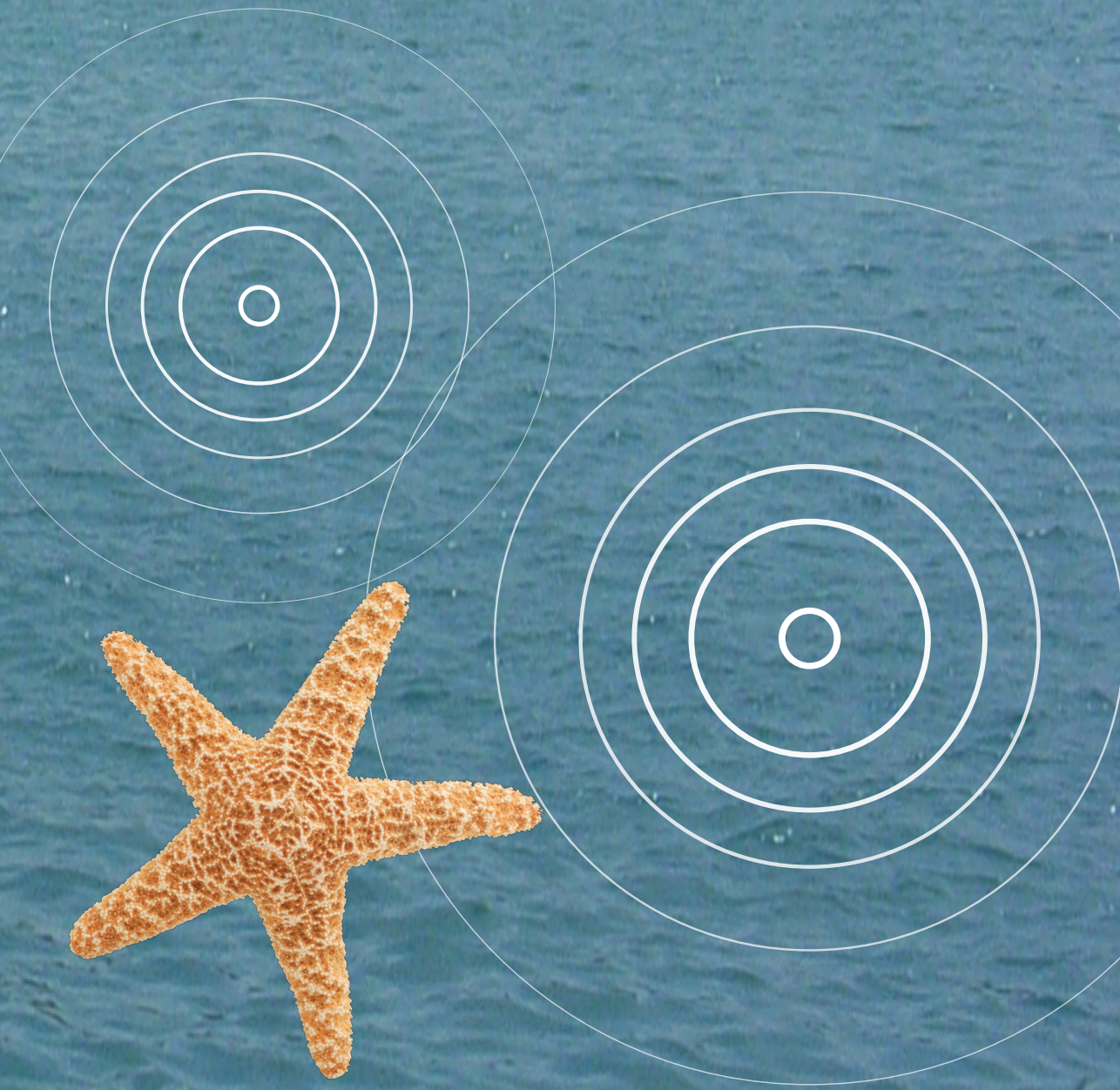
Echoes from the deep

Communication Scheduling, Localization and Time-Synchronization in Underwater Acoustic Sensor Networks



Wouter A.P. van Kleunen

Echoes from the deep

Communication Scheduling, Localization and Time-Synchronization in Underwater Acoustic Sensor Networks

Wouter A.P. van Kleunen

Graduation committee:

Chairman:	Prof.dr.ir. J.H.A. de Smit
Promoter:	Prof.dr.ing. P.J.M. Havinga
Assistant promoter:	Dr.ir. N. Meratnia

Members:

Prof.dr.ir. G.J. M. Smit	University of Twente
Prof.dr.ir. B. Nauta	University of Twente
Prof.dr.ir. A-J. van der Veen	Delft University of Technology
Dr. H.S. Dol	TNO
MSc. K.H. Grythe	Sintef

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

CTIT

CTIT Ph.D.-thesis Series No. 14-304
Centre for Telematics and Information Technology
University of Twente
P.O. Box 217, NL – 7500 AE Enschede

ISSN 1381-3617
ISBN 978-90-365-3662-2

Publisher: Gildeprint, Enschede
Cover design: Chantal Post

COMMUNICATION SCHEDULING, LOCALIZATION AND TIME-SYNCHRONIZATION IN UNDERWATER ACOUSTIC SENSOR NETWORKS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
Prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties,
in het openbaar te verdedigen
op woensdag 28 mei 2014 om 14.45 uur

door

Wouter Anne Pieter van Kleunen

geboren op 4 maart 1982
te Gouda, Nederland

Dit proefschrift is goedgekeurd door:
Prof. dr. ing.(promotor): Paul J.M. Havinga
Dr. ir. (assistent-promotor): Nirvana Meratnia

Abstract

Wireless Sensor Networks (WSNs) caused a shift in the way things are monitored. While traditional monitoring was coarse-grained and offline, using WSNs allows fine-grained and real-time monitoring. While radio-based WSNs are growing out of the stage of research to commercialization and widespread adoption, commercial underwater monitoring is still in the stage of coarse-grained and offline monitoring and research on Underwater Acoustic Sensor Networks (UASNs) is in the early stage.

Existing WSN research can only partially be applied to underwater communication and realization of large-scale mesh networks of underwater nodes requires rethinking of communication and networking protocols.

Acoustic communication is the most widely used type of communication for underwater networks. This is because acoustic communication is the only form of communication which allows long-range communication in underwater environments. Acoustic communication, however, poses its own set of challenges for the design of networking and communication protocols. The slow acoustic propagation speed of about $1500\,{}^{m}/_{s}$, limited available bandwidth, high transmission energy costs and variations in channel propagation are some of the challenges to overcome.

Existing Medium Access Control (MAC) protocols for underwater communication 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. We will show that an integrated approach has significant advantages over three separate solutions. We have developed a collision-free MAC protocol that provides both time-synchronization and localization in an energy-efficient and scalable way and with high throughput.

In this thesis we introduce a communication scheduling algorithm which we call Simplified Scheduling. A distributed scheduling approach reduces the computational and communication complexity of this scheduling algorithm to allow scheduling of large-scale networks.

We introduce a combined Time-of-Flight (ToF) and Direction-of-Arrival (DoA) localization and time-synchronization approach for non-cooperative networks, and introduce a cooperative combined localization and time-synchronization algorithm called aLS-Coop-Loc for cooperative networks. By combining localization and time-synchronization the communication overhead is reduced compared to separate solutions.

We show two examples of MAC protocols which combine the introduced scheduling and localization and time-synchronization techniques. In future work we will use these algorithms to design other efficient underwater MAC protocols which combine communication, localization and time-synchronization.

Samenvatting

Draadloze sensor netwerken veroorzaakte een verschuiving in hoe dingen gemonitord worden. Hoewel traditionele monitoring grofmazig en offline was, maken draadloze sensor netwerken fijnmazige en real-time monitoring mogelijk. Hoewel radio gebaseerde draadloze sensor netwerken uit het stadium van onderzoek zijn gegroeid, naar commercialisering en wijdverspreide adoptie, is commerciële onderwater monitoring nog in het stadium van grofmazige en offline monitoring en is onderzoek naar onderwater draadloze sensor netwerken nog in het vroege stadium.

Bestaande draadloze sensor netwerk onderzoek kan slecht gedeeltelijk worden toegepast op onderwater communicatie en de realisatie grootschalige mesh netwerken van onderwater nodes vereist heroverweging van communicatie en netwerk protocollen.

Akoestische communicatie is de meest wijdverspreide soort van communicatie voor onderwater netwerken. Dit is omdat akoestische communicatie de enige vorm van communicatie is die lange afstand communicatie mogelijk maakt in onderwater omgevingen. Akoestische communicatie, echter, brengt zo zijn eigen set van uitdagingen voor het ontwerp van netwerken en communicatie protocollen met zich mee. De langzame akoestische propagatie snelheid van ongeveer 1500 $^{m}/_{s}$, beperkt beschikbare bandbreedte, hoge transmissie-energie kosten en variaties in kanaal propagatie zijn enkele van de uitdagingen die moeten worden overwonnen.

Bestaande MAC protocollen voor onderwater communicatie nemen enkel datacommunicatie in acht, er is echter een noodzaak voor betrouwbare netwerk protocollen die niet alleen datacommunicatie maar ook positiebepaling en tijdsynchronisatie aanbieden. Wij laten een geïntegreerde aanpak zien die significante voordelen heeft over drie losstaande oplossingen. Wij hebben een MAC protocol ontwikkeld dat vrij is van collisies, dat zowel tijdsynchronisatie als positiebepaling aanbiedt op een energie-efficiënte en schaalbare wijze en met hoge doorvoersnelheden.

In dit proefschrift introduceren we een algoritme voor communicatie-scheduling die wij Simplified Scheduling noemen. Een gedistribueerde scheduling aanpak reduceert de computationele en communicatie complexiteit van dit scheduling algoritme en maakt scheduling van grootschalige netwerken mogelijk.

We introduceren een gecombineerde ToF en DoA positiebepaling en tijdsynchronisatie aanpak voor niet-coöperatieve netwerken en introduceren een coöperatieve gecombineerde positiebepaling en tijdsynchronisatie algoritme genaamd aLS-CoopLoc voor coöperatieve netwerken. Door het combineren van positiebepaling en tijdsynchronisatie wordt de communicatie overhead gereduceerde t.o.v. drie aparte oplossingen.

We laten twee voorbeelden van MAC protocollen zien die de geïntroduceerde schedulingen en positiebepaling en tijdsynchronisatie technieken combineren. In

toekomstig werk zullen wij deze algoritmen gebruiken om efficiënte onderwater protocollen te ontwerpen die communicatie, localisatie en tijdsynchronisatie combineren.

Acknowledgements

It was in 2006 when I started my master assignment and this whole journey into WSNs for me started. Before this I had never heard of WSNs. It seemed really appealing to me, programming small nodes with limited memory, battery powered and using low bandwidth radio to form large-scale networks. It sounded challenging and I do like a challenge.

Working the next couple of years at Ambient Systems on WSNs was really interesting, sometimes a bit frustrating, but I learned a lot. I especially enjoyed working together with Tjerk Hofmeijer, but also enjoyed working together with all the other colleagues, Mark, Ewald, Leon, Lodewijk, Eugen, Linda, Dennis, Arthur.

All this (working on WSNs and Ambient Systems) I owe to Paul Havinga (and the Pervasive Systems group). But things on underwater communication didn't really get started until ISSNIP 2009. I remember Paul his presentation and remember a mention of underwater sensor networks, which was just one of the many projects within the Pervasive Systems group. It really triggered me, underwater networks, how cool is that ? While being reluctant to do a PhD before, this pushed me to ask Paul if he thinks I am capable of doing a PhD and if he still had a position available on the underwater monitoring project. Luckily the position was still available. 4 years later I am here writing this thesis and I can only say: thank you Paul (and the Pervasive Systems group) for giving me this opportunity.

During my PhD many people have helped me which I would like to thank for their support. There is Nirvana Meratnia, who helped me structuring my ideas and learned me to write papers and become a scientist. My fellow PhD students within the SeaSTAR project Koen Blom and Saifullah Amir. Kyle Zhang, for being my roommate, making dinner and surviving in Norway, Bram Dil for his pointers helping me through the immense pile of available localization research and all the other colleagues of the Pervasive Systems group. Marck Smit and Tuncay Akal. Niels Moseley for making the SeaSTAR node (Appendix A.2), without his input I would have not been able to do the tests.

For the tests also many people were involved, helping out in all sort of ways or just getting sunburned. Kenneth Rovers, Guus Wijlens and the people from het Rutbeek. Again Koen, the development of the testbed, beamforming and much input from him has gone in Chapter 5. The off-shore test in Strindfjorden was supported and done in collaboration with the CLAM project (grant agreement no. 258359).

I would also like to thank my parents and my sister for their support and my wife for all the cupcakes, sandwiches, beers, meals and all love, affection and support.

Contents

List of Acronyms

AUV Autonomous Underwater Vehicle. 11, 23, 26, 27, 30, 131

CDMA Code Division Multiple Access. 23, 30

COTS Commercial Of-the-Shelf. 19, 87, 104, 132, 139

CSMA Carrier Sense Multiple Access. 132

DoA Direction-of-Arrival. 1, 3, 19, 27, 28, 30, 73–78, 80, 82–84, 109, 110, 112, 113, 118, 131, 133

DSSS Direct-Sequence Spread Spectrum. 139

DTN Delay Tolerant Network. 11

FDMA Frequency Division Multiple Access. 23, 30

FSK Frequency-Shift Keying. 140, 141

GPS Global Positioning System. 16, 88–90, 96–98, 100, 110, 120

LBL Long Baseline. 27

LMA Levenberg Marquardt Algorithm. 92, 143

LoS Line-of-Sight. 79, 80, 103

MAC Medium Access Control. 1, 3, 11, 15–19, 21–24, 29, 30, 37, 109, 110, 113, 114, 118–122, 125–127, 131–134

MDS Multi-Dimensional Scaling. 29, 88–94

MLE Maximum Likelihood Estimation. 76

PSK Phase-Shift Keying. 140

SBL Short Baseline. 27

SSBL Super Short Base Line. 101, 102

TDMA Time Division Multiple Access. 16, 17, 23

ToA Time-of-Arrival. 73, 89, 103, 132, 134, 139, 141, 142

ToF Time-of-Flight. 1, 3, 19, 27, 28, 30, 73–80, 82–84, 90, 91, 109, 110, 112, 113, 115, 118, 131–133

UAN Underwater Acoustic Network. 16, 26, 134

UASN Underwater Acoustic Sensor Network. 1, 11, 12, 15–18, 20, 22–24, 26, 27, 37, 38, 54, 77, 87, 88, 103, 120, 131–134, 139

USBL Ultra Short Baseline. 27, 120

WSN Wireless Sensor Network. 1, 5, 17, 27

CHAPTER 1

Introduction

Acoustic communication is the most widely used type of communication for underwater networks. This is because acoustic communication is the only form of communication which allows long-range communication in underwater environments. Acoustic communication, however, poses its own set of challenges for the design of networking and communication protocols. The slow acoustic propagation speed of about 1500 $^{m}/_{s}$, limited available bandwidth[1], high transmission energy costs[2], and variations in channel propagation are some of the challenges to overcome.

Existing underwater monitoring solutions are expensive and not energy-efficient. To allow fine-grained and real-time monitoring, the cost of hardware should be reduced, and the energy-efficiency of underwater communication, localization and time-synchronization should be improved. In this thesis we address the challenges of providing efficient communication, localization and time-synchronization in underwater acoustic MAC. Existing MAC protocols for underwater communication 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.

1.1 Applications of UASN

In [1] the applications of UASNs are classified based on the size of the deployment area and the density of the nodes within this area. This classification is shown in Figure 1.1. In the top-left of the figure we see applications of small number of nodes in large coverage areas. At current date this is the most widely used application of UASNs. Underwater sensors are usually stand-alone sensors using logging which are deployed and retrieved after a certain measurement period (days, months, a year) and the log file is read out. This provides fine-grained and offline monitoring.

Using Autonomous Underwater Vehicles (AUVs) it is possible to monitor large coverage areas by moving across the area. In such networks where multiple AUVs are used, infrequent connections between AUVs or an AUV and a base-station arise allowing flushing of logged data. Such a network is called a Disruption Tolerant Network or Delay Tolerant Network (DTN). This approach allow coarse-grained monitoring of large areas, however monitoring is done with large delays.

[1] Realistic data rates range between a few bits per second (long range, >1000 km) up to 10 kilobits per second (short range, <1 km)

[2] Transmission powers ranging from several watts to 50 watt

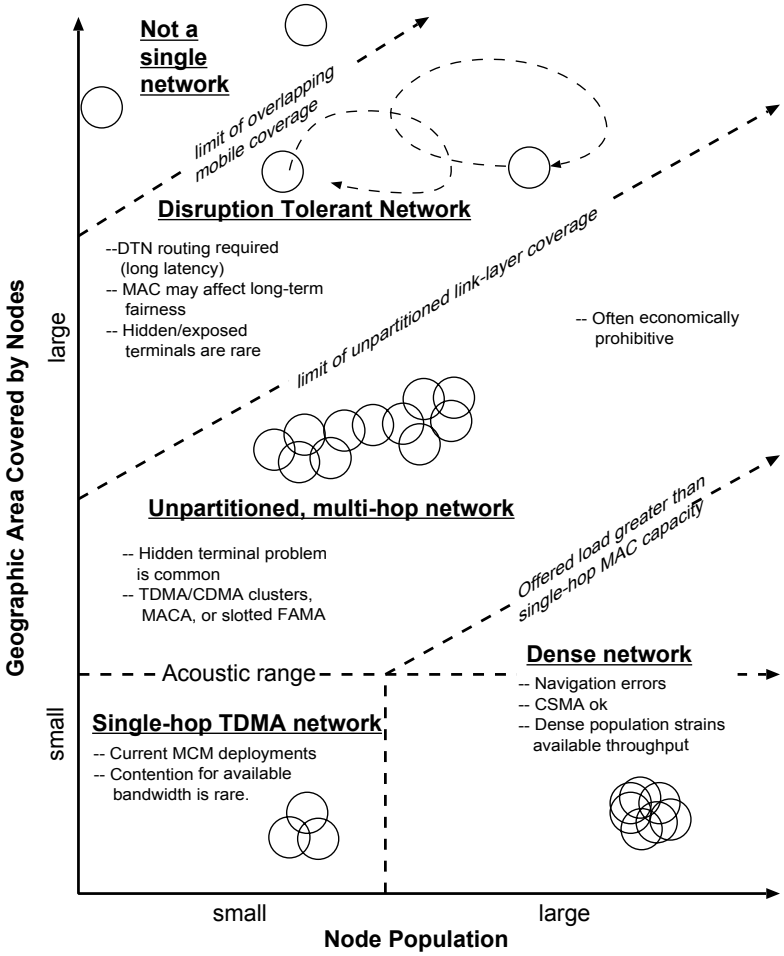


Figure 1.1: Classification of UASNs based on deployment size and node density [1]. In the top-left we see a small number of nodes deployed in a large coverage areas. Such networks have infrequent or no connection between nodes at all. They consists of data loggers or nodes running communication protocols which are disruption tolerant. When coverage area is smaller and node density is increased, multi-hop networks having continuous connection with neighbor nodes can be formed. These networks allow real-time and fine-grained monitoring.

On the other hand there is a demand for real-time and fine-grained monitoring on smaller coverage areas. These applications we see on the bottom of Figure 1.1. Because a large number of nodes are deployed in a relatively small area, it is possible to have fine-grained and real-time monitoring. Connections between nodes are more or less always available and sensor data can be sent to a base-station over a multi-hop connection in a real-time way.

