2012 IEEE 15th International Conference on Computational Science and Engineering

MDS-Mac: A Scheduled MAC for Localization, Time-Synchronisation and Communication in Underwater Acoustic Networks

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*—In this paper we describe a design for an underwater MAC protocol which combines localization, time-synchronisation and communication. This protocol is designed for small-scale clustered networks in which all nodes are able to communicate with each other. We consider an integrated design of localization, time-synchronisation and communication important because scheduled communication, localization and time-synchronisation require estimation of propagation delays between nodes to operate accurately and efficiently. For localization we use multidimensional scaling because it does not require the use of reference nodes.**

Our MAC design consists of two phases, i.e. an unscheduled coordination phase and a communication phase. During the first phase, propagation delays are estimated, relative positions are calculated using multidimensional scaling, and time-synchronisation is performed. During the communication phase, sensor-data is transmitted using scheduled communication.

Using simulation we evaluate the feasibility of such a design. By measuring the time required for the coordination phase at different modulation rates, we derive the required modulation rate for acceptable coordination phase times.

I. Introduction

Underwater acoustic sensor networks can be used for monitoring the underwater world which have traditionaly been either invisible or was monitored with significant delays and low granularity.

Our focus is on small clusters of static underwater nodes scattered in a small area. Examples of applications these clusters can be used for are monitoring vibrations around a site for an underwater oil-drilling site or monitoring the entry of a harbour for security reasons. Our aim is to provide communication between nodes, position estimation of nodes and time-synchronisation simultaniously through designing an integrated MAC.

Existing work on time-synchronisation [1], localization [2] and communication [3] [4] consider these three aspects individualy and does not consider their impact on each other. For example papers proposing scheduled communication do not consider the localization and time-synchronisation aspects of a system, even though position estimation and time-synchronisation are required for the scheduling to work properly. Papers proposing time-synchronisation such as [1], require an estimate of the propagation delay between nodes, however do not consider their impact on communication. Related work tends to limit its focus rather than looking at complete MAC protocol designs.

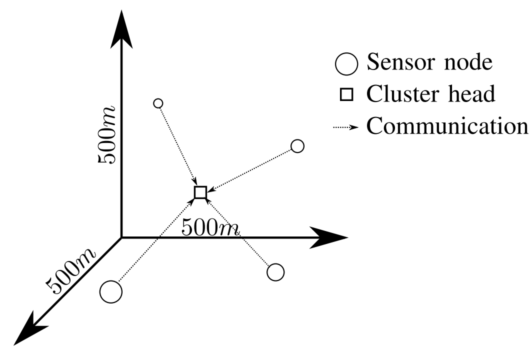
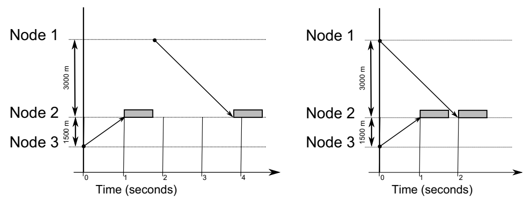


Fig. 1. Example of a 3d deployment of an underwater cluster

In this paper we propose a MAC protocol for combined time-synchronisation, localization and communication scheduling approach for small underwater clusters. Localization requires propagation-delays to estimate distances between nodes, communication scheduling requires the delays to calculate a collision-free schedule, and time-synchronisation requires the propagation delay to provide accurate results. Therefore it makes sense to combine these algorithms in a single MAC protocol and reuse propagation delay measurements to determine node position and to do time-synchronisation and communication scheduling. The MAC protocol that we propose provides means to measure the propagation delays and use this information to schedule communication, determine the position of nodes, and provide time-synchronisation.

Regarding localization, we provide relative position estimation. Our aim is to provide the localization service without requiring any setup phase for reference nodes. We see this as a significant benefit in terms of practical deployment of underwater sensor networks. Determining the positions of reference nodes under water can be difficult and requires the use of external positioning systems such as a Ultra Short Baseline (USBL) system attached to a boat or use of surface beacons equiped with GPS. When only relative positions are required,

978-0-7695-4914-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICCSE.2012.95



(a) Exclusive access (b) Scheduled

Fig. 2. Example of how scheduling can improve the throughput of underwater communication.

such as for communication scheduling, time-synchronisation or distributed beamforming, not having to configure reference nodes provides significant benefits in terms of easy, fast and cheaper deployment of such a network.

