of ordering the transmissions are limited. The first schedules will be formed using a limited amount of transmissions. Other nodes will extend their schedules using their transmissions but will not move existing transmissions to improve schedule times. This will result in less optimal schedules from a throughput point of view.

Another problem with STUMP-WR (Kredo II and Mohapatra, 2010) might be that transmissions still cause interference at certain nodes. This is because in a real network setup, nodes may not always be able to communicate with nodes that they might interfer. The interference range might actually be larger than the communication range. The completely distributed approach is an interesting approach nonetheless.

In (van Kleunen et al., 2011), we have already shown how the scheduling constraints for underwater communication can be reduced to a simplified set. Using these constraints we introduced two scheduling algorithms: one algorithm assumes a given order of transmissions and the other one selects the most optimal order of transmissions and derives a schedule. The set of scheduling constraints assumes all nodes are within transmission range of each other and all schedulings are done in a centralized manner. This necessitates availability of knowledge from all nodes and transmissions at a central place in the network, which introduces a significant communication overhead. In Figure 2 the set of simplified scheduling constraints are given. The set gives a constraint between two transmissions (δ_i and δ_j), which both have a source ($\delta_i.src$ and $\delta_j.src$), destination ($\delta_i.dst$ and $\delta_j.dst$), and duration ($\delta_i.duration$ and $\delta_j.duration$). The function T gives the propagation time between two positions. The simplified set gives a constraint between transmission δ_j and all previously scheduled transmissions (for all δ_i where $i < j$). A minimum start time can be calculated which satisfies the constraint with all earlier schedule transmissions. The proposed algorithm in (van Kleunen et al., 2011), which minimizes the schedule length, works by first calculating a two-dimensional table with the minimal delay between all pairs of transmissions. The algorithm considers every transmission as first transmission, calculates a minimal schedule with every transmissions as first transmission, and selects the best schedule. The schedule is calculated by scheduling transmissions iteratively. The transmission which has the minimum delay with respect to all previous transmissions is selected as next transmission. This is a greedy approach. The delays for all transmissions with respect to all previous scheduled transmission are updated after each iteration. The greedy approach of trying to minimize the delays leads to the most efficient schedule, in terms of schedule length. The resulting schedule is collision-free for all nodes in the network and it considers all nodes to be within communication range (one-hop). The transmission times are continous, although the algorithm can also be used for calculating slotted transmission times.

In (Stojanovic, 2008) and (Peleato and Stojanovic, 2007) the possibility of underwater cellular networks has been investigated. In the latter a protocol is proposed for channel sharing using cellular network, which only considers communication from and to the base-station.

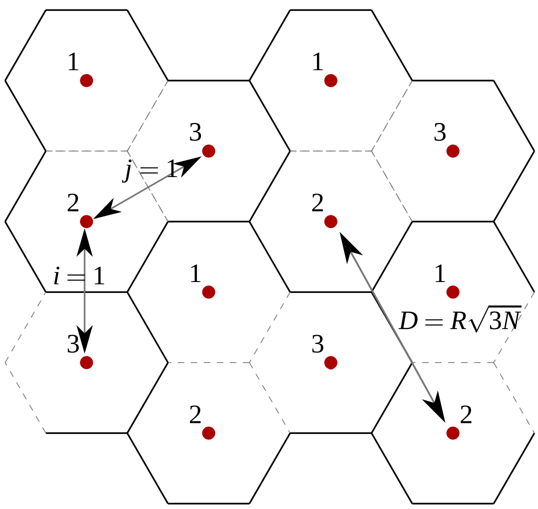


Figure 3: Cellular network example.

Figure 3 shows an example of a cellular network. The whole area is split into hexagonal cells and every cell is assigned a frequency. A group of cells together form a cluster, within which a frequency is used only once. The number of cells within a cluster determines the reuse distance (D), i.e, the minimum distance between two cells that share the same frequency.