Many future applications could benefits from such a monitoring network, examples of applications are environmental monitoring, pipeline and underwater drilling

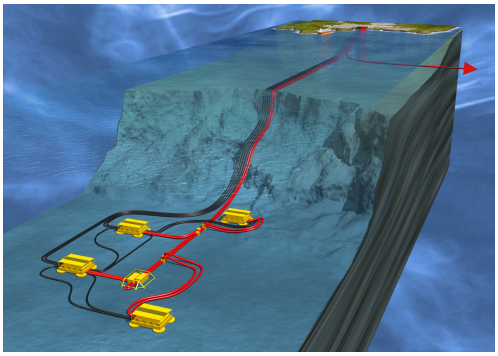


Figure 1.2: Subsea gas installation in the Storegga landslide area. [2]

monitoring and safety and security application. Below we will show some typical setups of these application domains. It is clear that setup and requirements differ significantly.

- **Pipeline monitoring.** Pipeline monitoring plays a crucial role in the prevention and detection of leaks in underwater pipelines. Pressure, corrosion or vibration sensors can be used to distinguish sections of pipeline prone to leaking. Furthermore, sensors that detect the presence of oil in water can be attached to the underwater nodes.

  Underwater pipelines can be very long, the longest underwater pipeline today stretches a length of 1200 kilometer [3]. The longest pipeline originates at a gas-field located in the Storegga landslide area. Its sub-sea gas installation can be seen in Figure 1.2. Envisioned is an application where sensor nodes are placed every 100 meter along the pipeline. A sketch of an Underwater Acoustic Sensor Network (UASN) employed for pipeline monitoring can be seen in Figure 1.3. Nodes use short range communication to send data to neighbor nodes along the pipeline. Data is forwarded to a gateway node which forwards its data to a surface bouy. This surface bouy can use a radio link to the shore to collect the data of all nodes on the pipeline on a central location.

- **Oil and gas exploration.** Oil and gas exploration requires advanced monitoring systems to prevent and identify possible problems. For extraction site monitoring the sensor nodes are placed close to the extraction site. This can be done by submerging nodes from a ship [4]. For deep water monitoring this results in a random deployment on the sea floor. Together, the nodes form a cluster, as shown in Figure 1.4.

- **Environmental monitoring.** Environmental monitoring can be categorized in pollution monitoring, ocean current and wind monitoring [5]. An improved understanding of oceans currents and wind improves weather forecasts. Another application is biological monitoring, such as monitoring of marine ecosystems. In environmental monitoring applications nodes can be placed in small-scale

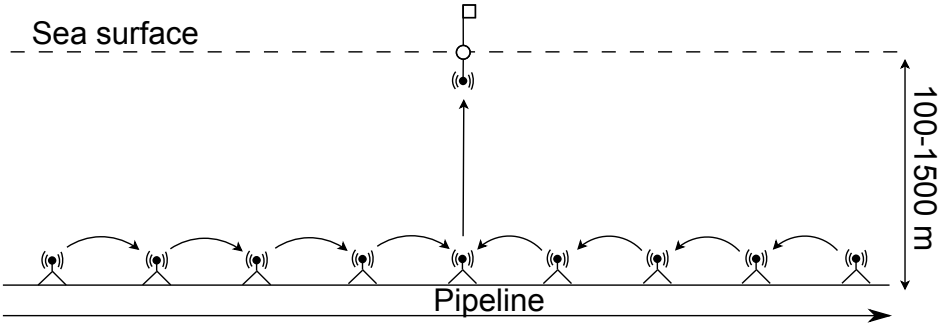


Figure 1.3: UASN for pipeline monitoring. Data is forwarded over a multi-hop communication to a gateway node node on the surface. Surface node has a radio-link to the shore to collect all the data from all sensors attached to the pipeline at a central location.

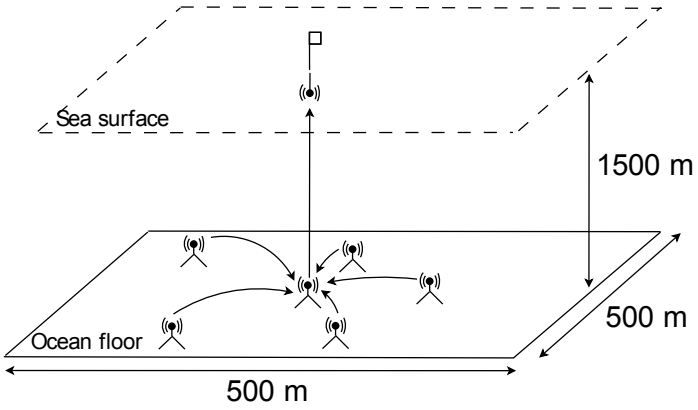


Figure 1.4: UASN for monitoring oil or gas extraction sites. A small cluster is formed at the ocean floor monitoring for example vibration. Efficient communication can be perfomed using communication scheduling. Time-synchronization is required for timestamping sensor measurements and using scheduled communication. Using a localization algorithm the positions of the nodes on the ocean floor is determined.

cluster to monitor small sites or stretch large areas such as monitoring coral-reefs.

- **Safety and security monitoring.** Safety and security monitoring applications are e.g. monitoring of ship and submarine movement in harbors. Deployment can be done in small-clusters such as at the entrance of a harbor or large-scale networks monitoring shipping activities an a lake or large area of the sea. Shipping activity can be detected and individual ships can be tracked by monitoring the sound of the ships.

1.2 MAC design constraints and limitations

To build static dense multi-hop underwater networks as has described in Section 1.1 it is not enough to provide only data communication. In these applications nodes are deployed for a long time underwater, the network should also provide time-synchronization to compensate for the clock drift in the nodes to allow accurate time-stamping of measurements. Moreover, because many nodes are deployed, it would be beneficial that nodes are able to determine their positions autonomously. It would be impractical to determine the positions of all the nodes manually, requiring costly and time-consuming deployment operations. An UASN should therefore provide the services of communication, localization and time-synchronization, and when designing we should consider the following aspects:

- **Energy-efficiency.** UASNs are generally not easy to deploy, therefore once nodes are deployed they should remain running for extensive periods (years) on batteries and possibly harvest their own energy. This requires energy-efficient MAC protocols.
- **Throughput.** Because the acoustic bandwidth is already very limited, a MAC protocol should use minimal overhead and provide maximum throughput possible.
- **End-to-end delay.** The end-to-end delay between the sensor producing the sensor data and the central gateway should be as small as possible to allow real-time monitoring of the environment.
- **Scalability.** Algorithms and protocols should scale to large number of nodes to allow fine-grained monitoring. This requires low computational complexity and low communication overhead.
- **Deployment cost.** To allow realization of networks with large number of nodes, the cost of such a network should be reduced. This requires reducing the cost of individual nodes but also reduction of the cost of deployment of such a network. The cost of nodes can be reduced by reducing the energy-consumption of nodes and thereby reducing the capacity of batteries required. The cost of deployment can be reduced by allowing the nodes to be self-organizing. Deployment of UASNs requires availability of ships, expensive equipment and personnel. If nodes are able to determine their position, time-synchronization and routing themselves, deployment time can be significantly reduced.
- **Autonomous.** Because nodes are deployed for extensive periods, nodes should provide autonomous localization and time-synchronization. Nodes should not require manual intervention in their operation, because this significantly increases cost.

Existing MAC protocols for UASNs generally consider the communication aspect only. As indicated in Section 1.1, the UASNs we are targeting require more than data communication only and in particular they also need localization and time-synchronization.

Existing work on these aspects (communication, localization and time-synchronization) however generally consider these aspects separately. To allow cost effective and energy-efficient deployment of networks for long-term underwater monitoring, it is needed to consider the performance of the whole system rather than considering aspects separately.

To perform more efficient underwater communication, MAC protocols exploit time-synchronization between nodes and use estimation of the propagation delay of transmissions. An example of an underwater MAC protocol which exploits both time-synchronization and propagation delay estimations is ST-MAC [6]. ST-MAC is a scheduled communication protocol which improves upon the basic Time Division Multiple Access (TDMA) scheduling by exploiting propagation delays. Scheduled communication is generally considered the most energy-efficient form of communication because it prevents energy wasting collisions from occuring. To perform scheduled communication, however, the positions of the nodes in the network (or propagation delays between nodes) should be known and time-synchronization is required. Hence, scheduled communication requires localization and time-synchronization.

To perform efficient time-synchronization, the propagation delays between nodes or the position of nodes should be known. To perform efficient localization, the nodes should be time-synchronized. In other words, efficient time-synchronization requires localization and efficient localization requires time-synchronization. It is therefore only possible to perform efficient localization and time-synchronization by combining them rather than looking at separate solutions. An example of such a localization approach is Global Positioning System (GPS) [7].

This leads us to the hypothesis of this thesis.

1.3 Hypothesis

The hypothesis of this thesis is as follows:

An integrated approach to Underwater Acoustic Sensor Network MAC protocols, combining localization, time-synchronization and communication has significant benefits over three separate solutions.

While answering the research questions we therefore not consider these aspect separately but rather consider the impact of each solution onto the other aspects of an integrated MAC protocol.

1.4 Research objectives

This research aims to overcome some of the challenges of acoustic communication and specifically focuses on challenges posed to MAC protocols for UASNs. The slow propagation speed and limited available bandwidth are problems we address using scheduled communication. Next to the problems imposed by the acoustic channel, UASN architectures impose their own set of challenges to overcome. Challenges such

as scalability to large number of nodes and a need for energy-efficient protocols to allow nodes to run on batteries.

Having sensor measurements without knowing where and when these measurements are taken is useless. This is true for both monitoring on air and underwater. Therefore localization and time synchronization play important role in WSNs. Traditionally localization and time-synchronization underwater have been performed separately. We argue and show a combined solution of communication, localization and time-synchronization is favorable in terms of energy-efficiency and scalability. This work focuses on development of algorithms to allow combined communication, localization and time-synchronization MAC protocols.

The overall research question of this work is:

How can communication, localization and time-synchronization be combined into an energy-efficient, reliable and scalable MAC protocol.

We attempt to combine communication, localization and time-synchronization to provide an integrated MAC for fine-grained and real-time multi-hop UASNs. We aim at providing a scheduled communication protocol, because scheduled communication is generally considered the most efficient and reliable way of communication. For doing scheduled communication an estimate of the position of the node and time-synchronization is required. Envisioned is a network running autonomously and for months to years on a single battery. Communication and localization should therefore be energy-efficient. Networks can range from small-scale cluster to large-scale networks with large number of nodes. The designed MAC protocols should therefore scale from a small number of nodes to large numbers of nodes in a single network. To answer the research question, we split up the work into two questions:

- **How can energy-efficient, scalable and reliable communication scheduling be performed in small-scale and large-scale UASNs.** Scheduled communication is generally considered the most efficient and reliable way of communication. However, existing scheduling algorithms such as TDMA do not consider the non-negligible propagation delay of the acoustic signal and are therefore suboptimal. Because of this propagation delay, underwater acoustic communication scheduling is non-trivial. In this work we strive to develop a simple scheduling algorithm for underwater communication, which is scalable, reliable, efficient in both setup overhead as well as run-time throughput and generally simple and easy to understand and implement.

- **How can localization and time-synchronization be performed in a energy-efficient, scalable and practical way in small-scale and large-scale UASNs.** Because of the non-negligible propagation delay of the acoustic signal, we consider localization (or dynamic positioning) and time-synchronization a combined problem. Existing time-synchronization protocols generally consider the propagation delay negligible. If the propagation delay is non-negligible, which is the case in acoustic networks, the delay needs to be estimated (inducing a significant communication overhead) or the position of nodes should be known (which is considered impractical because it requires an external positioning system). We consider a combined localization and time-synchronization more

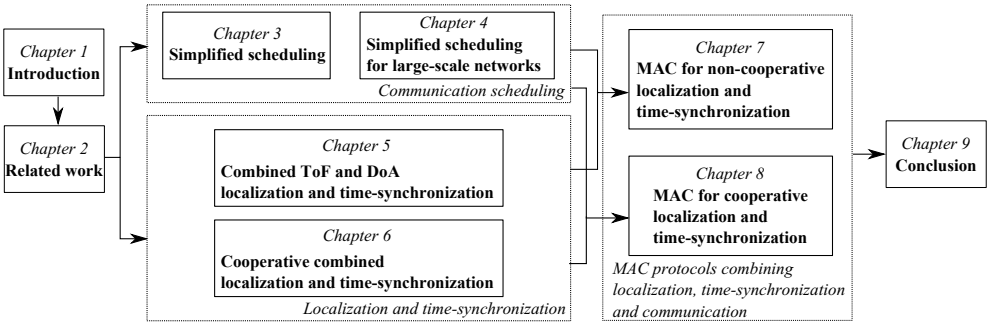


Figure 1.5: Outline of this thesis. First the related work is discussed, then our simplified scheduling and localization and time-synchronization approaches are introduced. Chapter 7 and Chapter 8 combine the proposed communication scheduling and localization approaches in different MAC protocols. We conclude this thesis with Chapter 9.

energy-efficient, more accurate and therefore favorable. We strive to find or develop combined localization and time-synchronization algorithms for both cooperative as well as non-cooperative networks. Moreover one-way ranging is preferred over two-way ranging, because it decreases the power consumption and increases the scalability of the approach. We therefore look at localization and time-synchronization approaches which use one-way ranging only.

To evaluate the performance of the proposed and existing algorithms we simulate the different solutions. We also strive to evaluate the performance of the algorithms in a real-world test setup to get results closer matching reality.

1.5 Thesis contributions

Figure 1.5 shows the outline of this thesis. Related work on communication, localization and time-synchronization in UASNs is discussed in Chapter 2. The contributions of this thesis are as follows:

- **A set of simplified scheduling constraints for underwater communication scheduling.** Existing scheduling approaches are sub-optimal, because of the use of timeslots, and generally difficult and cumbersome to use. Therefore in Chapter 3 we look at how to simplify the underwater scheduling by deriving a set of simplified scheduling constraints and show how these can be used to derive a simple scheduling algorithm. Also we show our unslotted scheduling approach outperforms existing slotted scheduling approaches.
- **A distributed approach to communication scheduling for large-scale networks.** In Chapter 4 we look at how scheduling can be performed in a distributed approach and perform a more extensive evaluation of centralized and distributed scheduling in terms of communication and computation complexity and efficiency of the calculated schedules. A distributed scheduling is required to

scale the network to large sizes and larger number of nodes. When scaling the number of nodes and the size of the network, care should be taken the amount of communication required to setup the communication schedule does not grow exponentially. A distributed approach to scheduling allows calculating a schedule as local as possible thereby reducing the amount of communication required. Moreover we extend the scheduling approaches with transmissions ordering, which allows reduction of the end-to-end delay in large-scale multihop networks.

- **A combined Time-of-Flight (ToF) and Direction-of-Arrival (DoA) localization and time-synchronization approach.** In Chapter 5 we propose a combined ToF and DoA localization approach. This approach uses one-way ranging and combines ToF and DoA to reduce the number of reference nodes required to perform localization and possibly increase the accuracy of localization. We have evaluated the performance of this approach using simulation and in an experiment in a dive-tank.

- **A cooperative combined localization and time-synchronization approach.** In Chapter 6 we propose a new cooperative combined localization and time-synchronization algorithm called aLS-Coop-Loc and compare this approach to a non-cooperative localization and time-synchronization approach. This approach can be used for small-scale clusters of nodes to perform relative localization and time-synchronization without requiring reference nodes and using one-way ranging. While combined localization and time-synchronization approaches exist for non-cooperative networks, no such approach existed for cooperative networks before. We perform both simulation as well as real-world tests to evaluate the performance. Tests were performed in different environments and with different hardware platforms. With the SeaSTAR node tests were performed in a short-range setup in a recreational water near the campus and in a fjord in Norway. With Commercial Of-the-Shelf (COTS) hardware from Kongsberg short-range tests were performed in a fjord in Norway and in the same fjord tests were performed with longer range communication.

- **The BigMAC protocol for non-cooperative localization and time-synchronization in a large-scale underwater localization system.**

  In Chapter 7 we propose a MAC protocol for a large-scale non-cooperative underwater localization and time-synchronization system. We evaluate in simulation how communication scheduling can improve the efficiency of such a MAC protocol as compared to unscheduled communication. For communication scheduling we combined the scheduling of Chapter 4 with broadcast scheduling, and for localization and time-synchronization we use the combined ToF and DoA localization approach proposed in Chapter 5.

- **The LittleMAC protocol for cooperative localization and time-synchronization in small-scale underwater clusters.**

  Chapter 8 shows a cooperative approach to communication and localization. This MAC protocol is designed for small autonomous clusters of nodes and

uses the aLS-Coop-Loc approach from Chapter 6 to calculate relative positions and time-synchronization without requiring reference nodes. Such an approach reduces the cost of deploying an UASN because determining the position of reference nodes is time-consuming and requires the use of an external positioning system. We evaluate the feasibility of such an approach using simulation and show that such a system can be designed even for systems supporting only low physical layer data-rates.

Finally we conclude this research and give directions for future research in Chapter 9.

Bibliography

[1] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," in *Proceedings of the 1st ACM International Workshop on Underwater Networks*, ser. WUWNet '06. New York, NY, USA: ACM, 2006, pp. 17–24. [Online]. Available: http://doi.acm.org/10.1145/1161039.1161045

[2] T. Eklund and G. Paulsen, "Ormen lange offshore project subsea development strategy and execution," *Proceedings of the 17th International Offshore and Polar Engineering Conference*, 2007.

[3] T. J. Kvalstad, F. Nadim, A. M. Kaynia, K. H. Mokkelbost, and P. Bryn, "Soil conditions and slope stability in the ormen lange area," *Marine and Petroleum Geology*, vol. 22, no. 1-2, pp. 299 – 310, 2005, ormen Lange - an integrated study for the safe development of a deep-water gas field within the Storegga Slide Complex, NE Atlantic continental margin.

[4] D. Pompili, T. Melodia, and I. F. Akyildiz, "Deployment analysis in underwater acoustic wireless sensor networks," in *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*. New York, NY, USA: ACM, 2006, pp. 48–55.

[5] Y. Xiao, Ed., *Underwater Acoustic Sensor Networks*. Auerbach Publications, 2010.

[6] 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

[7] B. W. Parkinson, A. I. for Aeronautics, Astronautics, GPS, and NAVSTAR, *Global positioning systems : theory and applications. Vol. 2.* American Institute of Aeronautics and Astronautics, 1996.

CHAPTER 2

Related work

2.1 Underwater acoustic communication

Underwater acoustic sensor networks are characterized by their significant delays and low communication speed. This is a result of the characteristics of the acoustic underwater channel. In [1] the characteristics of the acoustic underwater channel and the difficulties of underwater communication are discussed. Acoustic communication is different from radio communication and radio based physical, MAC and networking protocols can not be directly applied to underwater acoustic communication. We will review the properties of the acoustic channel and discuss the differences with radio communication.

The propagation speed of the acoustic signal is averaged around 1500 $^{m}/_{s}$, however the actual value depends on the, amongst others, salinity (S), temperature (T) and depth (D). An estimation using a nine-term equation of the speed of sound (c) underwater is given in [2]:

$$\begin{aligned} c = 1448.96 + 4.591T - 5.304 \times 10^{-2}T^2 + 2.374 \times 10^{-4}T^3 \\ +1.340(S-35) + 1.630 \times 10^{-2}D + 1.675 \times 10^{-7}D^2 \\ -1.025 \times 10^{-2}T(S-35) - 7.139 \times 10^{-13}D^3 m/s \end{aligned} \tag{2.1}$$

Generally the sound speed is assumed to be a constant ($\approx$1490 $^{m}/_{s}$) or a sound speed profile of the environment is measured and used.