Our MAC protocol combines localization, time-synchronisation and communication in a way not done before. Traditionaly underwater MAC protocols have focused on a single aspect only. In this paper we will show a design for a MAC protocol which combines all three aspects for underwater networks. Also we consider the overhead required for scheduled communication. Before scheduled communication can be applied, the nodes in the network need to be synchronised and localized. Rather than assuming position and time-synchronisation to be given, we consider the overhead of an unscheduled communication phase required for determining these positions and time-synchronisation. We have evaluated this overhead in simulation and will show the required data-rates for an acceptable overhead.

Previously we have looked at scheduling communication in underwater acoustic networks to reduce collisions, improve throughput, and reduce energy consumption. This resulted in a simplified scheduling approach described in [3], a distributed scheduling approach described in [5], and a combined time-synchronisation and localization system described in [10]. Figure 2 illustrates an example on how spatial-temporal scheduling can be used to mitigate the effects of the low propagation speed of the acoustic signal and on how it improves the throughput of underwater communication.

Different from traditional scheduling approaches which require position information, in this paper we propose to use measured propagation delays between nodes for scheduling. This measured propagation delay is in practice enough to provide scheduled communication. These propagation delay measurements can then also be used to determine relative positions of nodes using multidimensional scaling (MDS) [6]. Using propagation delays measurements it will also be possible to provide time-synchronisation on the nodes which can be used for time-stamping measurements and also is required for scheduled communication approaches.

The rest of this paper is organised as follows: In Section II we describe the related work in the areas of localization, time-synchronisation and communication scheduling in underwater acoustic networks. Section III describes the proposed design of the MAC protocol followed by Section IV in which we evaluate the performance of the MAC protocol to finish the paper with the conclusion in Section V.

II. Related Work

Underwater localization has been traditionally done with acoustic signals. Propagation delay information and angular information can be used to estimate the position of the blind nodes. Using propagation delay information between blind nodes, relative positioning can be performed without requiring reference nodes with known position. This relative positioning can be done using multidimensional scaling. An example of such a system is shown in [2], which uses stress-majorization [7] to determine the relative positions of nodes in underwater acoustic networks. The article discusses some issues encountered when doing underwater localization and does some performance measurements on multidimensional scaling for underwater applications.

For time-synchronisation in underwater acoustic networks much work has been done already. An example of a time-synchronisation protocol which can be used is TSHL [1], which shows that for time-synchronisation also an estimation of the propagation delay between the reference node and the unsynchronised node is needed. This is because propagation delays are more significant in acoustic networks than in radio-based networks and therefore clock-skew has a more significant impact on the accuracy of the synchronisation accuracy.

TSHL is based on first estimating the clock-skew of a node and then calculating the clock-offset using a clock-skew corrected estimation of the propagation delay between two nodes. The same (clock-skew corrected) information on the propagation delay between two nodes is required by localization and communication scheduling approaches.

Communication scheduling has been researched also and has resulted in scheduling algorithms such as ST-MAC [4], Stump [8] and Simplified Scheduling approach [3]. These scheduling algorithms use position information of nodes to determine propagation delays, but in practice require only propagation delay information between nodes to function.

For the design of our MAC protocol we chose to use our Simplified Scheduling algorithm [3], although other scheduling algorithms can be used as well. The simplified scheduling uses scheduling constraints to determine the delay between two transmissions.

III. Design

By looking at the related work, it becomes evident that localization, time-synchronisation and communication scheduling all require information on propagation delays between nodes to function properly.

The operation of our designed protocol as shown in Figure 3 is divided in a coordination and communication phase. Initially the network is in the coordination phase, in which nodes do range measurement between themselves and all other nodes in the network. Once all propagation delays between all pairs of nodes are known, all nodes can run a multidimensional scaling algorithm to determine their own position and position of all other nodes. The nodes are also able to synchronise their clocks using the time-information from the other nodes.