The work of Peleato et al. (Peleato and Stojanovic, 2007) does some form of scheduling by splitting up the time into two phases. In the first phase, nodes that are the furthest away from the base-station are allowed to transmit. In the second phase, the inner nodes will transmit. These phases should run more-or-less synchronized between the cells. The cells experience the most interference from surrounding cells during the second phase, while the first phase will be more or less free of interference. The nodes in the second phase are closer to their destination (which will be the base-station) or source (also the base-station). Therefore the signal will experience less attenuation and is able to achieve the desired signal to noise ratio even when the reuse distance (D) is small. Due to the fact that the reuse distance is small a higher throughput can be achieved.

3 EXTENDING THE SET OF SIMPLIFIED RULES

Before we begin describing our proposed scheduling algorithm, we first extend the set of simplified scheduling rules and explain how multihop scheduling can be added.

Two nodes are outside of interference range of each other if the signal of one node results in a received signal strength on the other node which is below a certain threshold (TH_{cp}). The value of this threshold (TH_{cp}) should be chosen in such a way that interfering signals are always below the receiver sensitivity of the node or the interfering signal can be guaranteed to be captured by the wanted transmission.

The received signal strength is dependant on the output power of the sender and the attenuation between the sender and the receiver. The attenuation between nodes is dependent on the absorption rate of the water and the spreading of the signal. This path loss equation can be written as follows:

$$10\log(d,f) = k \cdot 10\log d + d \cdot 10\log a(f) \quad (1)$$

The path loss is dependant on the carrier frequency (f) of the signal as well as the distance (d) between sender, and receivery. The spreading factor is

$$\begin{cases} \delta_j.start \geq \delta_i.start + \delta_i.duration & \text{if } \delta_i.src = \delta_j.src \\ \delta_j.start \geq \delta_i.start + \delta_i.duration + max(& \text{if } \delta_i.src! = \delta_j.src \text{ and } (Interfer(\delta_i.src, \delta_j.dst) \\ \quad T(\delta_i.src, \delta_i.dst) - T(\delta_j.src, \delta_i.dst), & \quad \text{or } Interfer(\delta_j.src, \delta_i.dst)) \\ \quad T(\delta_i.src, \delta_j.dst) - T(\delta_j.src, \delta_j.dst)) & \\ \delta_j.start \geq \delta_i.start & \text{otherwise} \end{cases} \tag{2}$$

Figure 4: Extended set of simplified scheduling constraints allowing multi-hop scheduling.

constant, which can either be spherical ($k = 2$), cylindrical ($k = 1$), or somewhere in between.

Using this formula we can calculate whether two nodes interfer with each other. Consider two transmissions δ_i and δ_j, both transmissions have a source ($\delta_i.src$ and $\delta_j.src$) and destination ($\delta_i.dst$ and $\delta_j.dst$). We will use the path loss function (PL) to calculate the difference of the received signal strengths at the destination of transmissions (δ_j):

$$\begin{gathered} Interfer(\delta_i, \delta_j) = TRUE \text{ if} \\ PL(\delta_j.src, \delta_j.dst) - PL(\delta_i.src, \delta_j.dst) \leq TH_{cp} \end{gathered} \tag{3}$$

Function (3) will return $false$ if transmission δ_i does not cause interference for transmissions δ_j. We will now show how this equation can be applied to the set of simplified scheduling rules. The interference rule only applies when two nodes are able to interfer with each others transmissions. If $\delta_i.src$ is out of range of $\delta_j.dst$ and $\delta_j.src$ is out of range of $\delta_i.dst$, there is no constraint between the two transmissions.

In Figure 4 the set of extended scheduling constraints is shown. We added the interference condition to the interference rule and added a scheduling rule in case two transmissions are outside interference range. This set of constraints can be used in large networks where nodes can be outside of each others interference range.

4 A CENTRALIZED SCHEDULING APPROACH FOR HIGH-EFFICIENCY SCHEDULES

The extended set of simplified constraints from Section 3 can be applied to design a scheduling algorithm with low complexity for large-scale underwater networks. The algorithm from (van Kleunen et al., 2011), which has $O(n^3)$ complexity, considers every transmission as the first transmission. To reduce the complexity, we can take a random or the first transmission as the transmission to be scheduled at time 0. This will reduce the complexity of the algorithm from $O(n^3)$ to $O(n^2)$.