The path loss of the signal can be modeled as follows [1]:

$$A(l, f) = (l/l_r)^k a(f)^{l-l_r}, \tag{2.2}$$

where f is signal frequency and l the transmission distance taken in reference to l_r. The path loss component k models the spreading loss and is usually between 1 and 2. The absorption coefficient can be obtained using an empirical formula [3]:

$$10 \log a(f) = \frac{0.11 f^2}{(1+f^2)} + \frac{44 f^2}{(4100+f^2)} + 0.000275 f^2 + 0.0003$$

This formula shows the strong frequency dependent component of the attentuation of the acoustic signal. The ambient noise is dependent on the environment of deployment. For ocean environments empirical formulas exist which model the noise from four sources: turbulence, shipping, waves and thermal noise [4]. The following

formulae give the power spectral density of the four noise components in dB relative to 1 $^{\mu Pa}/_{\text{hz}}$ as a function of frequency (f) relative to 1 kHz:

$$\begin{aligned} 10\log N_t(f) &= 17 - 30logf \\ 10\log N_s(f) &= 40 + 20(s-0.5) + 26\log(f) - 60\log(f+0.03) \\ 10\log N_w(f) &= 50 + 7.5w^{1/2} + 20\log f - 40\log(f+0.4) \\ 10\log N_{th}(f) &= -15 + 20\log f \end{aligned} \tag{2.3}$$

The frequency dependent absorption and noise and the slow propagation speed has significant impact on the design of MAC protocols for underwater communication networks. While traditional wireless MAC protocols can assume negligible propagation delays and use a large frequency band for communication, underwater MAC protocols should account for and compensate large delays and are very bandwidth limited.

2.2 MAC protocols

The main task of MAC protocols is to coordinate access to the communication medium. Without management of the medium, collisions occur and overall performance of the network degrades, hence the main objective of MAC protocols is to avoid collisions. MAC protocols should provide communication in an energy-efficient and scalable way and should reduce the latency of communication as much as possible. Because we are focusing on combining communication, localization and time-synchronization, we also look at related work in the area of localization and time-synchronization.

MAC protocols can prevent collisions by dividing the communication medium across different nodes in different ways. Figure 2.1 shows a classification of MAC protocols used in UASNs[5].

The medium can be divided into different frequency (Frequency Division Multiple Access (FDMA)), code (Code Division Multiple Access (CDMA)) or time (TDMA).

- **FDMA**, such as used by Seaweb [6], are faced with limited available frequency bandwidth and frequency dependent attenuation of the acoustic signal. The limited available frequency bandwidth and inefficient use of the frequencies result in low throughput of FDMA protocols. The frequency dependent attenuation causes big differences in power consumption and reliability of the communication when nodes are assigned different frequencies.

- **CDMA** protocols, such as used by UWAN-MAC [7], UW-MAC [8], EDETA [9] and HR-MAC [10] are more common than FDMA based protocols. CDMA based protocols do however require specialistic modems supporting CDMA transmissions and suffer from the near-far problem [11]. CDMA works by assigning different codes to different users in the network. This reduces the users throughput in comparison to a single-user case, but users can transmit without considering any of the other transmissions active. The power received by the receiver should be roughly the same for all users, otherwise the signal can not be decoded. This is called the near-far problem. In radio networks a closed-loop is used to regulate the power of the transmitters, however in underwater

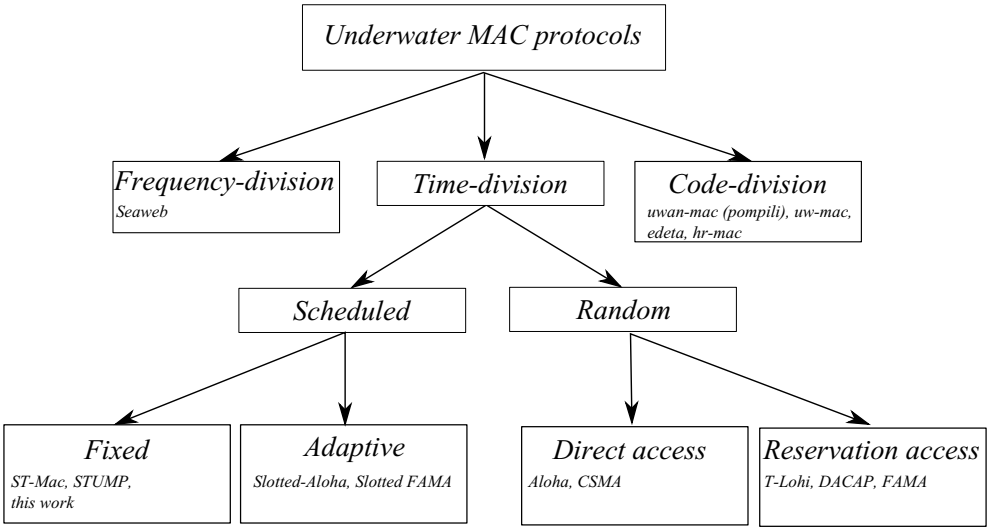


Figure 2.1: Classification of underwater MAC protocols.

networks with low propagation speed using a closed-loop is not very practical. While CDMA has been applied in many underwater MAC protocols, CDMA does require more complex receivers and few underwater modems support the usage of CDMA transmission and reception.

- **TDMA** approaches are the most common approach to medium access division in UASNs. The well-known ALOHA [12] protocol is used in underwater communication [13] and provides a very simple approach to MAC. A more underwater focused protocol such as Tone-Lohi [14] uses little coordination and operates in a decentralized manner. Random access approaches are easy to implement, robust because they use little or light coordination and adapt well to dynamic networks (such is the case with AUV). Random access approaches, however, are not very efficient in terms of energy-consumption, packet collisions may still occur, are not efficient in terms of bandwidth usage and usually provide very low throughput. Considering that bandwidth is already very limited and ineffective use of the bandwidth is quite wasteful.

  Fixed schedule-based approaches have significant benefits over other approaches, these benefits include improved success rate due to the avoidance of packet collision, reduced energy-consumption and improved throughput. Although TDMA is possible in underwater communication, this scheduling approach is generally considered inefficient because of the large propagation-delays of the acoustic signals and resulting large guard-times required. Scheduling approaches such as ST-MAC [15] and STUMP [16] schedule in such a way that the inefficiency of the large propagation delays is avoided. These scheduled based approaches use estimation of the propagation delay to schedule the reception of the packet.

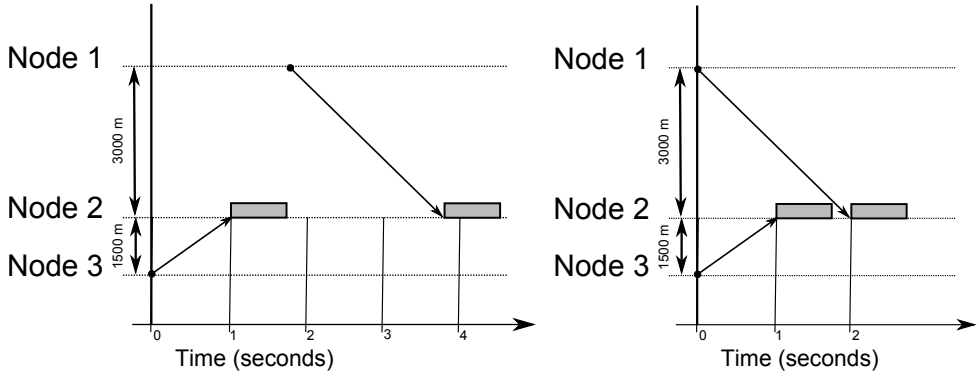
As has been noted in Section 1.1 we are focussing on applications using large number of nodes are deployed staticly in a relatively small area. One of our goals is to decrease the deployment and node cost to allow deployment of large number of nodes. Because CDMA requires more complex receivers and is not readily available on many existing underwater modems, we consider time-division approaches as the most viable approaches for these types of networks. Random access approaches are easy to implement and are robust, but are not very efficient in terms of energy-consumption and bandwidth usage. Fixed schedule approaches are able to provide energy-efficient and high-throughput communication, but exisiting approaches are cumbersome to use. In this work we propose a simplified scheduling approach for underwater communication. In Section 2.3 we look into more detail to existing underwater scheduling approaches.

Metrics

Different MAC protocol provide different trade-offs, to compare the MAC protocols, metrics need to be identified. We use the following metrics for evaluating MAC protocols:

- **Throughput.** The number of bits the MAC protocol is able to send per second. Ideally we would like to offer as much bandwidth as possible to the application running on the UASN. MAC protocols require a certain overhead for their operation, thereby reducing the throughput available for the application. This overhead should be kept to a minimum to provide transport of as much data as possible.
- **Scalability.** We are focusing on networks which have a large number of nodes on a limited coverage area and scalability of the proposed MAC protocol is important.
- **Energy-efficiency.** We would like to provide long-term deployment of networks, a MAC protocol should induce as little as possible overhead. In UASNs transmission of packets is one of the biggest energy consuming operations. MAC protocols should therefore introduce little overhead in terms of extra control packets to be transmitted (RTS/CTS packets) and large headers. Moreover collisions should be considered wasted transmissions and ideally MAC protocols avoid collisions completely.
- **End-to-end delay.** The end-to-end delay is the time it takes for a packet to travel from the generating sensor to the sink. The end-to-end delay should be as small as possible to allow real-time monitoring.

From looking at related work, it can be concluded that existing MAC protocols focus on communication only. In our view, looking at communication only for MAC protocols is too limited. A UASN requires not only communication, but also localization and time-synchronization. This is required, for example, for time-stamping and location-stamping sensor measurements. Moreover, time-based approaches to MAC requiring time-synchronization and scheduled based approaches require estimation



(a) Exclusive access (b) Scheduled

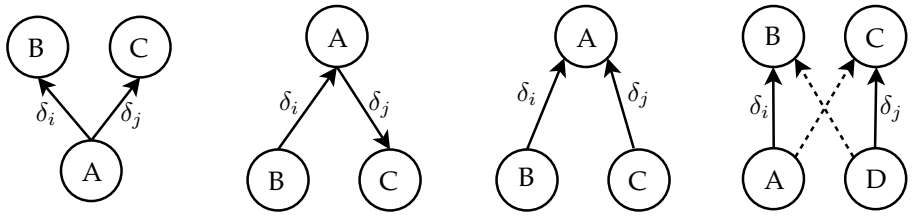
Figure 2.2: Exploiting spatial-temporal uncertainty in underwater communication with scheduling. Exclusive access of the medium is not required, rather reception of a packet needs to be timed exclusively. In the right picture is shown that two packets can be transmitted at the same time by node 1 and node 3, and, due to the difference in propagation delay to node 2, can both be received free of collisions.

of the position of nodes and propagation delays between nodes. Therefore it is important to consider how communication, localization and time-synchronization impact each other.

In [17] an evaluation of the impact of localization approaches on MAC protocols is presented, which shows that the choice of MAC has significant impact on localization performance in terms of time required for localization. Authors, however, consider only contention-based MAC protocols while many other underwater MAC protocols exist. At the same time there is an increasing interest in scheduling approaches for underwater communication. Examples of scheduling approaches for underwater communication include ST-MAC [15], STUMP [16].

2.3 Communication scheduling

Because of the slow propagation speed and the resulting large propagation times of the signal an uncertainty of the global state of the channel exists, this is called the space-time uncertainty [18]. Because of this 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 2.2 shows how two packets can be transmitted at the same time but are received without collision at the receiver. By exploiting the fact that we can have an estimation of the propagation delay, several transmissions can be scheduled at the same time as long as the reception of the packet is scheduled without interference. To do so, the scheduling algorithm needs to know all transmissions and all nodes within



(a) TX-TX conflict (b) TX-RX conflict (c) RX-RX conflict (d) RX interference

Figure 2.3: Illustration of all possible conflicts. Transmission tasks are denoted as δ_i and δ_j, shown are the difficult conflicts that may arise when scheduling the transmission of the two packets.

the network beforehand and should be able to make an estimation of the propagation delay of the acoustic signal between two nodes.

Because the propagation delay needs to be estimated and all transmissions should be known before scheduling the transmissions, scheduled communication is most suited for static networks. Setup of a schedule requires unschedled communication to collect the required information to perform scheduling and the benefits of using a schedule should outweigh the overhead of setting up such a schedule. This can usually be done only when the schedule stays valid for a long period of time.

The goal of scheduling is to coordinate the transmissions to avoid conflicts. A valid schedule should follow certain constraints to avoid packet collisions at the receiver. In both [15] and [19], 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 2.3). In Chapter 3 where we introduce our scheduling approach, we go into more detail of these scheduling conflicts.

In [20] a joint sensor deployment, link scheduling and routing approach is introduced. The approach uses an integer linear programming model to calculate optimal sensor deployment, link schedules and routes. Although such an approach is interesting, the computational overhead is large. Although no computational complexity is given, the article indicates a calculation time of three hours for a 30 node network. What the effects are for scaling this up to larger networks is unclear, authors note, however, a computational more efficient approach is required for larger networks. Moreover, authors use a slotted approach, however no indication of the effects of the slot size selection on the resulting schedule is given. In Chapter 3 we show an unslotted scheduling approach outperforms slotted scheduling approaches.

Existing scheduling approaches such as ST-MAC and STUMP are able to schedule communication but do so at the cost of complex scheduling algorithms. ST-MAC uses graph-coloring for scheduling, which may be cumbersome and uses time-slots, which is sub-optimal.

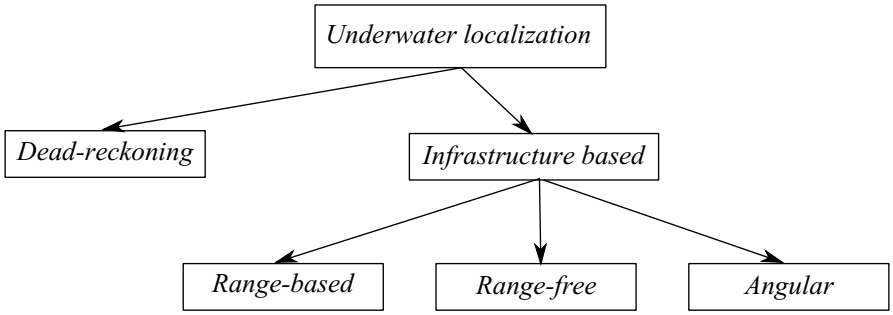


Figure 2.4: Classification of underwater localization approaches.

2.4 Localization and time-synchronization

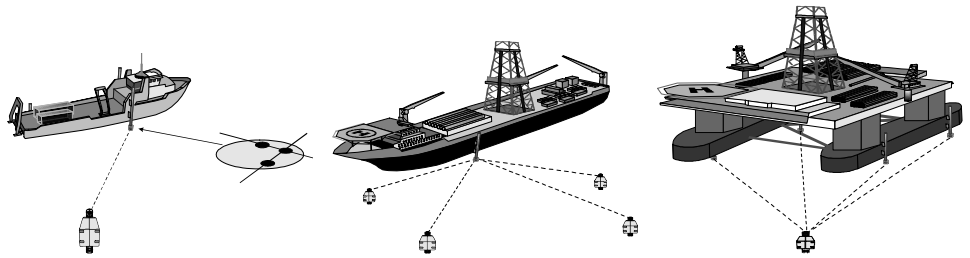
When performing measurements it is not only important what is measured, but also when and where. This gives localization and time-synchronization an important role in monitoring applications of Underwater Acoustic Networks (UANs). Figure 2.4 gives an overview of localization techniques used in UASNs. A more extensive overview of localization techniques is given in [21].

Dead-reckoning, calculating a position relative to a previously calculated position, using inertial naviation is commonly used in UASNs for tracking AUVs and remains an active field of research. Dead-reckoning approaches however provide accurate position for a limited time because of the cumulative error. For our targeted applications, static networks deployed for a long period of time, dead-reckoning does not provide a good long-term accuracy.

While WSN localization algorithms use range-free (connectivity information only) and range-based (using some estimation of the inter-distance) approaches and use ToF and signal strength based approaches for determining distances, localization approaches in UASNs generally use ToF and DoA based approaches. This is because the acoustic signal used in underwater communication propagates much slower than the radio signals used in traditional WSN, and ToF and DoA is relatively easy to estimate and provides an accurate estimate of node distance and incoming signal angle.

Ranging, or determining the distance between two nodes, can be performed using one-way to two-way communication. In a two-way ranging approach both nodes transmit packets. A packet is sent and the other node responds with a reply. The distance between the two nodes is calculated based on the round-trip time of the packet. The advantage of such an approach is that no time-synchronization is required to perform the ranging, the round-trip time can be calculated on the local clock of the initiator of the ranging. One-way ranging uses a single transmitter and a single receiver and requires time-synchronization to calculate distance.

Examples of commercial acoustic dynamic positioning systems which are in widespread use today are Long Baseline (LBL), Short Baseline (SBL) and Ultra Short Baseline (USBL) [22] systems. Figure 2.5 gives an example of the operation of these systems. These systems are used to track AUVs using reference infrastructure



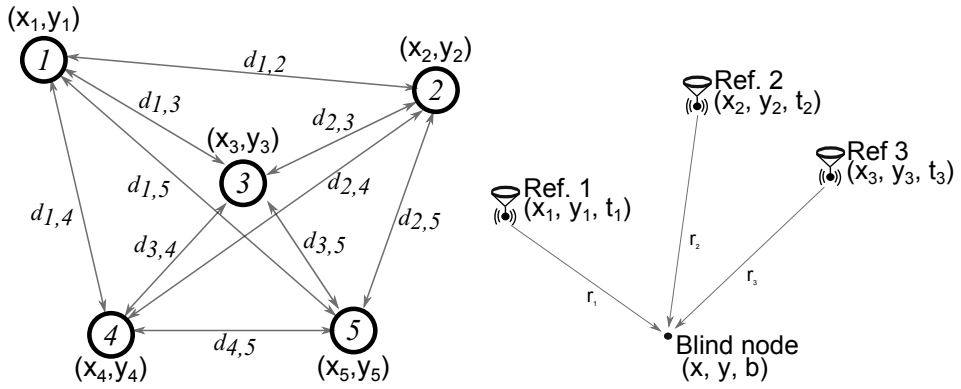
(a) Ultrashort baseline (b) Long baseline (c) Short baseline

Figure 2.5: Example of commercial approaches to underwater dynamic positioning, approaches are classified by the length of the baseline. In Figure 2.5(a) a beamforming array attached to the ship is used to determine the incoming angle of the signal, two-way ranging is used te determine the distance to the submerged pinger. In Figure 2.5(b) sea-floor mounted reference transponders are used, in Figure 2.5(c) the reference responders are attached to the ship. Images were taken from [22].

mounted on a ship or the sea bottom. Systems such as LBL and SBL use two-way ranging between reference transducers and the blind node to estimate the position of the blind-node. A system such as USBL uses two-way ranging between a reference transducer and a blind node and uses multi-element transducers at the reference node to determine the angle of the incoming signal. Using DoA and ToF information, the position of the blind node can be determined with only a single reference node.