The cluster-head will determine the routing and scheduling of the network. We assume the cluster-head is preconfigured and all nodes know the network identifier of the cluster-head. The cluster-head will also determine when the network switches from the coordination phase to the communication phase. This is done by broadcasting the schedule for the communication phase. All nodes within the network will follow the schedule for the communication period and will fallback to the coordination phase once the schedule has finished.

In the following sections we will look at how these two different phases are implemented, i.e. how coordination and scheduling is done and how information from the coordination phase can be used for relative localization and time-synchronisation.

A. Coordination phase

The coordination phase of the MAC protocol is all about determining the propagation delays between nodes and establishing time-synchronisation between nodes. The coordination phase of our MAC protocol is done using broadcast messages. Initialy the nodes are all unsynchronized and unaware of other nodes within the cluster. All nodes are assumed to have a preconfigured unique identifier and a preconfigured transmission interval. This phase is the initial coordination phase. Nodes will transmit broadcast message at a random time and will listen for broadcast messages from other nodes when not transmitting.

The content of the broadcast messages is described in Figure 4. The broadcast message contains the source address of the node and contains a processing delay allowing other nodes to do round-trip time measurements to the node. The round-trip is the time required to send a message to a node and for the node to send a message back. The time required to do a round-trip is dependant on the distance between two nodes and the propagation speed of the acoustic signal.

Figure 5 shows how round-trip time measurements can be done using the processing delay in the broadcast message. When node 3 sends a broadcast message, this message will be received by node 1. Node 1 will measure the time between receiving the broadcast message from node 1 and sending its own broadcast message. This information is stored in the processing delay field of the broadcast message. When node 3 receives the broadcast from node 1, it can measure the round-trip time and substract the processing delay on node 1. This information can be used to calculate the distance and the propagation delay of the acoustic signal between the two nodes.

For the protocol to properly function, a minimum initial coordination phase time is required. This time is dependent on several parameters, namely: the total number of nodes within the cluster, dimensions of the cluster and the packet error rate. The protocol also requires a sending interval to be used during the coordination phase. This interval is the average sending

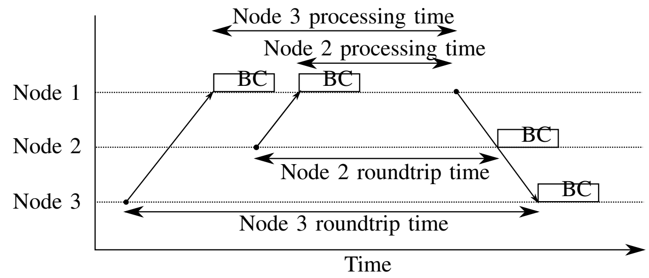


Fig. 5. Timeline for sending beacons and measuring propagation delays

interval at which the nodes send out broadcast messages during the coordination phase.

In the coordination phase, we use pure ALOHA and calculatethe optimal sending rate (G) can be calculated [9]. From the simulation resuls (Section IV) it becomes clear that the coordination phase can take a significant amount of time. Once the initial coordination is done, time required by the subsequent coordination phases can be shorter. If we reuse propagation delay measurements from previous coordination phases the next coordination phase will only update existing propagation delay measurements and therefore will be performed even faster.

Communication of sensor data is done during the communication phase. In the following subsections we will look at localization and time-synchronisation, done during the coordination phase, and communication done during the communication phase.

1) Localization: Once all pair-wise propagation delays are measured, relative localization can be done using multidimensional-scaling. We can calculate the distance between two nodes denoted as $\delta_{i,j}$ using the measured propagation delay denoted as $t_{i,j}$ and an estimat of the sound speed under water ($v_{sound} \approx 1500m/s$) as follows:

$$\delta_{i,j} = \frac{t_{i,j}}{2 * v_{sound}}$$

We use these distance estimates in the dissimilarity matrix δ expressed as:

$$\delta = \begin{pmatrix} \delta_{1,1} & \delta_{1,2} & ... & \delta_{1,i} \\ \delta_{2,1} & \delta_{2,2} & ... & \delta_{2,i} \\ ... & ... & ... & ... \\ \delta_{i,1} & \delta_{i,2} & ... & \delta_{i,i} \end{pmatrix}$$