When we calculate the schedule only once, there is also no need anymore to precalculate a table of delays for all transmission pairs. Any transmission pair will be considered at most once, but some will never be calculated. At the first iteration the algorithm will calculate the delays for $n-1$ pairs, the second iteration $n-2$, and so forth. This will further reduce the complexity from $O(n^2)$ to $O(\frac{1}{2}n^2)$. Because we do not calculate the delay table, the memory space complexity can also be reduced to $O(n)$.

The full algorithm can be seen in Figure 5. The algorithm initially schedules the first transmission. Inside the scheduling loop first all the minimum starting times for the remaining transmissions are calculated. The loop also finds the transmission with the minimum schedule time and removes this transmission from the set of to be scheduled transmissions. This is repeated until all transmissions are scheduled. In this way not only the computational complexity reduced, but also the algorithm is now small and easy to understand.

5 A DISTRIBUTED SCHEDULING APPROACH WITH LOW COMPUTATIONAL AND COMMUNICATION COMPLEXITY

In Section 4 we have presented an algorithm for scheduling large-scale underwater networks in a centralized manner. However this algorithm requires multi-hop communication to gather information about all required transmissions within the network. This has a significant overhead and because it is done before scheduling, this communication will be done in an unscheduled way.

To solve this communication overhead problem we propose a distributed scheduling approach based on a clustering concept. We propose a technique in which cluster-heads are time-schedule arbiters for a cluster and nodes will send a request to the cluster-head to do a communication. The clusters are assigned a timeslot, which can span up to several seconds and will schedule all the requested transmissions

$V \leftarrow transmissions$ {Set of all transmissions}
$schedule \leftarrow [N] = 0$ {Resulting schedule}
$schedule[0] = 0$ {Schedule the first transmission}
$time = 0$
$last = 0$
$V \leftarrow V \setminus \delta_0$ {Remove transmission from set}
{Scheduling loop schedules transmissions greedy}
while $!empty(V)$ **do**
  $time_{min} \leftarrow infinity$
  {Calculate minimum starting time for remaining transmissions}
  for $\delta \in V$ **do**
    $schedule[\delta_{index}] = max(schedule[\delta_{index}], time + constraint(\delta_{last}, \delta_{index}))$
    {See if this transmission has the smallest starting time}
    if $schedule[\delta_{index}] < time_{min}$ **then**
      $time_{min} \leftarrow schedule[\delta_{index}]$
      $index \leftarrow \delta_{index}$
    end if
  end for
  {Schedule transmission with smallest starting time first}
  $time = time_{min}$
  $last = index$
  $V \leftarrow V \setminus \delta_{index}$
end while

Figure 5: Reduced complexity algorithm for scheduling transmissions.

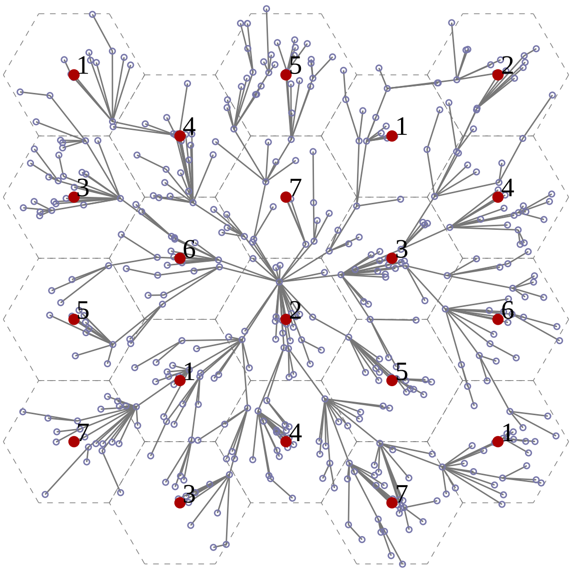


Figure 6: Example of a deployment.

in their timeslot. The timeslots can be reused in other clusters and this will ensure that no interference or minimal interference occurs between clusters.

In Figure 6 an example deployment is shown. The cluster-heads are in the center of their cluster and the numbers shown in the cluster indicate the used timeslot of the cluster. The small dots are sensor nodes scattered across the complete deployment area and the lines between nodes indicate communication links. Communication does not necessarily have to be done from or towards the cluster-heads and can be done to any node within the communication range. The links are set up in such a way that information is collected at a central sink.