Infrastructure based localization approach can be split up into cooperative and non-cooperative based approaches. Figure 2.6 shows an example of cooperative and non-cooperative localizations. The clear separation between the unlocalized and unsynchronized blind-nodes and the synchronized reference nodes with known positions we consider as a distinguishing factor between cooperative and non-cooperative localization. In cooperative localization there is no clear separation between reference nodes and blind-nodes and all nodes cooperate to determine their position and time-synchronization. Moreover cooperative localization uses considerably more measurements as all pair-wise distance measurements between the nodes in the network are used. This, potentially, increases the accuracy of localization and allows more flexible selection of the reference nodes. While in non-cooperative localization there is a clear separation between reference and nodes and blind-nodes, in cooperative localization this separation may be partial as only a number of reference nodes have reference information for only a single dimension.

An example of a time-synchronization is TSHL [23]. To perform time-synchronization an estimation of the propagation delay between the time-reference and the unsynchronized node is required. This requires two-way ranging or requires knowledge of the position or distance between nodes. When both positioning and time-synchronization is required, a combined approach is generally better. Approaches which attempt to minimize the communication overhead of time-synchronization,



(a) Cooperative

(b) Non-cooperative

Figure 2.6: Example of cooperative and non-cooperative localization. Cooperative localization determines the distances between all pairs of nodes in the network and there is no clear separation between reference nodes and blind-nodes. Non-cooperative localization uses ranging between reference nodes and blind-nodes and there is a clear separation between reference nodes and blind-nodes.

such as [24], sometimes assume the position of sensors are known. However in our view this marginalizes the overhead of localization and overhead of both aspects should be considered to evaluate the overhead of the whole system.

Existing work on time-synchronization and localization [25] consider these aspects separately. However combined localization and time-synchronization, similar to what is already done by GPS [26] or Silent Positioning [27], solve the problem of performing these two tasks separately and sequentially. This allows the position and time to be simultaneously estimated using one-way ranging only. This offers benefits in terms of accuracy of localization and time-synchronization, but can also reduce communication overhead and reduce energy-consumption by using one-way ranging and broadcasts.

One-way ranging offers significant benefits in terms of lower communication overhead compared to two-way ranging. With one-way ranging the number of communication required before localization is done is reduced significantly. Because bandwidth is very limited in underwater acoustics and data rates are very low, localization and time-synchronization using one-way ranging is very important. Another advantage offered by one-way ranging is reduction of energy consumption due to lower communication overhead offered by one-way ranging. To allow localization and time-synchronization using only one-way ranging, a combined localization and time-synchronization approach is required. In Section 6.3.2 we show the benefits of combining localization and time-synchronization in terms of communication overhead and power consumption.

Non-cooperative approaches combining localization and time-synchronization already exist, an example of which is the GPS system [26], however a cooperative

approach which combines localization and time-synchronization has never been proposed. Multi-Dimensional Scaling (MDS) localization is a well-known approach to cooperative localization. MDS provides localization but requires prior time-synchronization or two-way ranging.

Metrics

For evaluation localization and time-synchronization protocols we look at the following metrics:

- **Accuracy**. How accurate does the localization and time-synchronization algorithm calculate the real position and real clock-bias of the node.
- **Scalability**. A localization and time-synchronization approach induces a certain communication approach to do the measurements used as an input for calculating position and time. The communication required by the localization and time-synchronization algorithm should be as little as possible.
- **Energy-efficiency**. Next to the scalability effects, the localization and time-synchronization approach induced communication pattern has significant influences on the energy-consumption of the nodes. Because available energy is limited in autonomous battery-powered underwater nodes, energy-efficiency of the chosen localization and time-synchronization is an important concern. Transmission of data in underwater acoustic communication is a very expensive operation, in general and especially compared to the power consumption of receiving a packet, therefore the amount of transmissions should be kept to a minimum to preserve energy.

MAC protocols designed for a localization and time-synchronization system also have an influence on these criteria and are also evaluated using these metrics. Moreover the MAC protocol design for localization is evaluated using the following metric:

- **Time required for localization.** The time it takes to localize all the nodes in the network.

2.5 Conclusion

Underwater acoustic communication is different from radio communication. When designing protocols for underwater communication one should consider the slow propagation speed of the acoustic signal (compared to radio) and frequency dependent attenuation. Although the propagation speed of the acoustic signal is dependent on temperature and pressure, in the rest of the work we assume the propagation speed to be constant.

The main task of MAC protocols are to coordinate access to the communication medium. Many approaches exist for dividing the acoustic medium among different users in the network, the main approaches are: FDMA, CDMA and time-division.

Due to the limited available bandwidth, FDMA is not very practical. Because of the near-far problem, CDMA is not a practical approach. Many approaches exist in the time-division category of MAC protocols.

Contention based protocols, such as Aloha, FAMA and T-Lohi, are used often in underwater networks. They use distributed coordination and are well suited for dynamic networks, such as where AUVs are used. They, however, introduce a significant overhead and offer only low bandwidth. Fixed schedule based approaches, such as ST-MAC and STUMP, offer significant benefits in terms of throughput and are able to avoid collisions. Usage of time-slots in scheduling algorithms is sub-optimal (shown in Chapter 3). In Chapter 3 and Chapter 4 we show communication scheduling can be done in a much simpler way than is done by existing scheduling approaches. Moreover we show our greedy scheduling algorithm outperforms existing transmission ordering heuristics.

Regarding localization, ToF and DoA is the most widely used approach to underwater localization. Many systems use two-way ranging, however, this is a problem for the scalability and energy-consumption of such approaches. Time-synchronization approaches also use two-way ranging to estimate propagation delays between nodes or make an assumption that the position of nodes are known to reduce communication. By combining localization and time-synchronization (as done in GPS [26]) it is possible to determine the position of nodes and perform time-synchronization using one-way ranging. In Chapter 5 we show a combined ToF and DoA approach using one-way ranging, and in Chapter 6 we show a one-way ranging cooperative localization approach.

Bibliography

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

[2] K. V. Mackenzie, "Nine-term equation for sound speed in the oceans," *Acoustical society of America*, pp. 807–801, 1981.

[3] L. M. Brekhovskikh, Yu, L. M. Brekhovskikh, and Y. Lysanov, *Fundamentals of Ocean Acoustics*, 3rd ed. Springer, March 2003. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&path=ASIN/0387954678

[4] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," in *Proceedings of the 1st ACM international workshop on Underwater networks*, ser. WUWNet '06. New York, NY, USA: ACM, 2006, pp. 41–47. [Online]. Available: http://doi.acm.org/10.1145/1161039.1161049

[5] S. Climent, A. Sanchez, J. V. Capella, N. Meratnia, and J. J. Serrano, "Underwater acoustic wireless sensor networks: Advances and future trends in physical, mac

and routing layers," *Sensors*, vol. 14, no. 1, pp. 795–833, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/1/795

[6] J. Rice, B. Creber, C. Fletcher, P. Baxley, K. Rogers, K. McDonald, D. Rees, M. Wolf, S. Merriam, R. Mehio, J. Proakis, K. Scussel, D. Porta, J. Baker, J. Hardiman, and D. Green, "Evolution of seaweb underwater acoustic networking," in *OCEANS 2000 MTS/IEEE Conference and Exhibition*, vol. 3, 2000, pp. 2007–2017 vol.3.

[7] D. Pompili, T. Melodia, and I. Akyildiz, "A cdma-based medium access control for underwater acoustic sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 4, pp. 1899–1909, 2009.

[8] M. K. Watfa, S. Selman, and H. Denkilkian, "Uw-mac: An underwater sensor network mac protocol," *Int. J. Commun. Syst.*, vol. 23, no. 4, pp. 485–506, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1002/dac.v23:4

[9] S. Climent, J. V. Capella, N. Meratnia, and J. J. Serrano, "Underwater sensor networks: A new energy efficient and robust architecture," *Sensors*, vol. 12, no. 1, pp. 704–731, 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/1/704

[10] G. Fan, H. Chen, L. Xie, and K. Wang, "A hybrid reservation-based {MAC} protocol for underwater acoustic sensor networks," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1178 – 1192, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870513000048

[11] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," in *Proceedings of the 1st ACM International Workshop on Underwater Networks*, ser. WUWNet '06. New York, NY, USA: ACM, 2006, pp. 17–24. [Online]. Available: http://doi.acm.org/10.1145/1161039.1161045

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

[13] L. F. Vieira, J. Kong, U. Lee, and M. Gerla, "Analysis of aloha protocols for underwater acoustic sensor networks," in *Work in Progess poster at the First ACM International Workshop on UnderWater Networks (WUWNet)*. Los Angeles, California, USA: ACM, September 2006.

[14] A. A. Syed, W. Ye, and J. Heidemann, "T-Lohi: A new class of MAC protocols for underwater acoustic sensor networks," USC/Information Sciences Institute, Tech. Rep. ISI-TR-638b, April 2007, technical report originally released April 2007, updated July 2007. [Online]. Available: http://www.isi.edu/~johnh/PAPERS/Syed07a.html

[15] 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

[16] P. M. Kurtis Kredo II, "Distributed scheduling and routing in underwater wireless networks," *Globecom 2010*, 2010.

[17] J.-P. Kim, H. P. Tan, and H.-S. Cho, "Impact of mac on localization in large-scale seabed sensor networks." in *AINA*. IEEE Computer Society, 2011, pp. 391–396. [Online]. Available: http://dblp.uni-trier.de/db/conf/aina/aina2011.html#KimTC11

[18] A. A. Syed, W. Ye, and J. Heidemann, "T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks," in *IEEE INFOCOM*, 2008.

[19] J. Y. Yang Guan, Chien-Chung Shen, "MAC scheduling for high throughput underwater acoustic networks." submitted to IEEE WCNC 2011, Cancun, Quintana-Roo, Mexico, 2010.

[20] L. Badia, M. Mastrogiovanni, C. Petrioli, S. Stefanakos, and M. Zorzi, "An optimization framework for joint sensor deployment, link scheduling and routing in underwater sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 44–56, Oct. 2007. [Online]. Available: http://doi.acm.org/10.1145/1347364.1347374

[21] M. Erol-Kantarci, H. Mouftah, and S. Oktug, "A survey of architectures and localization techniques for underwater acoustic sensor networks," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 3, pp. 487–502, 2011.

[22] K. Vickery, "Acoustic positioning systems. a practical overview of current systems," in *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*, aug 1998, pp. 5 –17.

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

[24] D. Zennaro, B. Tomasi, L. Vangelista, and M. Zorzi, "Light-sync: A low overhead synchronization algorithm for underwater acoustic networks," in *OCEANS, 2012 - Yeosu*, May 2012, pp. 1–7.

[25] 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

[26] B. W. Parkinson, A. I. for Aeronautics, Astronautics, GPS, and NAVSTAR, *Global positioning systems : theory and applications. Vol. 2.* American Institute of Aeronautics and Astronautics, 1996.

[27] X. Cheng, H. Shu, Q. Liang, and D.-C. Du, "Silent positioning in underwater acoustic sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 3, pp. 1756–1766, May 2008.

Part I

Scheduled communication

CHAPTER 3

Scheduling for small-scale networks

The acoustic propagation speed under water poses significant challenges to the design of UASNs and their MAC 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 chapter, we derive a set of simplified scheduling constraints and propose two unslotted scheduling algorithms with relatively low complexity for both known and unknown orders of transmissions. Our simulations show that for large packet sizes our scheduling approach without slots is on average 4% better in terms of throughput than existing scheduling approaches with slots, while for small packet sizes scheduling without slots results in 30% shorter schedule lengths (total time to execute the schedule). We also compare our "smallest delay first" heuristic algorithm with the "highest transmission load first" heuristic of ST-MAC, the "lowest transmission load first" heuristic and random transmission ordering of STUMP and show that our heuristic algorithm performs on average 10% better for small packets and 2% for large packets compared to these heuristics.

3.1 Introduction

In this chapter we look at how scheduling of underwater communication can be done in a simple and efficient way. Both [1] and [2] 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.

The approach in [1] uses graph-coloring for scheduling, which may be cumbersome and requires time-slots, which is sub-optimal. In addition, the authors do not model the processing time of the packet, which results in reception of the packet possibly spanning through several time-slots. The approach described in [2], 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 chapter, 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 4% better in terms of throughput than scheduling with

slots for large packet sizes, while for small packet sizes scheduling without slots results in 30% shorter schedule lengths. Moreover we compare different heuristics for ordering the transmissions and show a our greedy scheduling approach, scheduling transmissions with minimum delay first, outperforms heuristics used by other scheduling approaches.

In Section 3.2 we explain how to derive the set of simplified scheduling 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 3.4, will calculate the shortest transmission schedule using a given order of transmissions. While the second algorithm, described in Section 3.5, 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 3.6.

3.2 Underwater communication scheduling constraints

Initially we look at scheduling transmissions in a fully-connected dense UASN, in Chapter 4 we will extend the scheduling approach for scheduling larger-scale networks. 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 [2] so that they describe how one transmission task can be scheduled after another transmission task. In other words, let us consider two transmission tasks denoted as δ_i and δ_j and let us assume that δ_j should be scheduled after δ_i. We aim to determine what constraints should be applied between the current and all previous transmissions. In case we want to schedule transmission δ_{n+1}, we should verify the constraints with all transmissions from δ_0 up to and including transmission δ_n.

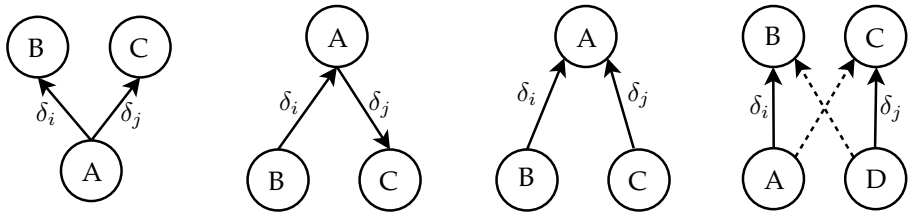
For each transmission, we need to calculate the transmission starting time δ_{start}. Every transmission has a certain duration $\delta_{duration}$, source δ_{src} and destination δ_{dst}. We assume the unspecified function T will give the transmission delay between two nodes.

We specify the constraints in such a way that transmission δ_j is scheduled after or at the same time as transmission δ_i. In other words:

$$\left\{ \delta_j.start \geq \delta_i.start \quad \text{if } j \geq i \right. \tag{3.1}$$

Figure 3.1 illustrates the possible conflicts that arise. 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 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 (δ_i) is finished when the second transmission (δ_j) starts. This can be formulated as:



(a) TX-TX conflict (b) TX-RX conflict (c) RX-RX conflict (d) RX interference

Figure 3.1: Illustration of all possible conflicts. Transmission tasks are denoted as δ_i and δ_j, shown are the difficult conflicts that may arise when scheduling the transmission of the two packets.

$$\delta_i.start + \delta_i.duration \leq \delta_j.start \tag{3.2}$$

- TX-RX conflict: A node can not receive a packet when it is transmitting a packet. So the second transmission (δ_j) should start after the first transmission (δ_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:

$$\delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_i, dst) \leq \delta_j.start \tag{3.3}$$

$$\delta_i.start + T(\delta_i.src, \delta_i, dst) \geq \delta_j.start + \delta_j.duration \tag{3.4}$$

- RX-RX conflict: In this type of conflict a single node is the destination of two transmissions. To correctly schedule the transmission, the second transmission (δ_j) should start after the first transmission (δ_i) has been received. This can be described using the following formula:

$$\delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_i, dst) \leq \delta_j.start + T(\delta_j.src, \delta_j.dst) \tag{3.5}$$

- RX-Interference: To avoid that two transmissions interfere with each other, a transmission should not be interfering with the reception of another node. This means that if we want to schedule δ_j after δ_i, the message of transmission δ_j should arrive at destination $\delta_i.dst$ after the message from δ_i has been received and processed by $\delta_i.dst$, but the message from δ_j should also arrive at node $\delta_j.dst$ after the message from δ_i has been processed at node $\delta_j.dst$. In other words:

$$\delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_i.dst) \leq \delta_j.start + T(\delta_j.src, \delta_i.dst) \quad (3.6)$$

$$\delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_j.dst) \leq \delta_j.start + T(\delta_j.src, \delta_j.dst) \quad (3.7)$$

The above constraints overlap with the equations found in [1] and [2]. The difference, however, is that the equations in these articles make no assumption about the order of transmissions. This implies that it is possible to have delay of 1 between δ_i and δ_j, and a delay of -1 between transmission δ_j and δ_i. But a negative delay means that δ_i is actually scheduled before δ_j, which makes scheduling more difficult.

Because we want to schedule the transmissions in a given order, the delays must all be ≥ 0. So when transmission δ_j is scheduled after transmission δ_i, we will not have a negative delay, because $\delta_j.start \geq \delta_i.start$. However when scheduling a transmission, all relations to all transmissions before the "to be scheduled" transmission should be validated. To do so, in the next section we will reduce and re-define the rules to find a set of constraints that allows much easier scheduling.

3.3 A simplified set of scheduling constraints

In this section we will discuss how the scheduling rules can be simplified to have a set of constraints to be used for easier scheduling algorithms.

If we look back at the constrains given in Section 3.2, we observe that:

- Both equation (3.6) and equation (3.7) are applied at the same time, but we can write them as follows in a single equation:

$$\begin{aligned} \delta_i.start + \delta_i.duration + max(T(\delta_i.src, \delta_i.dst) - T(\delta_j.src, \delta_i.dst), \\ T(\delta_i.src, \delta_j.dst) - T(\delta_j.src, \delta_j.dst)) \leq \delta_j.start \end{aligned} \quad (3.8)$$

- For all the rules except rule (3.4) we can choose any transmission time greater than $\delta_j.start$ and still get a valid schedule, i.e:

$$\delta_j.start' \geq \delta_j.start \quad (3.9)$$

  Equation (3.9) will give a valid schedule for any start time greater than the minimum start time.

- In the case of RX-RX conflict, because $\delta_i.dst = \delta_j.dst$, equation (3.5) and equation (3.8) will become the same. Since we have chosen to assume the transmission order as given, it is not needed to distinguish these different cases in our situation.

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

Figure 3.2: Set of simplified scheduling constraints

- If we model transmission as a node receiving its own message, we can see that equation (3.3) and rule (3.8) also map to the same problem. In mathematical notation, a node receiving its own message would imply that $\delta.src = \delta.dst$, and $T(\delta.src, \delta.src) = 0$. Therefore equation (3.10) can be written as equation (3.11). On the other hand, one can note that equation (3.11) is equal to equation (3.3).

$$\delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_i.dst) \leq \delta_j.start + T(\delta_j.src, \delta_j.src) \quad (3.10)$$

$$\delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_i.dst) \leq \delta_j.start \quad (3.11)$$

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 (3.4). As a result, when an algorithm uses this set of rules, it may give a slightly suboptimal solution in some cases.

Because we have assumed the order of transmissions is given, the problem of scheduling is greatly simplified and we basically only have to follow two scheduling constraints, namely equation (3.2) and equation (3.8):

- if transmission δ_i is from the same source as transmission δ_j ($\delta_i.src = \delta_j.src$), we can calculate the minimum transmission time using equation (3.2).
- if the transmission is sent from different nodes $\delta_i.src \neq \delta_j.src$, we should apply equation (3.8).

Because we have dropped equation (3.4), 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 (3.9)).

The resulting set of simplified scheduling constraints are summarized in Figure 3.2.