The goal of multidimensional scaling is, using this dissimilarity matrix δ, to find the position vectors of the nodes $x_1, ..., x_i \in \mathbb{R}^3$. This can be done by minimizing the following cost function:

$$\min_{x_1,....,x_i} \sum_{i<j} w_{i,j}(\|x_i - x_j\| - \delta_{ij})^2 \qquad (1)$$

Where $\|x_i - x_j\|$ is the Euclidian distance between two vectors. This can be done using an iterative steepest descent approach as described by Kruskal [6]. However the iterative

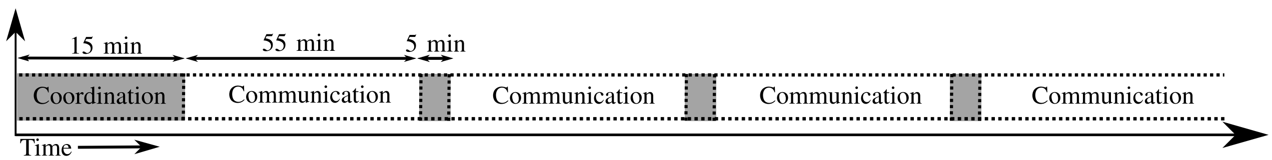


Fig. 3. Timeline of a network in operation

Field	Size	Description
$length$	1 byte	Length of packet
$type$	1 byte	Type of packet (coordinate)
$source$	1 byte	Unique identifier of source node
$time$	8 bytes	Time in μsec
$processing.delay[16]$	2 bytes	Delay between last received broadcast and sending of this broadcast
$measured.rtt[16]$	2 bytes	Measured round-trip time with neighbour node
Total	74 bytes	

Fig. 4. Information contained in round-trip time measurement broadcast message

majorization approach described by Van Leeuw [7] provides a significantly better approach in terms of guarantees on, and rate of, convergences. Therefore we chose to use the iterative majorization approach.

Using weights it is possible to calculate the position of nodes even when pair-wise distance measurements are unavailable. When a distance measurement is unavailable, we use 0 for its weight. It is also possible to use the inverse distance estimate of a link. This means that when the measurement from $\delta_{i,j}$ is unavailable, we use the measurement from $\delta_{j,i}$.

In theory it is possible to use multidimensional scaling with weights to estimate the missing propagation delays. This allows the cluster-head to do the scheduling faster, allowing the network to spend less time in the coordination phase and more in the communication phase.

2) Time-Synchronisation: Using the TSHL [1] approach it is possible to do time-synchronisation. Using the measured arrival times of the beacon messages from other nodes it is possible to get an estimation of the clock skew. Because we measure the propagation delay during the coordination phase, it is also possible to accurately determine the clock offset.

The time-synchronisation will be done during the coordination phase as well. By measuring the reception time of message of a neighbour, the clock-skew of the local node in relation with this neighbour can be estimated. By measuring the clock-skew in relation with all neighbours in the network, an average clock-skew can be calculated.

Because round-trip time information is available, the propagation delay of a packet can be estimated. This information can be used to determine the clock-offset at the local nodes. This is done by looking at the transmission time of the packet (which is stored in the broadcast packet) and using the propagation delay.

B. Communication

During the communication phase, the actual sensor data transmission is performed. Rather than using unscheduled communication, the communication during the communication phase is scheduled. The cluster-head is responsible for calculating the schedule and will broadcast the schedule to be used during the coordination phase. The content of a schedule broadcast message is shown in Figure 6.

Field	Size	Description
$length$	1 byte	Length of packet
$type$	1 byte	Type of packet (Schedule)
$time_{start}$	4 byte	Start time of schedule in seconds
$time_{end}$	4 byte	End time of schedule in seconds
$time_{interval}$	4 byte	Interval between schedule iterations in seconds
$total.links$	1 byte	Total number of links
$source[i]$	4 bits	Source node of link[i]
$dest[i]$	4 bits	Destination node of link[i]
$time[i]$	2 bytes	Time when the packet can be send in ms
Total	$14 + (total.links * 3)$ bytes	

Fig. 6. Information contained in schedule message

Before calculating a schedule, the cluster-head will first determine the routing in the network. For example in a data-collection network, the cluster-head will schedule all nodes to send their measurements to a gateway node. Other communication patterns can also be scheduled, according to the requirements of the application at hand.