The size of the clusters is dependant on the communication range of the nodes. We assume that all nodes in the network use the same output power for transmissions and will therefore have the same communication range. All nodes within the cluster should be able to communicate with the cluster-head, therefore the cluster size should not be bigger than the communication range. We assume the radius of the cluster is exactly the size of the maximum communication range. The actual size can be calculated using the path loss Equation (1).

The clusters in our approach are similar to cells in a cellular network. If we assume the shape of a cluster in our approach is hexagonal, we can then use the equations from cellular networks to calculate the number of timeslots required. The number of timeslots determines the reuse distance, one may recall that the reuse distance is the minimum distance between two clusters that share the same timeslot, see Figure 3 for an example.

The number of timeslots can not arbitrarly be chosen and is determined from the following formula:

$$N = i^2 + ij + j^2 \tag{4}$$

The i and j parameters determine the reuse distance of a timeslot along two axis. The reuse distance (D) can be calculated from the number of cells per cluster (N) and the cell radius (R):

$$D = R\sqrt{3N} \tag{5}$$

The reuse distance is the minimum distance between two interfering senders in the network. The larger the distance between two interferers, the less interference experienced during communication. If a total of 3 timeslots are used, the closest distance between two interfering nodes is exactly the radius of the cluster. If more timeslots are used, the distance between two interfering nodes will be larger, resulting in less noise from neighbouring clusters.

The nodes within a cluster all register their transmissions to the closest cluster-head. The cluster-head is therefore able to schedule all the transmissions within its cluster. After doing so, it will send the minimum length of its local schedule to the central cluster-head. The central cluster-head will assign timeslots to the clusters and determine the length of each timeslot. The timeslots do not necessarily have

	Cluster						
	1	2	3	4	5	6	Max
Slot 1	1.33			1.57			1.57
Slot 2		1.61			1.43		1.61
Slot 3			1.37			1.45	1.45
Slot length	1.57	1.61	1.45	1.57	1.61	1.45	

Figure 7: Results of calculating slot length based on cluster schedule lengths.

to be of equal time. The central cluster-head will assign the maximum schedule length of all clusters that share the same timeslot.

The cluster-heads will determine the order of transmissions within their cluster. This can be done using different optimization criteria as presented in (van Kleunen et al., 2011). We will be using the greedy approach in which transmissions are scheduled based on minimum delay.

For scheduling the transmissions within a cluster we can use the algorithm from (van Kleunen et al., 2011) or the reduced complexity algorithm from Section 4. The second algorithm will yield a smaller computational and memory space complexity, but because the number of transmissions per cluster is in practice limited, the first algorithm may as well be a good option.

Figure 7 shows an example of how the algorithm works. The table shows for all clusters the calculated cluster schedule lengths. The cluster-head schedules all transmissions within its cluster and determines the clusters schedule length. The central scheduler determines the maximum of all schedule lengths per slot and assigns the maximum schedule length to the slot. The cluster-heads and the central scheduler only need to communicate the schedule length and slot lengths.

6 EVALUATION OF COMMUNICATION AND COMPUTATION COMPLEXITY

We evaluate the different centralized and distributed scheduling approaches. We will discuss briefly their complexity in terms of number of communications required as well as computational complexity of different approaches. The complexity overview of all scheduling approaches can be seen in Figure 8.

- **Centralized Scheduling:** In this case we assume all transmission as well as position information is collected in a central location. The communication complexity is $n \cdot hops_{avg}$ (The average number of hops), because all transmission information needs to be sent over a multi-hop link to the central scheduler. For scheduling the links we will use the algorithm described in (van Kleunen et al., 2011), whose complexity is $O(n^3)$.
- **Reduced Complexity Centralized Scheduling:** This is the algorithm described in Section 4. The computational complexity of this algorithm is $O(\frac{1}{2}n^2)$. The communication complexity is the same as the other centralized scheduling approach, namely $O(n^3)$.
- **Distributed Scheduling:** In the distributed situation, the transmissions are sent only to the cluster-head ($O(n)$ communications). The cluster-head will calculate a schedule for its own cluster and will forward the length of its schedule over a multi-hop link to the central scheduler ($O(hops_{avg}k)$ communications). On average, the number of transmissions per cluster is n/k, which results in a computational complexity of $O((n/k)^3)$ per cluster, but also for the whole network.
- **Distributed Reduced Complexity Scheduling:** It is similar to the distributed approach, but the scheduling per cluster uses the reduced complexity centralized scheduling algorithm. This reduces the scheduling algorithm complexity to $O(\frac{1}{2}(n/k)^2)$ per cluster. The communication complexity remains $O(hops_{avg}k)$.