Because we have assumed the order of the transmissions as given, the performance of the scheduling algorithm (in terms of resulting schedule length) is very much dependent on the order of the transmissions prior to scheduling. In Section 3.4 we look at how scheduling can be performed given a order of transmissions. This approach assumes that the transmissions need to be scheduled in a certain order due

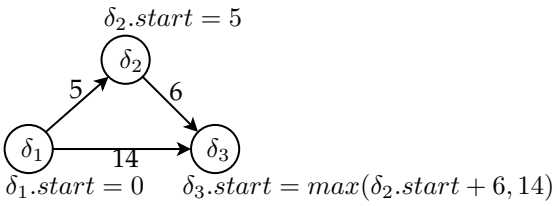


Figure 3.3: Calculating the new delay for transmission δ_3 after scheduling δ_2 after δ_1. Using the set of constraints a minimum delay between every transmission can be calculated. The scheduling algorithm enforces the minimum delay between the transmission. In the above example δ_2 is scheduled after δ_1 and requires a delay of 5 (this was determined using the scheduling constraints), the transmission start time is therefore set to $\delta_2.start = 5$ using the scheduling algorithm. Transmission δ_3 requires a delay of 14 with δ_1 and 6 with transmission δ_2 and is therefore scheduled at $\delta_3.start = max(\delta_2.start + 6, 14)$.

to external constraints. In Section 3.5 however we look at how scheduling can be performed without a prior given order of transmissions. In this algorithm we apply a greedy heuristic approach to ordering the transmissions to minimize the schedule length and attempt to find an optimal order of transmissions yielding the highest throughput.

3.4 Algorithm for scheduling a fixed order of transmissions

We will now describe a scheduling algorithm with $O(N^2)$ complexity, where N is the number of transmissions, for scheduling a given transmission order. This scheduling algorithm uses the simplified scheduling constraints described in equation (3.2) to find the minimum transmission times which give no packet collisions at the receivers.

What the algorithm presented in Figure 3.4 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.

Every time we schedule a transmission, we calculate the maximum delay required for the transmission and schedule the transmission. After that we update the required delays for all other transmissions. How these delays are updated is also shown also in Figure 3.3.

In "Calculate the end-time of the schedule" loop, the end-time of the schedule is calculated. This is the time when all transmissions are received by their destinations.

Require: $transmissions$ the set of all transmission records, S the set of indices containing the order in which to schedule the transmissions
Ensure: $\delta_*.start$ results in a collision-free schedule, $time_{end}$ the total time required to execute the schedule
$V \leftarrow transmissions$ {Set of all transmissions}
$schedule \leftarrow [N] \leftarrow 0$ {Resulting schedule initialized to zero}
$\delta_0.start \leftarrow 0$ {Schedule the first transmission}
for $j \leftarrow 1$ **to** N **do**
  $c[0][j] \leftarrow constraint(V[0], V[j])$
end for
$time \leftarrow 0$
for $i \leftarrow 1$ **to** N **do**
  $delay \leftarrow max(c[i-1][S[i]], constraint(V[S[i-1]], V[S[i]]))$
  $\delta_{V[j]}.start \leftarrow time + delay$
  for $j \leftarrow 1$ **to** N **do**
    $c[i][j] \leftarrow max(c[i-1][j], constraint(V[i], V[j])) - delay$
  end for
  $time \leftarrow time + delay$
end for
{End time calculation loop}
$time_{end} \leftarrow 0$
for $i \leftarrow 1$ **to** N **do**
  $time_{end} \leftarrow max(time_{end}, \delta_i.start + T(\delta_i.src, \delta_i.dst) + \delta_i.duration)$
end for

Figure 3.4: Scheduling algorithm for a fixed order of transmissions, the algorithm takes the set of all transmissions ($transmissions$) and calculates a start time for transmission for every transmission ($\delta_*.start$) such that a collision-free transmission is ensured. To do so, the algorithm uses the set of constraints to calculate the minimum delay between every transmission. The function $constraint$ returns the minimum delay between two transmissions determined using the simplified set of constraints. The maximum delay for every transmission is stored in the two dimensional array c (the conflict array), the first index is the iteration number and the second the transmission. After every schedule step the delays are updated according the process described in Figure 3.3. The algorithm schedules the transmission in the order stored in the set S, first index $S[0]$ is scheduled, next $S[1]$, etc...

3.5 Heuristic algorithm for finding a 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 3.4. 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 minimum delay. We will evaluate every transmission as first transmission and take the minimum schedule from all evaluated schedules. This algorithm has $O(N^3)$ complexity in the number of links.

The outer loop of the algorithm presented in Figure 3.5 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 3.4. 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 3.4.

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.

Require: $transmissions$ the set of all transmission records
Ensure: $schedule_{min}$ contains the most time efficient schedule and $time_{min}$ is the shortest schedule time of all greedily scheduled transmission schedules.

```
V ← transmissions {Set of all transmissions}
time_min ← ∞
schedule_min ← []
for v ∈ V do
    T ← [N]
    T[0] ← v
    c ← [N * N] = 0
    δ_T[0].start ← 0
    for j ← 1 to N do
        c[0][j] ← constraint(V[0], V[j])
    end for
    {Set S contains the transmissions still to be scheduled}
    S ← V \ T[0]
    time ← 0
    for i ← 1 to N do
        delay_min ← ∞
        link_min ← 0
        for j ← 1 to Length(S) do
            delay ← max(c[i − 1][S[j]], constraint(V[i], V[S[j]]))
            if S[j] ≠ i and delay < delay_min then
                delay_min ← delay
                link_min ← S[j]
            end if
        end for
        delay ← delay_min
        T[i] ← S[link_min]
        δ_S[link_min].start ← time + delay
        for j ← 1 to N do
            c[i][j] ← max(c[i − 1][j], constraint(V[i], V[j])) − delay
        end for
        {Remove scheduled transmission from transmissions to be scheduled}
        S ← S \ link_min
    end for
    {Calculate the end-time of the schedule}
    time_end ← 0
    for i ← 1 to N do
        time_end ← max(time_end, δ_i.start + T(δ_i.src, δ_i.dst) + δ_i.duration)
    end for
    {Store the optimal schedule (min time)}
    if time_end < time_min then
        time_min ← time_end
        schedule_min ← T
    end if
end for
```

Figure 3.5: Heuristic algorithm for finding transmission order with minimum schedule time. The algorithm takes the set of all transmissions to be scheduled ($transmissions$) and attempts to find the most efficient schedule using a greedy scheduling approach. To do so, every transmission in the set of transmissions is tried as the first transmission. From there, the other transmissions are scheduled in a greedy approach, the transmission having minimum delay is scheduled next. The schedule having the smallest end time ($time_{end}$) is considered the most efficient schedule and is stored in $schedule_{min}$.

Parameter	Value
Area size:	500m by 500m
Data rate:	1000bps
Propagation speed:	1500 $^{m}/_{s}$
Node placement:	random / uniform
Communication range:	unlimited

Figure 3.6: Parameters used during simulation, nodes are uniformly deployed in an area of 500m by 500m. Assumed is all the nodes are able to communicate and interfere with each other. Simulations are run for 1000 different deployments for each setup.

3.6 Algorithm evaluation

To validate how effective the scheduling algorithms are, we implemented the second algorithm (heuristic algorithm to find the minimum schedule time). We randomly scattered a number of nodes in an area of $500m$ by $500m$ and assumed that all nodes are able to hear each other. The propagation speed is assumed to be 1500 $^{m}/_{s}$, while the data bit rate is $1000bps$. The summary of our simulation parameters is given in Figure 3.6. We generated a number of transmissions with random source and destination. For the packet processing time we assumed a packet of size 32. We ran the algorithm to find a minimum schedule time. To evaluate the algorithms we looked at the throughput, which can be calculated by the total numbers of transmissions scheduled (N), the packet size in bits ($packet_{size}$) and schedule length ($schedule_{time}$) as:

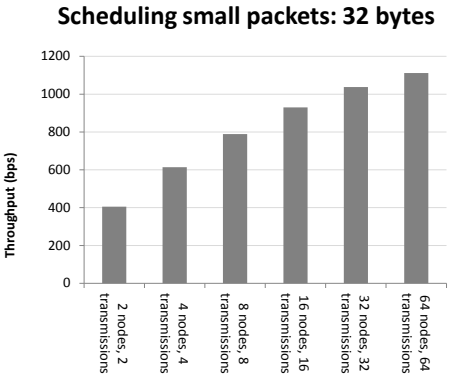
$$throughput = \frac{N * packet_{size}}{schedule_{time}} \tag{3.12}$$

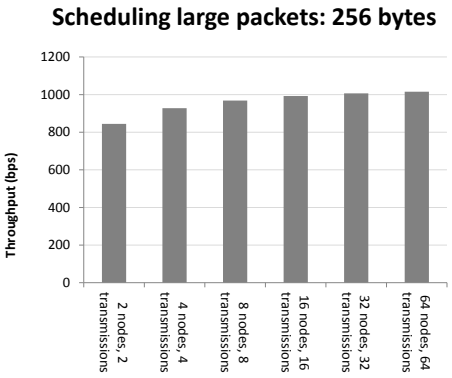
We performed this experiment for different number of nodes/transmissions ranging from 2 nodes with 2 transmissions up to 64 nodes with 64 transmissions.

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. From the results shown in Figure 3.7(a), we can see that when more transmissions are to be scheduled, the algorithm will calculate more efficient schedules.

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 3.7(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.

In Figure 3.8 we show the effect of an increasing modulation rate of the acoustic modem on the throughput of the resulting schedule. In simulation for the same deployment (1000 deployments of 64 nodes, 64 transmissions on an area of 500m by 500m), we increase the modulation rate from 100bps to 10000bps. What can be seen from the results is that for lower bitrates the schedule throughput linearly follows the modulation rate. For higher modulation rates however, the resulting schedule throughput is lower than the modulation rate of the node. This is due to the effect of the propagation delay. Because the propagation delays remain the same while





Setup	Throughput
2 nodes, 2 transmissions	405 bps
4 nodes, 4 transmissions	614 bps
8 nodes, 8 transmissions	789 bps
16 nodes, 16 transmissions	930 bps
32 nodes, 32 transmissions	1038 bps
64 nodes, 64 transmissions	1112 bps

(a) Small packets setup (32 bytes)

Setup	Throughput
2 nodes, 2 transmissions	844 bps
4 nodes, 4 transmissions	928 bps
8 nodes, 8 transmissions	969 bps
16 nodes, 16 transmissions	992 bps
32 nodes, 32 transmissions	1007 bps
64 nodes, 64 transmissions	1015 bps

(b) Large packets setup (256 bytes)

Figure 3.7: Resulting throughput of our scheduling algorithm in different setups

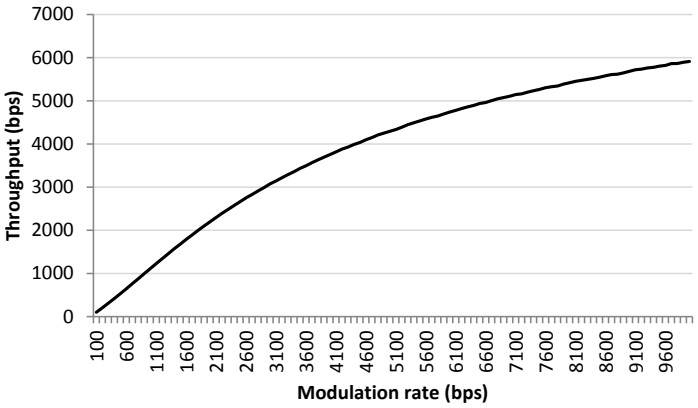


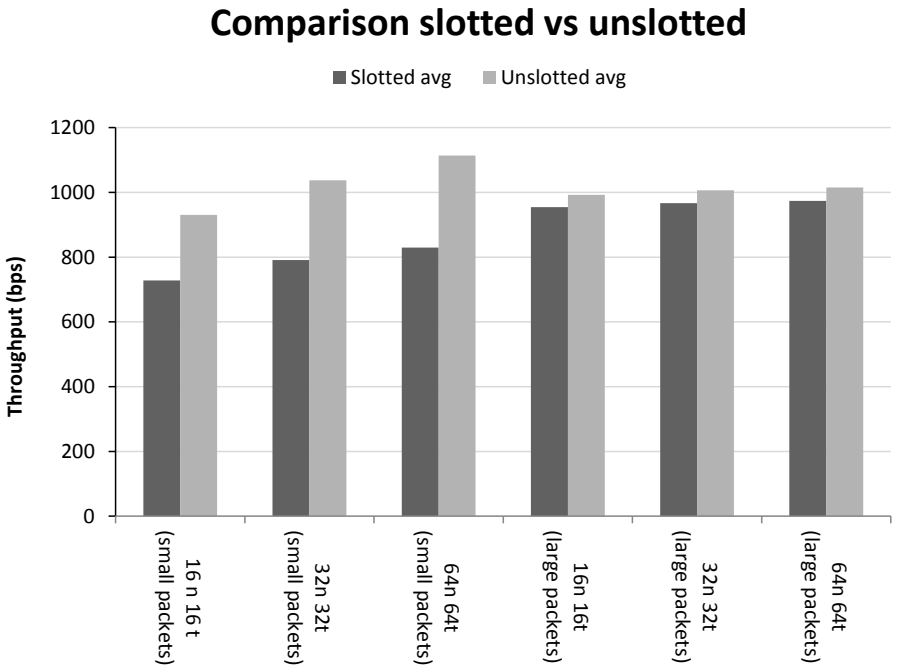
Figure 3.8: Schedule throughput as a result of increasing modulation rate of the acoustic modem. What can be clearly seen is that an increase of the modulation rate does not result in a linear increase of the schedule throughput. This is due to the effects of the propagation delays.

the modulation time decreases, the effect of the propagation delay on the schedule efficiency increases. In Section 4.5.2 we look into more detail of this relation.

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 3.9. 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 has a severe impact on the schedule times and the unslotted approach works about 30% better in these situations. In the case of having large packets, the effects of using slots is less severe and the unslotted approach is still about 4% better.

ST-MAC [1] first schedules transmissions with the highest load first, STUMP schedules transmissions nodes with the lowest load first or in a random order. Therefore, we compared our heuristic (schedule transmission with minimum delay first) against these heuristics. To do so, we changed the setup of the network and doubled the number of transmissions. This means that for a 16 node network, we scheduled 32 random transmissions. We did this to make a larger difference between busy and low-load nodes. Then we calculated the traffic load of a node based on the number of incoming transmissions on this node. Results of scheduling using these heuristics can be seen in Figure 3.10. Again, the difference is much more severe for smaller packets and the heuristic of scheduling transmission based on node load proved to be worse than our proposed heuristic. In the case of small packet transmissions our heuristic resulted in 10% better schedules on average and 2% for large packets. Scheduling transmissions in a random order results in a 30% worse schedule in the case of small packets and a 7% worse schedule in case of large packets. Interesting to see is that scheduling minimum load nodes first or maximum load nodes first result in about the same schedule efficiency.



(a) Slotted vs unslotted

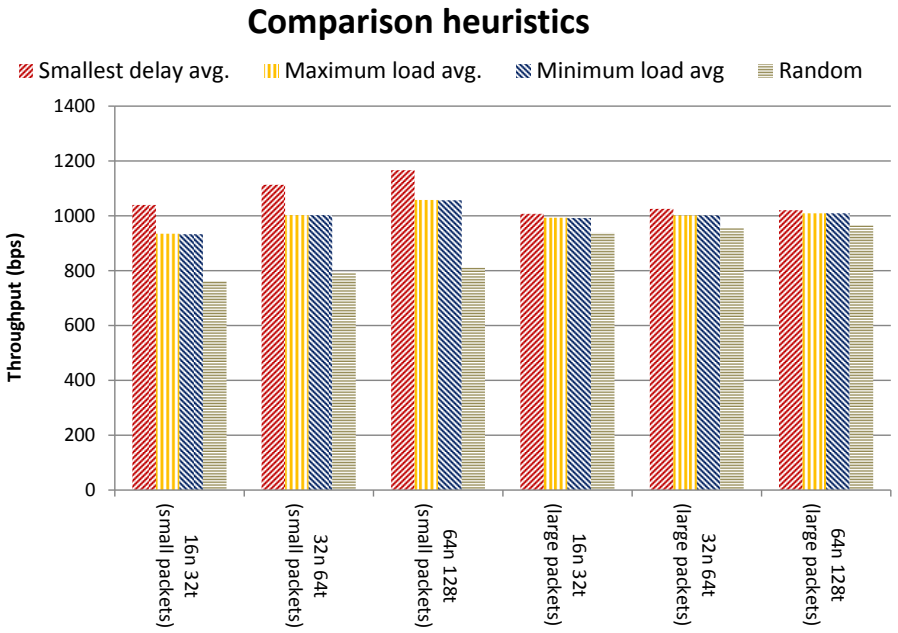
Setup	Slotted avg.	Unslotted avg.	Diff.
16 nodes, 16 transmissions	728.20 bps	930.41 bps	27 %
32 nodes, 32 transmissions	791.01 bps	1037.08 bps	31 %
64 nodes, 64 transmissions	829.24 bps	1113.49 bps	34 %
		Average diff.:	31 %

(b) Setup with nodes scheduled to send small packets (32 bytes)

Setup	Slotted avg.	Unslotted avg.	Diff.
16 nodes, 16 transmissions	954.61 bps	992.43 bps	4 %
32 nodes, 32 transmissions	966.78 bps	1006.53 bps	4 %
64 nodes, 64 transmissions	971.50 bps	1015.26 bps	4 %
		Average diff.:	4 %

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

Figure 3.9: Comparison of a slotted and unslotted scheduling approach. For small packet scheduling the effect of using time-slots when scheduling has a significant impact on the performance of the resulting schedule.



(a) Throughputs plotted for different heuristics

Setup	Min delay	Max load	Min load	Random
16 nodes, 32 transmissions	1039 (1.00)	934 (0.90)	933 (0.90)	762 (0.73)
32 nodes, 64 transmissions	1113 (1.00)	1002 (0.90)	1002 (0.90)	794 (0.71)
64 nodes, 128 transmissions	1166 (1.00)	1058 (0.91)	1057 (0.91)	956 (0.70)

(b) Setup with nodes scheduled to send small packets (32 bytes)

Setup	Min delay	Max load	Min load	Random
16 nodes, 32 transmissions	1007 (1.00)	993 (0.99)	992 (0.99)	938 (0.93)
32 nodes, 64 transmissions	1025 (1.00)	1002 (0.98)	1002 (0.98)	956 (0.93)
64 nodes, 128 transmissions	1021 (1.00)	1009 (0.99)	1009 (0.99)	965 (0.95)

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

Figure 3.10: Comparison of smallest delay, maximum load and naive heuristics scheduling

3.7 Conclusion

Scheduling the transmission in an underwater acoustic communication network can be beneficial in terms of reduced packet loss, energy-consumption and latency and increased throughput. However scheduling the transmissions is not easy. In this chapter 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 yield a schedule that is free of collisions at the receiver and is significantly easier to schedule than existing approaches. 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 30% worse schedule times for small packet size. We have also shown that the heuristic for ordering transmissions (smallest delay first) outperforms heuristics used by other existing scheduling approaches.

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

Bibliography

[1] 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

[2] J. Y. Yang Guan, Chien-Chung Shen, "MAC scheduling for high throughput underwater acoustic networks." submitted to IEEE WCNC 2011, Cancun, Quintana-Roo, Mexico, 2010.