In case the cluster-head can not make the routing decision, the cluster-head may schedule broadcast messages for all nodes. In this way, the node itself may make a decision on which node to address. In the schedule packet a broadcast message is denoted by setting the source and destination field of a link to the same value.

For scheduling the transmissions, the Simplified Scheduling algorithm is used. The difference with the existing scheduling algorithm is that rather than using the positions of nodes to estimate the distance between two nodes we can use the measured propagation delays. The function $T(src, dst)$ used

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

Fig. 8. Extended set of simplified scheduling constraints allowing broadcast scheduling.

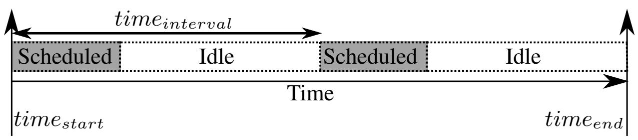


Fig. 7. Example of different time denoters in communication phase

by the simplified constraints from Figure 8 can be replaced by a lookup of the measured propagation delay. To estimate the propagation delay between node i and node j, the average of the measured propagation delay $t_{i,j}$ and $t_{j,i}$ is used.

For scheduling the transmissions, we use the algorithm from [5]. This algorithm is shown in Figure 9. What the algorithm does is simply keeping a minimum starting times for all transmissions. From all these possible starting times, the minimum time is selected and this transmission is to be scheduled next. The starting time of this transmission is now fixed and all unscheduled transmissions are updated. This is done by taking the maximum of the already calculated starting time and the minimum starting time between the transmission and the just scheduled transmission. This is repeated until all transmissions are scheduled.

In Figure III-B the time parameters of the schedule are explained. The communication schedule is repeated between $time_{start}$ and $time_{end}$, this is done with an interval $time_{interval}$. The schedule itself may be shorter than the interval period. This results in a idle period between different schedule runs. The time when a node can send its scheduled packet ($time[i]$ in the schedule) is calculated relative to the start of an interval.

In Figure 10 the content of a data message is described. The packet contains the identifier of the source and destination nodes. Again, the source and destination identifiers are the same if the message is a broadcast message. The data field contains the packet data.

The data messages described in Figure 10 are only sent on the scheduled time as calculated by the cluster head. The reception of the data message is therefore as well determined. A node may choose to listen only when scheduled data messages are expected and turn off its modem when it is not expecting any data, thus conserving energy.

$V \leftarrow transmissions$ {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

Fig. 9. Reduced complexity algorithm for scheduling transmissions.

Field	Size	Description
length	1 byte	Length of packet
type	1 byte	Type of packet (Data)
source	4 bits	Unique identifier of source node
destination	4 bits	Unique identifier of destination node
data	x bytes	Packet data
Total	$3 + x$ bytes	

Fig. 10. Information contained in data message

IV. Performance Evaluation

For the performance evaluation, we assume a network of 16 nodes randomly deployed on an area of 500mx500m. We assume nodes are able to communicate with any other node at the rate of 1000bps. In this scenario we run the MAC protocol and measure how long coordination phase takes, if we measure (1) all pairwise propagation delay or (ii) we require only one of the pairwise propagation delays (either $t_{i,j}$ or $t_{j,i}$). We also measure the time until the cluster-head detects all nodes in the network. The result are shown in Figure 11(a).

We have compared the approach of relative localization with an approach in which beacons are used. This approach is similar to the approach we have shown in [10]. In this scenario 5 out of 16 nodes in the cluster are preconfigured with a position and the other nodes will listen to these beacons to determine their position. Once they determine their position they will start broadcasting their position information as well. The information of the round-trip time measurements in the broadcast messages are replaced with the position in this scenario. The coordination phase finishes when the cluster-head collects the position information from all nodes within the network.