The packet size of all approaches is constant and does not grow with respect to the number of nodes in the network. From the evaluation of the complexity of the different approach, we can see that the distributed approaches have a much lower computational and communication overhead compared to the centralized approaches. The scalability of the distributed approaches is therefore much better than the centralized approaches.

7 EVALUATION OF SCHEDULING EFFICIENCY

To evaluate the scheduling efficiency of the different approaches, we implement them in c++. We evaluate the algorithms for different sizes of deployments. The parameters can be found in Figure 9(a). The network size ranges from 500 up to 8000 nodes scattered randomly over an area. The communications are set up in such a way that all data is collected at a central sink, similarly to the deployment in Figure 6.

For the different distributed scheduling approaches a reuse distance should be selected. We evaluated the distributed algorithms with both 3 as well as 7 timeslots.

Scheduling approach	Computational	Communication	Packet size
Centralized	$O(n^3)$	$2(n \cdot hops_{avg})$	$O(1)$
Reduced Complexity Centr.	$O(\frac{1}{2}n^2)$	$2(n \cdot hops_{avg})$	$O(1)$
Distributed	$O((n/k)^3)$	$2(n + k \cdot hops_{avg})$	$O(1)$
Distributed Reduced Complexity	$O(\frac{1}{2}(n/k)^2)$	$2(n + k \cdot hops_{avg})$	$O(1)$

n = Number of transmissions
k = Number of clusters

Figure 8: Complexity of different scheduling approaches compared.

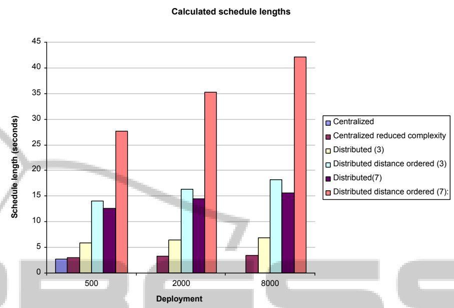
Parameter	Value
Communication range:	500m
Data rate:	1000bps
Propagation speed:	1500 m/s
Node placement:	random / uniform

(a) General parameters

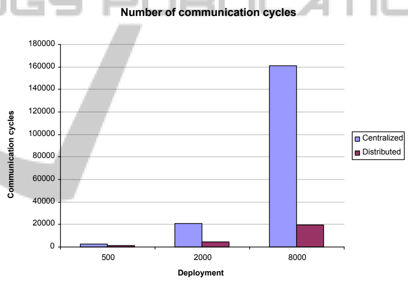
Parameter	Small	Medium	Large
Clusters:	4 x 3	7 x 7	14 x 14
Area size:	3.2 x 3.1km	5.5 x 6.6km	11 x 13km
Nodes:	500	2000	8000

(b) Different deployment sizes

Figure 9: Simulation parameters.



(a) Schedule length of different scheduling approaches



	Scheduling approach	
Network size	Centralized	Distributed
500	2813	1040
2000	21147	4418
8000	160878	19540

(b) Number of communication cycles required to setup schedule

Figure 10: Results of simulation for different deployments and scheduling approaches.

The evaluation results are shown in Figure 10(a). We see that the centralized approach performs the best, which is expected. This is due to the fact that the centralized approach has all link and deployment information of the network during the scheduling, while the distributed approach splits up the scheduling in sub-problems and uses local information only. The centralized approach places a lower bound on the achievable schedule length.