CHAPTER 4

Scheduling for large-scale networks

In this chapter we look at how the computational and communication overhead of Chapter 3 can be reduced to allow scheduling large-scale networks. We present two algorithms for scheduling communications, i.e. a centralized scheduling approach and a distributed scheduling approach. The centralized approach achieves the highest throughput at a reduced computational complexity as compared to the approach presented in Chapter 3, while the distributed approach aims to minimize both the computation as well as the communication overhead. We further show how the centralized scheduling approach can be extended with transmission dependencies to reduce the end-to-end delay of packets in multi-hop networks.

We evaluate the performance of the centralized and distributed scheduling approaches using simulation. The centralized approach outperforms the distributed approach in terms of throughput, however we also show the distributed approach has significant benefits in terms of communication and computational overhead required to setup the schedule.

We propose a novel way of estimating the performance of scheduling approaches using the ratio of modulation time and propagation delay. We show the performance is largely dictated by this ratio, although the number of links to be scheduled also has a minor impact on the performance.

4.1 Introduction

Scheduling approaches such as ST-MAC [1] and STUMP [2] are able to exploit the spatial-temporal uncertainty but do so at the cost of complex scheduling algorithms. In Chapter 3 we have shown how to derive a simplified set of constraints which greatly simplifies the scheduling of underwater communication.

In this chapter we will add an interference rule to the the set of simplified constraints as presented in Chapter 3 and present a centralized and distributed scheduling approach. We present a centralized scheduling approach which reduces the computational overhead and a distributed scheduling approach which reduces the communication overhead to allow scheduling of large-scale networks.

We will also extend the centralized scheduling approach with scheduling dependencies which allows enforcing a certain order of transmission. We show that these scheduling constraints can be used to reduce the end-to-end delay of packets in a multi-hop data collection network by delay the transmission of a parent node until all the packets of children nodes are received.

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

Figure 4.1: Set of simplified scheduling constraints

For performance evaluation, we compare the centralized and distributed scheduling approaches using simulation. We will show that the distributed scheduling approache has significant benefits in terms of communication and computational overhead, while the centralized approach is able to achieve the highest throughput. We also show that scheduling constraints can be used to reduce the end-to-end delay of packets. We present a novel way to estimate the performance of scheduling approaches using the ratio of modulation time and scheduling time.

4.2 Interference rule to allow large-scale scheduling

In this section we will further simplify the set of constraints to allow scheduling in large-scale networks.

The scheduling constraints place restrictions on the transmissions start time of the "to be scheduled" transmissions. We denote the transmissions tasks as δ where a single transmission task i from the complete set of transmission tasks is denoted as δ_i. For each transmission we need to calculate the transmission start time $\delta_i.start$. Every transmission has a certain duration $\delta_i.duration$, source $\delta_i.src$ and destination $\delta_i.dst$. We assume the function T will give the transmission delay between two nodes. This function can be implemented by calculating the distance between two nodes and using the estimated propagation speed to calculate the propagation delay.

The set of simplified scheduling constraints we have derived in Chapter 3 has been shown in Figure 4.1. We can simplify this set of constraints further by making the following observation: When the source of transmission i is the same as the source of transmission j then the equation (4.1) can be rewritten as equation (4.2).

$$T(\delta_i.src, \delta_i.dst) - T(\delta_j.src, \delta_i.dst), \tag{4.1}$$

$$T(\delta_i.src, \delta_i.dst) - T(\delta_i.src, \delta_i.dst), \tag{4.2}$$

Equation 4.2 will always evaluate to 0. The same can be done for the second equation of the maximum from the second rule from the simplified set of constraints. This makes the maximum term of the second rule to be the $max(0, 0)$ when $\delta_i.src = \delta_j.src$ and makes rule one not required.

An interference condition can be added to the set of simplified scheduling constraints. This allows scheduling of large-scale networks where nodes may be outside

$$\begin{cases} \delta_j.start \geq \delta_i.start + \delta_i.duration + max(& \\ \qquad T(\delta_i.src, \delta_i.dst) - T(\delta_j.src, \delta_i.dst), & \text{if } Interfer(\delta_i.src, \delta_j.dst) \\ \qquad T(\delta_i.src, \delta_j.dst) - T(\delta_j.src, \delta_j.dst)) & \\ \delta_j.start \geq \delta_i.start & \text{otherwise} \end{cases} \tag{4.5}$$

Figure 4.2: Scheduling constraints with interference condition allowing scheduling of large-scale networks.

of each others interference range. 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 transmission.

The received signal strength is dependent on the output power of the sender and the attenuation between the sender and the receiver. The attenuation between nodes depends on the absorption rate of the water and the spreading of the signal. This path loss equation [3] can be written as follows:

$$10\log(d, f) = k \cdot 10\log d + d \cdot 10\log a(f) \tag{4.3}$$

The path loss depends on the carrier frequency (f) of the signal as well as the distance (d) between sender, and receiver. The spreading factor is constant, which can either be spherical ($k = 2$), cylindrical ($k = 1$), or something in between. The frequency dependent attenuation is given by the function $a(f)$.

Using this formula we can calculate whether two nodes interfer with each other. Consider two transmissions δ_i and δ_j, which both have 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):

$$Interfer(\delta_i, \delta_j) = \text{TRUE if } (\text{PL}(\delta_j.src, \delta_j.dst) - \text{PL}(\delta_i.src, \delta_j.dst) \leq \text{TH}_{cp}) \tag{4.4}$$

Function (4.4) 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 if $\delta_j.src$ is out of range of $\delta_i.dst$, there is no interference and therefore no constraint between the two transmissions.

The complete set of simplified scheduling constraints, given in Figure 4.2, consists now of the following two scheduling rules:

1. If both transmissions interfere, the interference rule should be used. This rule ensures that the second transmission arrives at the receiver when the first transmission has been received completely.

2. If both transmissions do not interfere, both can be scheduled at the same time.

4.3 Scheduling algorithms

The extended set of simplified constraints, described in Section 4.2, can be applied to design a scheduling algorithm with low complexity. In this section we present the following three algorithms:

- A centralized scheduling approach with high throughput. This approach assumes all information is collected at a central node *prior* to scheduling. Because all information is available at a single point this approach will achieve the *highest throughput*.
- A distributed scheduling approach with *low computational and communication complexity* for large-scale underwater networks. This approach provides a trade-off between the efficiency of the resulting schedule and the amount of communication and computation required to setup the schedule.
- A centralized scheduling approach with transmission dependencies for *low-latency end-to-end communication*. While the first centralized scheduling approach optimizes for throughput, this approach shows how transmission dependencies can be used for optimizing the end-to-end delay.

The centralized scheduling algorithm is a reduced complexity version of the algorithm shown in Chapter 3. The algorithm tries to determine a schedule with minimal time length and therefore aims to achieve the highest throughput possible. The distributed algorithm uses the first algorithm to schedule clusters and allows scheduling of large-scale networks with reduced complexity. Finally the last algorithm is a centralized scheduling approach that uses dependencies between transmissions to optimize the end-to-end delay over multi-hop communication. Rather than trying to achieve the highest throughput possible, this approach minimizes the average time it takes for all packets to travel over multiple hops.

We have chosen these three approaches because we believe they can be used in existing and future UASN applications. The first approach provides a simple approach for scheduling small-scale networks. The second approach shows how scheduling can be done in large-scale networks and provides a balance between network setup overhead and throughput. The last approach can be used to reduce the end-to-end delay of packets or for applications requiring certain transmissions order such as data-aggregation and other distributed processing approaches.

4.3.1 A centralized scheduling approach with high throughput

The extended set of simplified constraints can be applied to design a scheduling algorithm with low complexity for underwater networks. The algorithm from Chapter 3, which has $O(n^3)$ complexity, considers every transmission as the first transmission. To reduce the complexity, we can take a random transmission as the transmission to

be scheduled at time 0. This will reduce the complexity of the algorithm from $O(n^3)$ to $O(n^2)$.

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.

The algorithm continously updates the start time for the unscheduled transmissions. This is done by calculating the maximum of the previously calculated start time and the new start time calculated using the scheduling constraints. It ensures collision free reception by taking the maximum start transmission time.

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, in the second iteration for $n-2$, 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 4.3.

4.3.2 A distributed scheduling approach with low computational and communication complexity

The algorithm presented in Section 4.3.1 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 reduce this communication overhead, 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 in their timeslot. The timeslots can be reused in other clusters and this will ensure that minimal interference occurs between clusters.

Figure 4.4 illustrates an example deployment setup. 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 dependent on the communication range of the nodes. We assume that all nodes in the network use the same output power and carrier frequency 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

Require: $transmissions$ the set of all transmission records
Ensure: $schedule$ is a valid schedule with minimum transmission time.
  $V \leftarrow transmissions$ {Set of all transmissions}
  $schedule \leftarrow [N] \leftarrow 0$ {Resulting schedule}
  $schedule[0] \leftarrow 0$ {Schedule the first transmission}
  $time \leftarrow 0$
  $last \leftarrow 0$
  $index \leftarrow 0$
  $V \leftarrow V \setminus \delta_0$ {Remove transmission from set}
  {Scheduling loop schedules transmissions greedy}
  while $V \neq \emptyset$ **do**
    $time_{min} \leftarrow \infty$
    {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 4.3: Reduced complexity algorithm for scheduling transmissions. The algorithm takes the set of all transmissions ($transmissions$) and schedules the transmissions using a greedy approach. The resulting schedule is stored in $schedule$. The scheduling delays are calculated in the $constraint$ function using the set of scheduling constraints. Different from the algorithm in Section 3.5 this approach does not calculate a schedule for every transmission as the first transmission, rather picks an initial transmission and attempts to schedule the other transmissions as efficiently as possible using greedy scheduling.

radius of the cluster is exactly the size of the maximum communication range. The actual size can be calculated using the path loss expressed in Equation (4.3).

The clusters in our approach are similar to cells in a cellular network. If we assume that 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 4.5 for an example where the number of timeslots can not arbitrarly be chosen and is determined from the following formula [4]:

$$N = i^2 + ij + j^2 \tag{4.6}$$

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

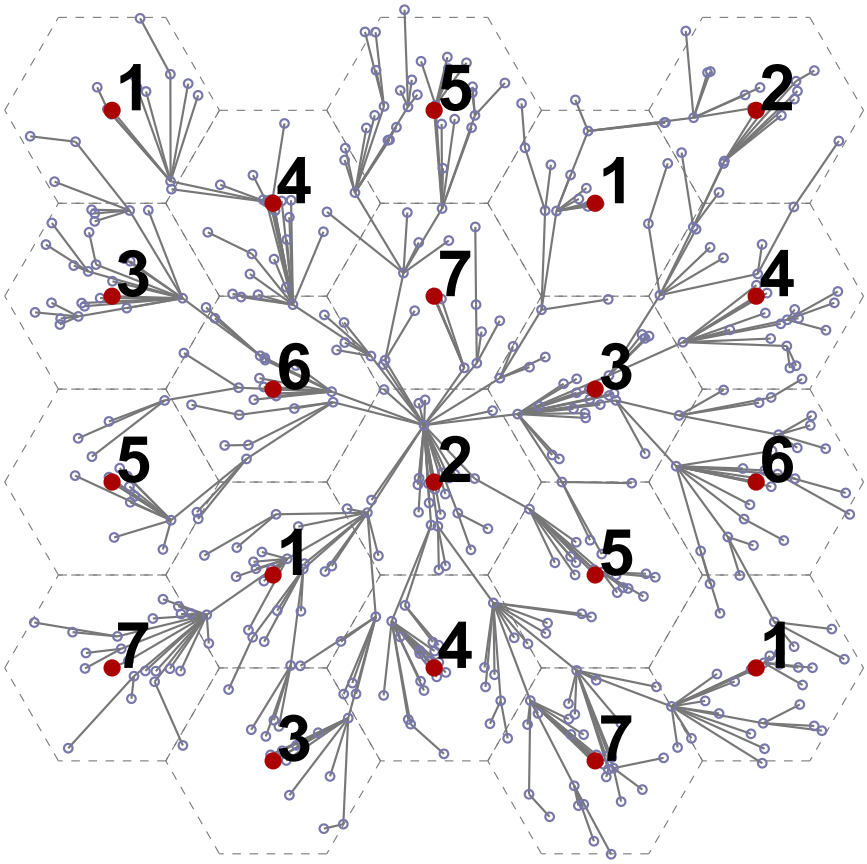


Figure 4.4: Example of a cellular network deployment with timeslot reusage. Timeslots are reused in cells which are spatially separated with a predefined distance dependent on the number of timeslots available.

the cell radius (R) [4]:

$$D = R\sqrt{3N} \tag{4.7}$$

The reuse distance is the minimum distance between two interfering senders in the network. The larger the distance between two interferers, the less interference will be 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 neighboring clusters.

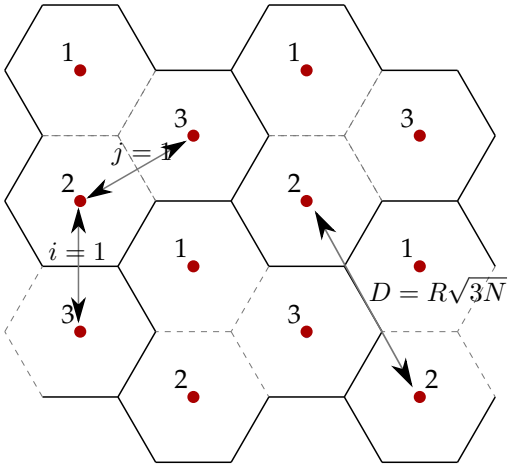


Figure 4.5: Cellular network example. Reuse distance (D) of a timeslot is dependent on the number of timeslots that are used (N) and the communication distance of a transmission R. The number of timeslots (N) can not be arbitrarily chosen but dependent on the i and j parameters. When the number of timeslots is higher the interference of cells using the same timeslot is reduced.

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 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 Chapter 3. We will be using the greedy approach in which transmissions are scheduled based on the minimum delay.

For scheduling the transmissions within a cluster we can use the algorithm from Chapter 3 or the reduced complexity algorithm from Section 4.3.1. The algorithm presented in Section 4.3.1 will yield a smaller computational and memory space complexity, but because the number of transmissions per cluster is in practice limited, the algorithm presented in Chapter 3 may as well be a good option.

Figure 4.6 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 cluster-head determines the maximum of all schedule lengths per slot and assigns the maximum schedule length to the slot. The schedule length and slot lengths are then the only information the central cluster-head needs to communicate to the other cluster-heads.

	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 4.6: Results of calculating slot length based on cluster schedule lengths. Cluster-heads calculate the total length of their timeslot and communicate this to the central scheduler. The central cluster-head will use the maximum slot length of all clusters sharing the same timeslot.

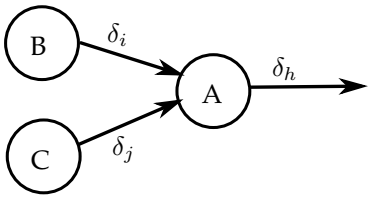


Figure 4.7: Multi-hop scheduled communication. Transmission dependencies can be used to ensure transmission δ_i and δ_j are received before transmission δ_h is scheduled to be transmitted. This reduces the end-to-end delay of packets travelling multi-hop through a networks or allows node A to perform aggregation on the data received from node B and node C.

4.3.3 A centralized scheduling approach with transmission dependencies

The following algorithm is an extension of the approach shown in Section 4.3.1. This approach uses dependencies to force an ordering in the scheduling of transmissions. This can be used to reduce the average end-to-end delay for sending packets over multi-hop connections. An example in which dependencies can be used is shown in Figure 4.7. In this scenario two nodes (B andCB) are sending their packets towards node A and node A is forwarding the (aggregated) packet. In this example it would be beneficial for the end-to-end delay to first schedule the transmissions from A and C in such a way that they are received by A before A starts transmitting. If A receives the packets from B and C after it has transmitted its packet it will have to wait for a new iteration of the schedule to forward the packets from B and C.

To add scheduling to the algorithm of Section 4.3.1, we first have to define a list of dependencies. We store the dependencies in a list D and every dependency is a tuple defining a dependency between two transmissions. In the example shown above, transmission δ_h will have to be transmitted after the reception of transmission δ_i and δ_j. The dependency list D therefore contains tuples (δ_i, δ_h) and (δ_j, δ_h).

Our algorithm for scheduling with dependencies is shown in Figure 4.8. When scheduling with dependencies, the options of possible transmissions are limited by the dependencies. When a transmission still has dependencies it is considered blocked and can not be scheduled. This transmission still has a tuple in the dependency list D. Only the transmissions without dependencies are considered as next transmission and

from all these transmissions we greedily select the next transmission. Once we schedule a new transmission we can remove the tuples of dependent transmissions from the dependency list D, thereby freeing or unblocking possible new transmissions.

The first transmission we schedule is a transmission which has no dependencies. To find this transmission we go through the list of transmissions and find an entry which has no dependency entry in the list D, we do this by finding a transmission δ, which does not have any tuple in D: $(*, \delta) \notin D$. After this, we schedule the first transmission and remove all dependencies for this transmission. Once we have selected the first transmission, we start scheduling the minimum next transmission with no dependency. While determining the minimum transmission to be scheduled we only consider transmissions which have no dependencies. Once the transmission is scheduled, we remove all dependencies related to this transmission. This frees up new transmissions which can be considered during a next round of the scheduling algorithm. We continue until all transmissions are scheduled. Because at every round we consider only transmissions which have no dependencies in the dependency list D, we ensure an ordering of the transmissions.

The dependencies can be derived from the routing algorithm, for example in a data-collection network all data is routed towards a lower hop node until it reaches a single central node in the network. When the parent for a node is selected by the routing algorithm, a transmission is generated from the node to the parent. Next to the transmission also a dependency for this transmission with the parents transmission should be generated. This causes the transmission to be scheduled from the highest hop nodes first to the lower hop transmissions. Care should be taken the dependencies do not contain cycles, this renders the dependencies to be unschedulable.

Require: $transmissions$ the set of all transmission records, $dependencies$ the set of transmission scheduling dependencies.
Ensure: $schedule$ is a valid schedule with minimum transmission time and following the transmission dependencies.
$V \leftarrow transmissions$ {Set of all transmissions}
$D \leftarrow dependencies$ {Set of all dependencies}
$schedule \leftarrow [N] \leftarrow 0$ {Resulting schedule}
{Find the first transmission that can be scheduled}
for $\delta \in V$ **do**
  if $(*, \delta) \notin D$ **then**
    $index \leftarrow \delta_{index}$
  end if
end for
$schedule[index] \leftarrow 0$ {Schedule the first transmission}
$time \leftarrow 0$
$last \leftarrow index$
$D \leftarrow D \setminus (\delta, *)$ {Remove all dependencies}
$V \leftarrow V \setminus \delta_{index}$ {Remove transmission from set}
{Scheduling loop schedules transmissions greedy}
while $V \neq \emptyset$ **do**
  $time_{min} \leftarrow \infty$
  {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 $(*, \delta) \notin D$ and $schedule[\delta_{index}] < time_{min}$ **then**
      $time_{min} \leftarrow schedule[\delta_{index}]$
      $index \leftarrow \delta_{index}$
    end if
  end for
  {Schedule transmission with smallest starting time first}
  $time \leftarrow time_{min}$
  $last \leftarrow index$
  $V \leftarrow V \setminus \delta_{index}$
  $D \leftarrow D \setminus (\delta, *)$
end while

Figure 4.8: Reduced complexity algorithm for scheduling transmissions with dependencies. Transmissions are scheduled again in a greedy approach, however transmissions are not scheduled before all dependencies are resolved. Initially a transmission is searched which has no dependencies, this transmission is scheduled first. Next, the transmission which has the smallest delay and has no dependencies is scheduled. Once a transmission is scheduled, any tuple dependent on the transmission is removed from the set of transmissions and new transmissions may be freed for scheduling.