From the results we can see that using low modulation rates, it takes a considerable time before all nodes in the cluster are detected and all pair-wise measurements are performed. When a modulation rate of 5000 bps or higher is used, the setup times become acceptable. The number of bytes sent during the coordination phase is shown in Figure 11(b). The number of bytes sent is largely independent of the modulation rate.

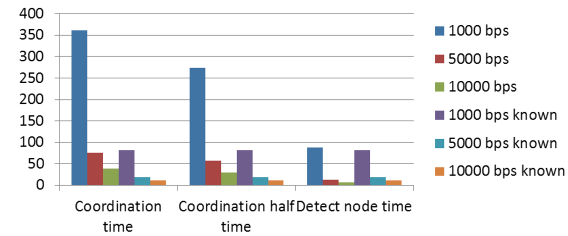
Comparing the relative scheduling approach to an approach in which preconfigured beacons are used, we can see that the approach using preconfigured beacons performs much better. For the lower data rates, it seems using preconfigured beacons is the only viable option, because the coordination times for round-trip measurements are too large. For the higher data rates, using preconfigured beacons still outperforms the completely relative localization scenarios. However not having to preconfigure beacons may be benificial in certain use cases. In scenarios in which beacons can not be preconfigured or when absolute coordinates are not required at the nodes, relative positioning provides a suitable approach for higher data rates.

Regarding the efficiency of the scheduled communication, we calculate two types of schedules for the communication phase: one with all nodes sending data to a single node (a data-collection network) and one in which all nodes are scheduled a broadcast message. The latter case allows the nodes themselves to do the routing decision. We see these two scenarios as typical use-case scenarios for our designed MAC protocol. The resulting schedule lengths are shown in Figure 12.

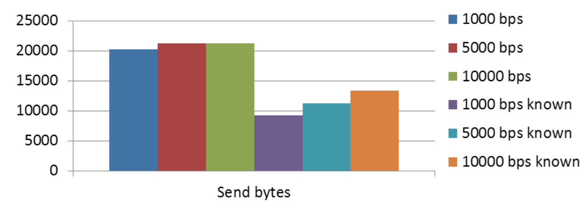
We also evaluate the performance of the multidimensional scaling localization approach. This performance is dependant on the range-estimation error. We model the range estimation error as a normal distribution $\mathcal{N}(0, \sigma^2)$ with the value σ^2 set to the values of 1% to 10% of the range. We use Matlab to calculate the positions of the nodes and compared them to the real positions of the nodes. We run this for 100 random deployments, the values of the absolute errors can be seen in Figure IV.

V. Conclusion

We have shown a design for a MAC protocol, which combines scheduled communication, localization and time-synchronisation. This MAC protocol is designed for small cluster with limited number of nodes. Because scheduled communication, localization and time-synchronisation require



(a) Time required for coordination



(b) Number of bytes send

Fig. 11. Measurement of coordination phase at different modulation rates

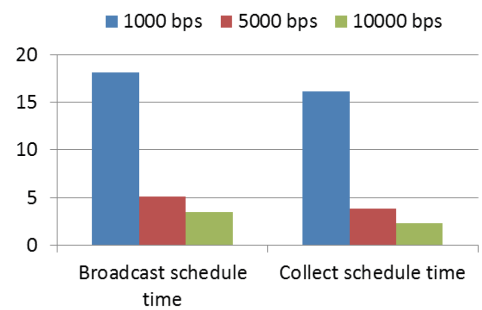


Fig. 12. Schedule lengths for all 16 nodes sending 128 bytes

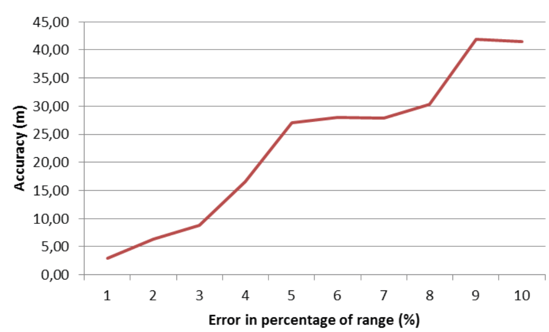


Fig. 13. Accurary of localization.

measurement of propagation delays, we consider an integrated MAC design to be important.