The reduced complexity centralized algorithm performs only slightly worse, the difference in schedule lengths is only marginal. Therefore the reduced complexity centralized algorithm is a good alternative to the full complexity centralized algorithm. In Section 4 and Section 6 we have already shown that the reduced complexity algorithm has large benefits in terms of computation and memory complexity. From the results of the simulation, we can conclude these benefits come at almost no cost in terms of schedule efficiency.

Among the distributed approaches, the distributed approach which minimizes schedule length and uses 3 timeslots, performs about twice as worse as the centralized approach. The approach that orders the transmissions based on distance of the transmission performs worse. The fact that the distributed approach performs worse when the network size increases is because for every timeslot the maximum schedule length from all clusters using that timeslot is used. If more clusters use the same timeslot, the maximum schedule length over all these clusters will go up.

In Figure 10(b) the amount of communications cycles required to set up the network are shown. The difference between the centralized and distributed approach can be seen quite clearly. The centralized approach does not scale very well to larger sizes and requires large number of communication cycles. The distributed approach grows almost linearly with the size of the network, the number of communication cycles required is a little over 2 times the number of nodes in the network. The packet size of the messages is independant of the number of nodes in the network as has been noted before and contains only position and transmission information, or total schedule length for the cluster heads.

8 CONCLUSIONS

Scheduling algorithms for underwater communication allows mitigating the effects of the long propagation delay of the acoustic signal. Scheduling has significant benefits in terms of throughput, energy consumption, and reliability.

In this paper we extended the set of simplified scheduling constraints of (van Kleunen et al., 2011) to allow scheduling of multi-hop networks. We have introduced a centralized and a distributed scheduling techniques for underwater acoustic communication systems.

The reduced complexity centralized approach has $O(\frac{1}{2}n^2)$ computational complexity but introduces significant communication overhead.

The distributed approach groups all transmissions together in clusters from which they originate. Nodes within a cluster communicate with the cluster-head only for scheduling their link. Our approach does not place any restrictions on the communication patterns. It does not restrict communication between base-station and node and nodes can communicate directly with other nodes within communication range.

Each cluster-head will calculate a schedule for its cluster and will forward the total schedule length of its cluster to a central scheduler. The central scheduler will schedule the timeslots and assign a timeslot to each cluster. Compared to the centralized approach, the distributed approach has a much lower communication and computational overhead.

We evaluated the communication and computational complexity and showed that the distributed approach is much more scalable towards larger networks. We also evaluated the schedule lengths of different scheduling approaches. The reduced complexity centralized approach calculates only marginally less efficient schedules, and is therefore a good replacement for the full complexity approach. The schedule lengths of the distributed approach are on average 270% of the centralized approach when 3 timeslots are used, and 580% when 7 timeslots are used. This shows that when the scalability, computational and communication benefits are irrelevant a centralized approach is still much preferred.

Future work includes considering the effects of acoustic signal such as refraction, multipath and propagation speed variability on performance. Other effects that will be considered are node dynamics, position estimation errors and time-synchronisation errors.

ACKNOWLEDGEMENTS

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

REFERENCES

Hsu, C.-C., Lai, K.-F., Chou, C.-F., and Lin, K. C.-J. (2009). St-mac: Spatial-temporal mac scheduling for underwater sensor networks. In *INFOCOM*, pages 1827–1835. IEEE.

Kredo II, K. and Mohapatra, P. (2010). Distributed scheduling and routing in underwater wireless networks. *Globecom 2010*.

Molins, M. (2006). Slotted fama: a mac protocol for underwater acoustic networks. In *In IEEE OCEANS06, Singapore*, pages 16–19.

Peleato, B. and Stojanovic, M. (2007). A channel sharing scheme for underwater cellular networks. In *OCEANS 2007 - Europe*, pages 1–5.

Stojanovic, M. (2008). Design and capacity analysis of cellular-type underwater acoustic networks. In *IEEE Journal of Oceanic Engineering, vol.33, No.2*, pages 171–181.

Syed, A. A., Ye, W., and Heidemann, J. (2007). T-lohi: A new class of mac protocols for underwater acoustic sensor networks. Technical report.

van Kleunen, W., Meratnia, N., and Havinga, P. J. (2011). A set of simplified scheduling constraints for underwater acoustic mac scheduling. *AINA 2011*.