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 4.9: Complexity of different scheduling approaches compared.

4.4 Evaluation of communication and computation complexity

To evaluate the different centralized and distributed scheduling approaches, we will first discuss briefly their complexity in terms of number of communications required as well as computational complexity of different approaches. An overview of the complexity of all scheduling approaches can be seen in Figure 4.9.

- Centralized Scheduling: In this case we assume all transmissions as well as position information are 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 Chapter 3, whose complexity is $O(n^3)$.
- Reduced Complexity Centralized Scheduling: This is the algorithm described in Section 4.3.1. 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. This results in $O(hops_{avg}k)$ number of 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

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:	2 x 2	3 x 3	4 x 3
Area size:	$\approx 1.0x1.0km$	$\approx 1.5x1.5km$	$\approx 1.5x2.0km$
Nodes:	100	200	500

(b) Different deployment sizes

Figure 4.10: Simulation parameters.

approaches. The scalability of the distributed approaches is therefore much better than the centralized approaches.

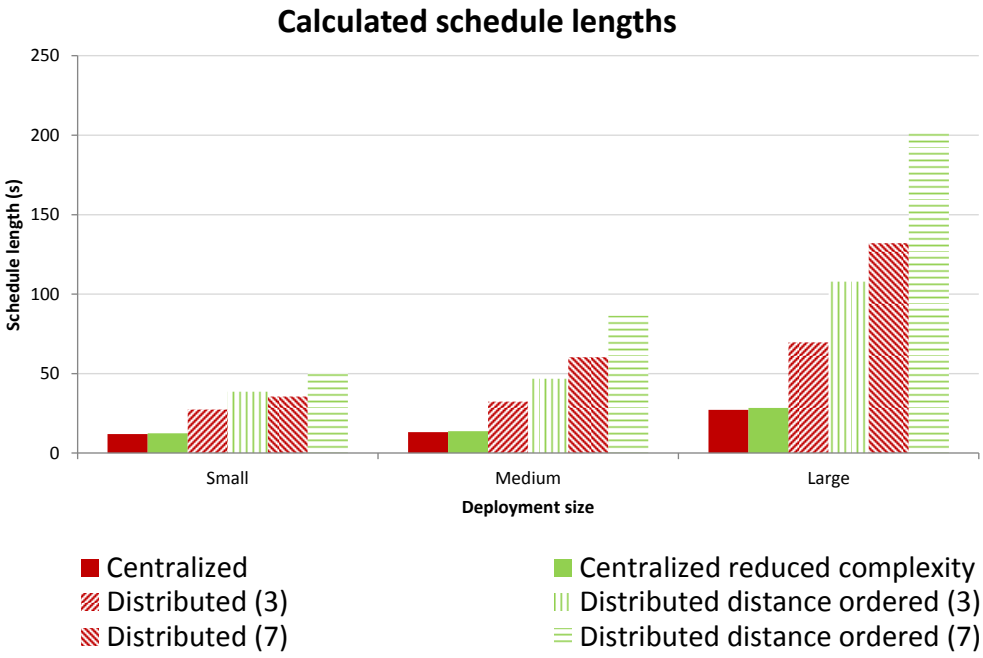
4.5 Evaluation of scheduling efficiency

We evaluated the algorithms for different sizes of deployments. The parameters can be found in Figure 4.10(a). The network size ranges from 100 up to 500 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 illustrated in Figure 4.4.

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. 3 timeslots is the minimum number of timeslots required and the reuse distance in this case will be exactly the interference range. Using 7 timeslots increases the reuse distance beyond the interference range, this provides a guard band for when the interference range in reality can not be that accurately estimated.

The evaluation results are shown in Figure 4.11. We see that the centralized approach performs the best as 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 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 4.3.1 and Section 4.4 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



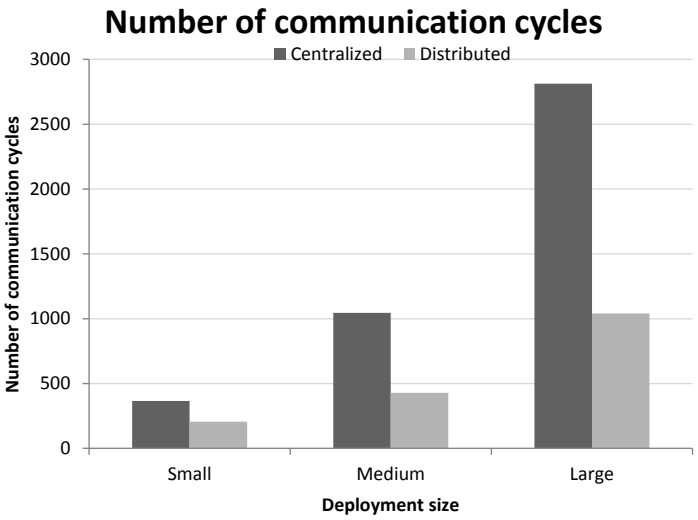
	Network size		
	Small	Medium	Large
Centralized	11.99	13.12	27.23
Centralized reduced complexity	12.46	13.73	28.45
Distributed (3)	27.43	32.38	69.62
Distributed distance ordered (3)	38.60	46.79	107.80
Distributed (7)	35.56	60.29	132.05
Distributed distance ordered (7)	50.31	86.38	201.44

Figure 4.11: Schedule length of different scheduling approaches in different network sizes. Scheduling approaches are run from small networks (100 nodes) to large networks (500 nodes). Compared are centralized scheduling approaches and distributed scheduling approaches with respectively 3 timeslots and 7 timeslots. More timeslots result in a large timeslot reuse distance, improving performance when the interference range of the transmissions are underestimated.

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.

The schedule lengths of the distributed approach are on average 244% of the centralized approach when 3 timeslots are used, and 414% when 7 timeslots are used. This shows that when the scalability, computational and communication overhead is



	Scheduling approach	
Network size	Centralized	Distributed
Small	365	205
Medium	1045	428
Large	2813	1040

Figure 4.12: Number of communication cycles required to setup a single schedule for different scheduling approaches and network sizes.

not a concern a centralized approach is still much preferred because it yields the most efficient schedule.

In Figure 4.12 the amount of communications cycles required to set up the network is shown. The difference between the centralized and distributed approach can be seen quite clearly. The centralized approach does not scale very well to large network 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 independent 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.

4.5.1 End-to-end delay

A criteria for optimization, next to the criteria of optimizing for throughput, may be the time it takes for a packet to travel from the source node to the central node. In this section we will look at this end-to-end delay and we will look at how scheduling dependencies can be used to reduce this end-to-end delay. We have simulated the network in the same setup as before. However in this scenario we have setup the

transmissions to aggregate the result of the child nodes. Using the shortest hop distance routing algorithm we determine for every node a parent. Every parent will have to send a packet of size 32 bytes plus the number of childs times 32 bytes. So every node is able to send its own data and forward the data from all its childs. The total size of data a parent will have to send depends on the number of children (n) as follows:

$$size_{total} = 32 + 32 * n \tag{4.8}$$

In this scenario we look at the length of a single run of the schedule, but we will also look at how long it takes for every packet to travel from the source node to the central node. If no dependencies are used, a packet may be in the network for several subsequent runs of the schedule. If dependencies are used, the parent will wait for all packets to arrive from the children and then starts transmitting his packet and forwards all of his children packets. The results are shown in Figure 4.13.

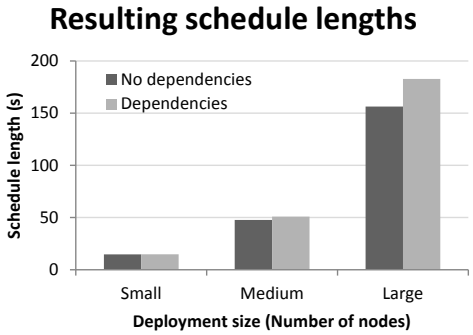
What can be seen from these results is that when scheduling with dependencies, the schedule length is increased. This results in a lower throughput. However the average end-to-end delay is decreased because the packets can all be delivered to the central node in a single run. One can see that for the small network the difference in end-to-end delay is not that substantial. This is because in these small networks the number of hops a packet has to travel is small and the length of a schedule run is still short. For medium and large networks the difference is substantial.

Looking at the results, one should consider the application and whether it makes sense to optimize for end-to-end delay. Considering that quite large networks with large number of hops need to be constructed before the difference becomes noticable. One scenario where using dependencies does make sense is when a very low duty cycle is used. The network may sense, send data and then go to sleep for a considerable time. For example the network may sense at a rate of every 10 minutes or every hour. In this scenario it would make sense to optimize for end-to-end delay, because every subsequent run of the schedule may add a delay of 10 minutes or an hour. Packets that require multiple runs of the schedule before being delivered add a delay of many minutes between every iteration of the schedule.

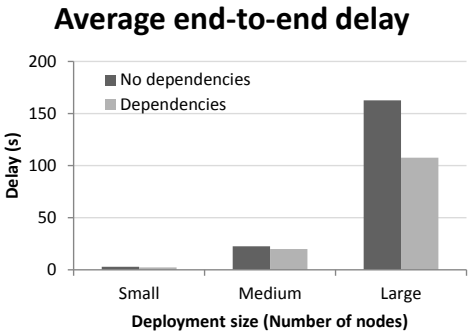
Another scenario for optimizing the end-to-end delay would be when distributed processing such as aggregation is used. In such a scenario a parent can not send before it has received all data from its children. In this situation the ordering of transmissions is required and the scheduling algorithm with dependencies can be used to achieve a good throughput for the communication in such a network.

4.5.2 Scheduling efficiency at different propagation time / modulation time ratio

In this section we evaluate the performance of scheduling independent of the size of the deployment or the modulation rate used at the low-level modem. To do so, we realise that the two most important factors of the performance of the schedule are (I) the propagation time between nodes and (II) the modulation time of (the time to transmit) the packets. We can simulate the scheduling approach under different sizes



(a) Schedule lengths



(b) End-to-end delay

	Network size		
	Small	Medium	Large
No dependencies	14.65	47.68	156.32
Dependencies	14.79	50.84	182.74
Increase	0.9 %	6.6 %	16.9 %

(c) Schedule lengths (seconds)

	Network size		
	Small	Medium	Large
No dependencies	2.92	22.58	162.67
Dependencies	2.41	19.93	107.65
Reduction	17.5 %	11.7 %	33.8 %

(d) End-to-end delay (seconds)

Figure 4.13: Example of how scheduling with dependencies can improve the end-to-end delay with a small increase in schedule length.

of deployments, different packet sizes and different modulation rates, however we can also simulate independent of these parameters by taking the modulation time / propagation time ratio. We define this ratio as follows:

$$ratio_{scheduling} = \frac{time_{modulation}}{time_{propagation}}$$

To give an example, say we want to send 125 byte packets at a modulation rate of 1000bps. The modulation time for a packet in this example is 1 second. Say the average distance between nodes in our network is 1500 meters, this is an equivalent of a propagation time of 1 second. In this scenario our scheduling ratio is 1.

In another scenario in which we want to send 125 byte packets at a modulation of 10kbps, the modulation time for a packet is 100ms. Having an average distance of

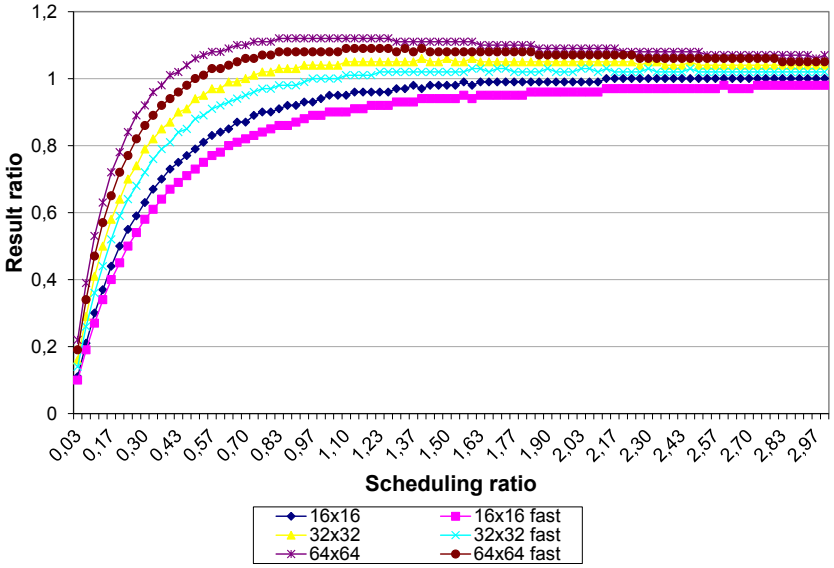


Figure 4.14: Scheduling at different ratios

150m between nodes results in the same scheduling ratio.

To define a result value independent of the modulation time, we use the following ratio:

$$ratio_{result} = \frac{throughput_{schedule}}{throughput_{modem}}$$

We look at the resulting throughput of the schedule in relation to the throughput available of the modem. This makes the result independent of the chosen modem throughput. We calculate the schedule with different scheduling ratios, for a deployment of nodes uniformly deployed. We simulate this for different number of nodes: 16 nodes, 32 nodes and 64 nodes.

Interesting to see is that there is a certain optimum for the performance around a scheduling ratio of 1. When going to lower scheduling ratios, the efficiency starts to decrease rapidly. In these scenarios the propagation time takes the overhand in the schedule and the schedule effectively becomes sparse. When going to the higher scheduling ratios than 1, the efficiency of the schedule decreases a little bit but seems to converge to a result ratio of 1.

What can be seen from Figure 4.14 is that when the propagation time takes the overhand, the efficiency of communication in terms of bandwidth starts to decrease. We have shown this for scheduled MAC protocols but the result may also be valid for unscheduled MAC protocols in the underwater environment applying some sort of collision avoidance. Because of this relation it makes sense to aggregate more packets into a single transmission if the propagation delays are large, or switch to a lower modulation rate. This can be done without losing too much efficiency.

Another observation from Figure 4.14 is that for different number of nodes the performance slightly differs. We tried to define an indicator for the performance of the scheduling using the modulation time and the propagation time. However from the results it becomes clear that this does not fully describe the performance. The ratio gives a good indication on what the expected performance will be, but when more nodes and therefore more links are in the network the performance does slightly increase. We believe this is because when more links are available the greedily approach of the scheduling algorithm works better because at every step it has more options to choose from.

Using ratios we determined the efficiency of communication scheduling independent of data-throughput of the modem, data packet sizes and node distances. The performance of the schedule is dependent on the number of links. We have shown that communication scheduling achieves the highest throughput when the ratio modulation time and propagation time is 1. This shows that when a certain average distance is dictated by a deployment, the modulation time and data size should be selected accordingly.

4.6 Conclusion

Scheduling algorithms for underwater communication allow 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 chapter we introduced a centralized and a distributed scheduling technique for underwater acoustic communication systems. The centralized approach achieves the highest throughput of all scheduling approaches but does this at the cost of high computational and communication overhead.

The distributed approach groups all transmissions in the cluster 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 sink 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.

Comparing communication and computational complexity of the proposed algorithms shows that the distributed approach is much more scalable to large networks. Ita also shows that to allow scheduling of large-scale networks a distributed approach is required. 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 distributed approach results in decreased throughput and should only be used when energy-consumption and communication overhead is a concern.

We have also introduced an approach to allow scheduling with dependencies

between transmissions. This allows to restrict scheduling transmissions in a certain order. We have used this to schedule transmissions from higher hop nodes before transmissions from lower hop nodes. This allows aggregation of data from the outside of the network to a central data collection node at the center of the network and can be used to reduce the end-to-end delay of packets in a data-collection network. In simulation the end-to-end delay is improved up to 33.8 % at the cost of increasing the total schedule length by up to 16.9 %.

We presented a novel way of estimating the performance of scheduling based on the ratio of modulation time and propagation time. We have shown that this ratio gives a good indication of the expected performance even though the number of links also has a small impact on the performance. To obtain good scheduling results, the ratio between propagation delay and modulation time should be around one.

Bibliography

[1] 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.

[2] P. M. Kurtis Kredo II, "Distributed scheduling and routing in underwater wireless networks," *Globecom 2010*, 2010.

[3] D. E. Lucani, M. Stojanovic, and M. Médard, "On the relationship between transmission power and capacity of an underwater acoustic communication channel," *CoRR*, vol. abs/0801.0426, 2008.

[4] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

Part II

Localization and Time-Synchronization

CHAPTER 5

Underwater Localization by combining ToF and DoA

In this chapter we present a combined ToF and DoA localization approach suitable for shallow underwater monitoring applications such as harbor monitoring. Our localization approach combines one-way ranging and DoA estimation to calculate both position and time-synchronization of the blind-node. We will show that using this localization approach, we are able to reduce the number of reference nodes required to perform localization. Our approach is also capable of tracking and positioning of sound sources under water.

We evaluate our approach through both simulation and underwater experiments in a ten meter deep dive-center (which has many similarities with our target application in terms of depth and reflection). Measurements taken at the dive-center show that this environment is highly reflective and resembles a shallow water harbor environment. Positioning results using the measured Time-of-Arrival (ToA) and DoA indicate that the DoA approach outperforms the ToF approach in our setup. Investigation of the DoA and ToF measurement error distributions, however, indicate the ToF-based localization approach has a higher precision. Shown is that both ToF and DoA and the combined approach achieve sub-meter positional accuracy in the test environment.

Using the error distributions derived from the measurement in the dive-center, we run simulations of the same setup. Results from the simulation indicate ToF is more accurate than DoA positioning. Also in simulation all approaches achieve sub-meter accuracy.

5.1 Introduction

Accurate localization under water is a challenging task. To estimate the position of a blind node, underwater localization systems use either two-way acoustic ranging between blind and reference nodes or one-way acoustic ranging with synchronous clocks (on both the blind and the reference nodes) [1] [2].

While range-based localization systems use timing information and knowledge about the propagation speed to calculate distance estimates, DoA-based systems use angular information of the incoming signal to determine the position of the blind nodes. In this chapter, we focus on one-way ranging with synchronous reference nodes and an asynchronous blind node. Such an approach improves scalability and

reduces the amount of energy consumed by the overall system because of reduced communication. For UASN we consider this important aspects because it allows scaling to large number of nodes. Our localization approach combines ToF and DoA information to calculate position and to provide time-synchronization on the blind nodes. Using both ToF and DoA, we are able to reduce the number of reference nodes required to perform localization and will achieve higher localization accuracy. By combining ToF and DoA, our approach is also capable of tracking sound sources and positioning sound sources under water.

The target application of our localization approach is shallow underwater monitoring applications such as monitoring harbor areas, which are located at shallow water reservoirs where many reflections are present. We evaluate our approach through both simulation and underwater experiments in a ten meter deep dive-center (which has many similarities with our target application in terms of deepness and reflection).

The rest of this chapter is organized as follows. In Section 5.2 we introduce our ToF, DoA and combined ToF and DoA localization approaches. An overview of the experimental setup at the dive-center and the data acquisition testbed can be found in Section 5.3. The results of our experimental study are discussed in Section 5.4. Using the error distributions derived from the experimental setup, we run the simulation as presented in Section 5.5. Finally, in Section 5.6 we summarize the results in our conclusion.

5.2 Underwater localization

In this section we will introduce our ToF, DoA and combined ToF and DoA localization approaches. We start by introducing the time-synchronization approach and will continue by describing our position estimation approaches.