The operation of the MAC protocol is split into two phases: The first phase, the coordination phase, is used to measure the propagation delays between nodes. The second phase, the communication phase, is used to communicate sensor data.

After the coordination phase, all propagation delays between the nodes are known and a localization algorithm can be run. We applied multidimensional scaling to determine the relative positions of all nodes in the cluster. We also showed that time-synchronisation can be established after the coordination phase and a communication schedule for the communication phase can be calculated.

During the communication phase, nodes communicate using scheduled communication. Using the measured propagation delays we show how a communication scheduled for this phase can be calculated.

Our work is novel because it provides a combined MAC protocol for localization, time-synchronisation and communication. Traditional MAC protocols consider these three aspects separately. We have perfomed measurements on the two phases of the protocol. We have shown that to get reasonable setup times for the coordination phase an modulation rate of at least 5000bps is required.

We have compared the relative positioning approach to an approach where beacons with known positions are used. When beacons with known positions are used, the coordination phase can be done significantly quicker. For lower data rates (1000 bps) relative positioning does not seem to be a viable approach. In these scenarios beacons with known positions should be used to provide acceptable coordination phase times. For higher data rates relative positioning does provide acceptable coordination phase times. Using beacons with known positions still outperform relative positioning. However for certain applications setting up reference nodes may be too time consuming and require costly equipment for deployment. Also some applications may not require absolute position information on the nodes. For these types of applications, using relative positioning for underwater networks is a viable approach.

Acknowledgment

This work is supported by the SeaSTAR project funded by the Dutch Technology Foundation (STW) and the European Commission under the 7th Framework Programme (grant agreement no. 258359 CLAM).

References

[1] A. A. Syed and J. Heidemann, “Time synchronization for high latency acoustic networks,” in *In Proc. IEEE InfoCom*, 2006.

[2] K. Y. Foo and P. R. Atkins, “A relative-localization algorithm using incomplete pairwise distance measurements for underwater applications,” *EURASIP J. Adv. Signal Process*, vol. 2010, pp. 11:1–11:7, January 2010. [Online]. Available: http://dx.doi.org/10.1155/2010/930327

[3] W. A. P. van Kleunen, N. Meratnia, and P. J. M. Havinga, “A set of simplified scheduling constraints for underwater acoustic mac scheduling,” in *The 3rd International Workshop on Underwater Networks (WUnderNet-2011), Singapore*. Singapore: IEEE Computer Society, March 2011, pp. 902–907.

[4] C.-C. Hsu, K.-F. Lai, C.-F. Chou, and K. C.-J. Lin, “St-mac: Spatial-temporal mac scheduling for underwater sensor networks.” in *INFOCOM*. IEEE, 2009, pp. 1827–1835. [Online]. Available: http://dblp.uni-trier.de/db/conf/infocom/infocom2009.html#HsuLCL09

[5] W. A. P. van Kleunen, N. Meratnia, and P. J. M. Havinga, “Mac scheduling in large-scale underwater acoustic networks,” in *8th International Joint Conference on e-Business and Telecommunication, ICETE 2011, Sevilla, Spain*. USA: IEEE Computer Society, July 2011, pp. 27–34.

[6] J. Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis,” *Psychometrika*, vol. 29, pp. 1–27, 1964, 10.1007/BF02289565. [Online]. Available: http://dx.doi.org/10.1007/BF02289565

[7] J. de Leeuw, “Applications of convex analysis to multidimensional scaling,” in *Recent Developments in Statistics*, J. Barra, F. Brodeau, G. Romier, and B. V. Cutsem, Eds. Amsterdam: North Holland Publishing Company, 1977, pp. 133–146.

[8] K. Kredo II and P. Mohapatra, “Distributed scheduling and routing in underwater wireless networks,” *Globecom 2010*, 2010.

[9] A. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall Professional Technical Reference, 2002.

[10] W. A. P. van Kleunen, N. Meratnia, and P. J. M. Havinga, “Scheduled mac in beacon overlay networks for underwater localization and time-synchronization,” in *The Sixth ACM International Workshop on Underwater Networks (WUWNet) , Seattle, USA*. New York: ACM, December 2011, p. 6.