Figure 5.1 shows an example of how localization can be performed using DoA or ToF information. Reference nodes send out beacon messages which are received by the blind-node. With DoA localization, the angle of the incoming signal at the blind node is estimated and used to calculate the position of the blind node (x, y). With ToF localization, the difference between the transmission time of the signal (s_1, s_2 and s_3) and reception time (r_1, r_2 and r_3) is recorded and used to calculate both the position (x, z) as well as the clock bias (b) of the blind node.

5.2.1 Time synchronization

Time synchronization is the process of synchronizing the clocks of different nodes. This notion of 'time' does not necessarily have to be global over all nodes and nodes in the network can agree on a local time for the complete network.

Let us assume the clock bias (or phase) of the clocks of the reference nodes $i = 1 \dots N$ are denoted by ϕ_i and the clock of an unsynchronized blind node is denoted by ϕ_b. For the ToF measurement we assume that the reference nodes are synchronized and the blind node has an unknown clock bias denoted by b. For four reference nodes and one blind node, we can write this as the following equation:

$$\phi_1 = \phi_2 = \phi_3 = \phi_4 = \phi_b + b \tag{5.1}$$

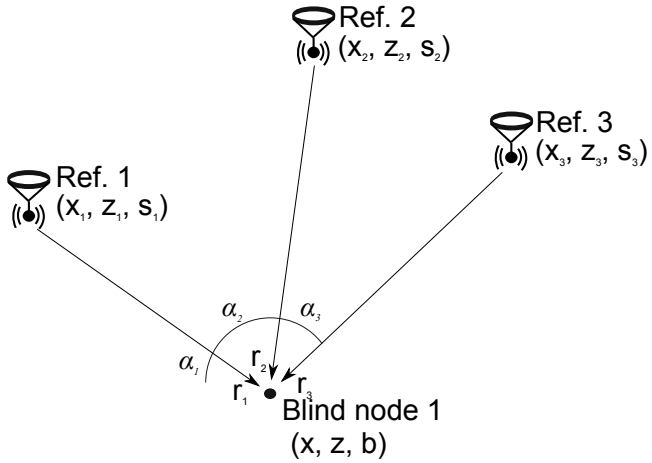


Figure 5.1: Example of our DoA and ToF scenario. In our scenario we attempt to calculate the x displacement and depth z of our blind node using DoA and ToF localization. DoA localization uses the angles of incoming signals from reference nodes (α_1, α_2 and α_3), ToF uses the difference between sending time (s_1, s_2 and s_3) and receiving time of the signal (r_1, r_2 and r_3) to estimate the position of the blind node. In this scenario the orientation of the blind node is known and absolute angle measurements can be used.

The process of time synchronization is determination of the clock bias b of the blind node. We assume there is no clock-skew and the clock bias is constant. In reality, there is clock-skew. However, because the measurements are taken in a very short time period, we consider the clock-skew to have minimal impact.

5.2.2 Localization

To perform localization, ToF and DoA estimates can be used separately or combined. This section describes these three localization methods. We assume that the positions of the four reference nodes are known and the position of reference node i is denoted by (x_i, z_i). The goal of the localization algorithm is to determine the unknown position of the blind node (x, z) and the unknown clock bias b.

ToF based localization

Given the start time s_i of the reference transmission from TX_i and its measured arrival time r_i at the (unsynchronized) blind node, time synchronization and position estimation can be performed simultaneously. This is done by minimizing the following least-squares estimation:

$$\epsilon_{\text{ToF}} = \min_{(x,z,b)} \sum_{i=1}^{N} \left(\sqrt{(x-x_i)^2 + (z-z_i)^2} - v \cdot (s_i - b - r_i)\right)^2 \tag{5.2}$$

Herein, b is the clock bias of the blind node and assume v is a constant propagation speed of the signal in water ($\approx 1491\text{m/s}$). The unknown parameters (x, z, b) can be

estimated using an iterative optimization approach such as Levenberg-Marquardt [3], also described in Appendix B.

DoA based localization

To perform localization based on DoA, we fit the parameters (x, z) to the angles α_i determined by the DoA estimation. To do so, the following equation needs to be minimized:

$$\epsilon_{\text{DoA}} = \min_{(x,z)} \sum_{i=1}^{N} (\tan^{-1}(\frac{x - x_i}{z - z_i}) - \alpha_i)^2 \tag{5.3}$$

Angular information is not dependent on the clock bias of the blind node. Using angular information we are only able to estimate the position of the node and the equation only minimizes x and z.

Combined ToF and DoA based localization

The combined ToF and DoA approach requires minimization of the following equation:

$$\epsilon = w_{ToF} \cdot \epsilon_{\text{ToF}} + w_{DoA} \cdot \epsilon_{\text{DoA}}. \tag{5.4}$$

For combining the ToF and DoA measurements, weights for the different cost criteria (w_{ToF} and w_{DoA}) need to be determined.

Figure 5.2 shows the error distribution of the ToF and DoA measurements. The ToF measurements are assumed to be normally distributed $\mathcal{N}(\mu_{ToF}, \sigma^2_{ToF})$. For the distribution of the angular information an extra step is needed to convert it to a positional error. The distribution of the angular information σ_{DoA} can be derived from the measurements, the positional error distribution can be calculated using the distance d and actually follows the curve shown in Figure 5.2.

Theoretically, the correct way of calculating the most likely position would be to use a Maximum Likelihood Estimation (MLE) and to model the distribution of the angular error using polar coordinates, as has been done by Peng and Sichitiu [4]. Because our distances are small (as this is the case in the harbor monitoring application), we simplify the error model using a straight line rather than a curve. The error distribution of the angular information is modelled as:

$$\mathcal{N}(0, (d \tan(\sigma_{DoA}))^2) \tag{5.5}$$

The error model of the angular information is dependent on the distance estimation d, the distance between the estimated position and the reference nodes, and needs to be updated after every estimation step. Weights for the combined optimization are selected based on the variance of the error distributions of the measurements $w_{ToF} = \frac{1}{\sigma^2_{ToF}}$ and $w_{DoA} = \frac{1}{(d \tan(\sigma_{DoA}))^2}$. Because we use an iterative optimization approach it is straightforward to update these weights after each optimization step.

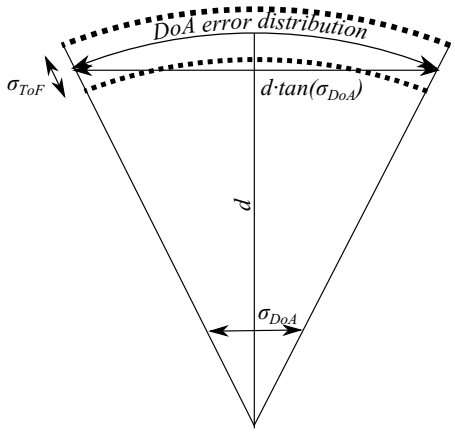


Figure 5.2: Error distribution of the ToF and DoA measurement. Positional error of a DoA measurement can be calculated by using the σ_{DoA} and using the (estimated) distance between the blind node and the reference node (d). The positional error is distributed as $\mathcal{N}(0, (d\tan(\sigma_{DoA}))^2)$.

5.3 Experimental study

To evaluate the performance of our combined localization approach we have performed underwater experiments in a ten meter deep dive-center. An overview of the setup used in our underwater localization experiment is presented in Section 5.3.1.

5.3.1 Dive-center experiment

During the dive-center experiment, reference transmissions were transmitted sequentially from four transducers attached to a tube floating half a meter below the water surface. At the bottom of the tank, at ten meters of depth, a four-element array (our blind node) captured the reference transmissions. The received signals were recorded using a multi-channel underwater testbed.

Figure 5.3 shows a technical drawing of the setup, the four reference transducers are denoted by TX_1 to TX_4 with their respective coordinates:

$$(x_1, z_1), (x_2, z_2), (x_3, z_3), (x_4, z_4)$$

The goal of the localization algorithm is to determine the unknown location (x, z) of the array at the bottom of the tank. In terms of UASN, the reference transducers can be considered as reference nodes and the array as blind node. Figure 5.4 shows a segmented panorama of the actual test setup in the dive-center.

The multi-channel acoustic signal processing testbed described in [5] has been used to capture the signals of the four-element array. The testbed consists of an off-the-shelf FPGA board and a data-acquisition extension board. A simplified schematic of the testbed is shown in Figure 5.5. Underwater acoustic pressure waves are converted to voltages by the four piezoelectric transducers of the array. After amplification and

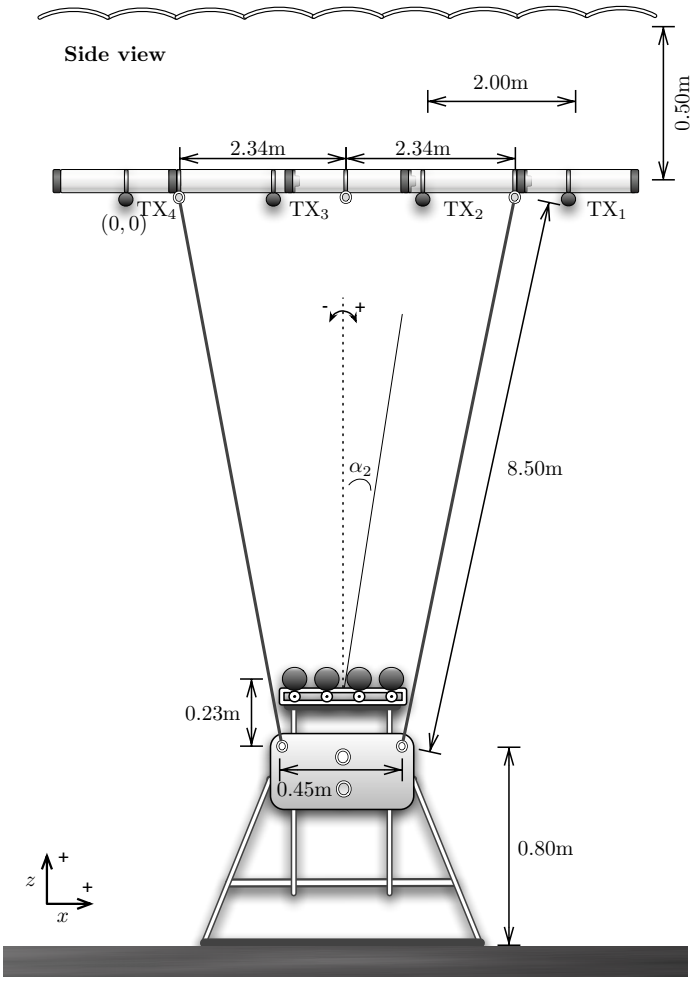


Figure 5.3: Technical drawing of dive-center setup (not to scale).

filtering, all analog signals are synchronously sampled and copied to the on-board memory of the FPGA board. After each experiment, raw sample data was copied from the on-board SDRAM to a PC for offline post-processing.

5.3.2 Methodology

The reference transmissions are frequency sweeps from 20khz to 40kHz. Each sweep has a duration of 90ms and is followed by 10ms of silence. After each sweep the active transducer is switched.

Based on the multi-channel recording of the array, the ToF and the DoA of a reference transmission can be determined. The ToF is found by cross-correlation of the received signals and the reference signals. A linear beamscan, in the 20khz to

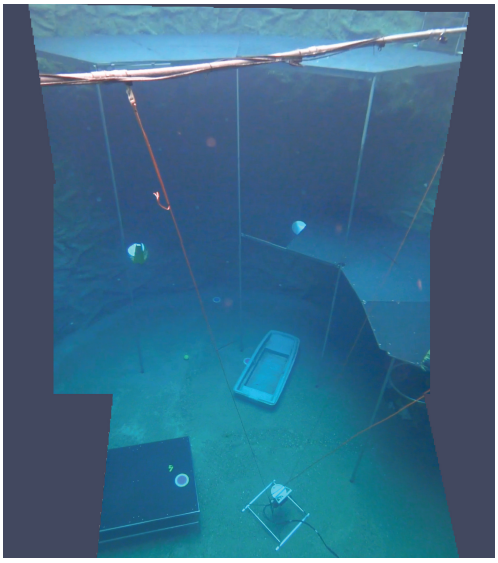


Figure 5.4: Segmented panorama of the dive-center setup.

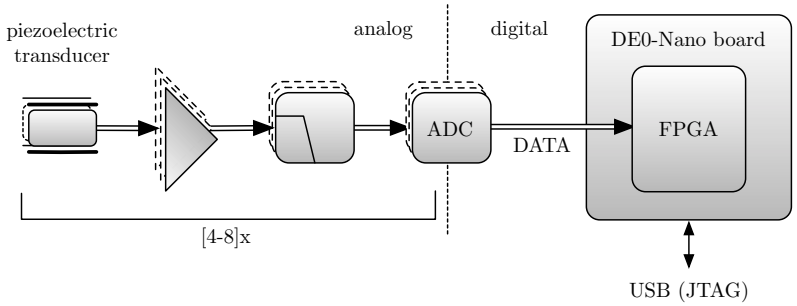


Figure 5.5: Overview of the multi-channel underwater testbed.

22kHz subband, is used to find the DoA of an incoming reference transmission.

5.3.3 Acoustic Ranging

The acoustic range is a distance estimate based on the ToF and the acoustic propagation speed v in the medium. To determine an estimate of the ToF an approximation of the arrival time of the reference signal is required. The (discrete-time normalized) cross-correlation $c_{sr}[l]$ of the transmitted reference signal (s) can be used to accurately determine the arrival time of the reference signal by correlating with the received signal (r) received by the bottom mounted array:

$$c_{sr}[l] = \frac{1}{\sqrt{c_{ss}[0]c_{rr}[0]}} \cdot \sum_{n=1}^{N-l} s[n]r[n+l], \qquad l \geq 0. \tag{5.6}$$

Herein, l is the lag parameter, N is the number of samples of the input sequences and $c_{ss}[0]$ and $c_{rr}[0]$ are the mean-square signal powers of s and r. The lag time of the Line-of-Sight (LoS) arrival, in terms of samples, can be found by application of a simple threshold on the baseband representation of the cross-correlation results. We assume that the first peak having a correlation of at least 50% of the maximum correlation represents the LoS path.

Given the sample frequency of s and r, the lag time of the LoS arrival is rounded to find an estimate of the arrival time.

5.3.4 DoA Estimation

The four-element transducer array is used to estimate the DoA angle of a reference transmission. In our experiment, the acoustic bandwidth of the sweep signal is in the order of its carrier frequency. Decomposing this wideband sweep into multiple 'narrow' subbands validates the use of narrowband beamforming to create directivity [6]. Narrowband beamforming (based on phase shifts) in the 20-22 kHz subband has been used to determine the signal power in a certain direction. A linear scan over all broadside look angles is performed to determine the signal power for every angle. The latter is known as the spatial spectrum. The angle corresponding to the strongest signal power is considered the actual LoS arrival angle.

5.4 Experimental results

A typical cross-correlation and lag time from a received underwater transmission in the test environment is shown in Figure 5.6. What can be seen from this cross-correlation is that many multi-path signals are present in the environment. Moreover some of the multipath signals were received stronger than the LoS signal. These are the same difficult conditions attributed to shallow water and high reflection that we expect to face for shallow harbor monitoring. Successful and accurate localization in such environment can also guarantee the accuracy and success of our approach for harbor monitoring.

Figure 5.7 shows the results of the spatial spectra for four reference transmissions. Based on the linear beam scan, we found the mean DoA angles of the reference transducers ($TX_4 \ldots TX_1$) to be -17.9, -3.86, 10.66 and 20.86 degrees.

Using the measurements we have determined the error distribution for ToF to be $\sigma_{ToF} = 0.1219 \cdot 10^{-5} m$. For the DoA the error distribution of the measured angles is $\sigma_{DoA} = 0.1817°, 0.1140°, 0.0548°, 0.0548°$. If we use the knowledge that the distance between the array and the beacons is about 8.3514 meters, the error distribution of the angular localization is approximately $d \cdot \tan(\sigma_{DoA}) = 2.65 \cdot 10^{-2} m, 1.66 \cdot 10^{-2} m, 0.80 \cdot 10^{-2} m, 0.80 \cdot 10^{-2} m$. Using this information we can conclude that according to these measurement the ToF measurement is more accurate than the DoA measurement.

Figure 5.3 shows that the center of the array is, in theory, positioned at $(x = 3.00, z = -8.05)$. Based on ToF estimates for the reference transmissions, the position of the array is estimated. The results of ToF-based localization (for forty-eight distinct

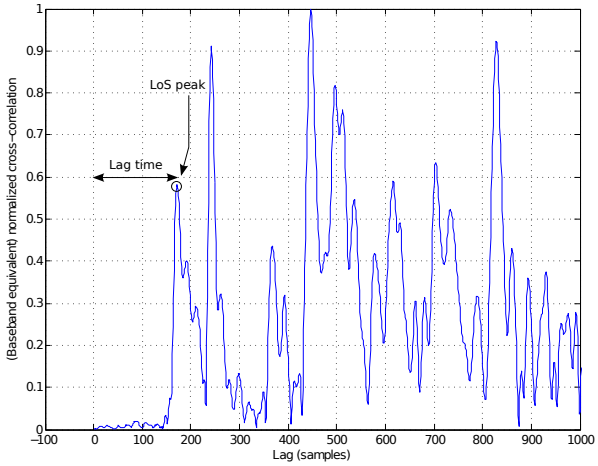


Figure 5.6: Cross-correlation results of acoustic reference sweep.

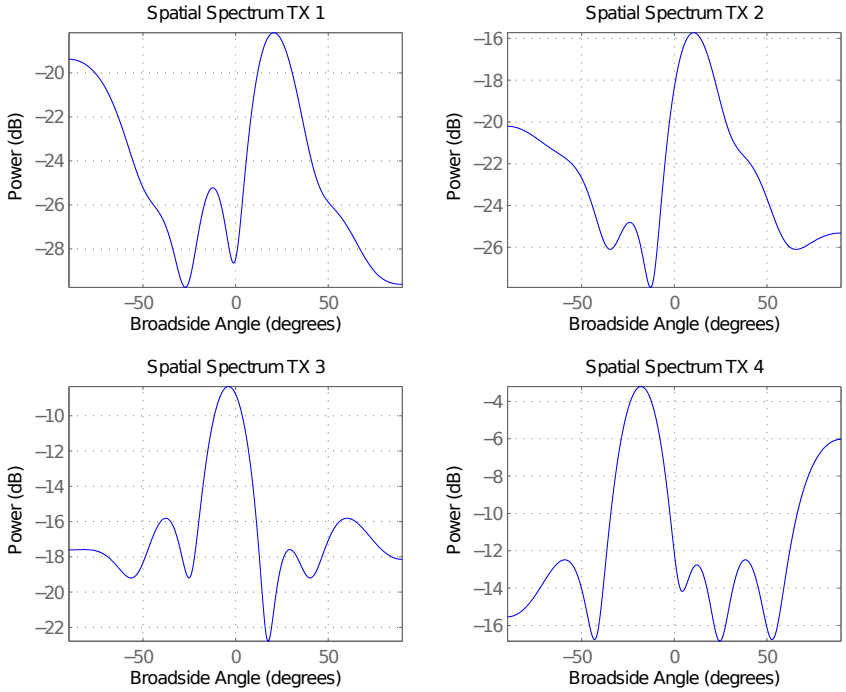


Figure 5.7: Measured spatial spectra.

measurements) and the DoA-based localization (for twelve measurements) are shown in Figure 5.8.

Results of both localization methods indicate that the array was located to the left of the calculated deployment position. When we look at the distance between

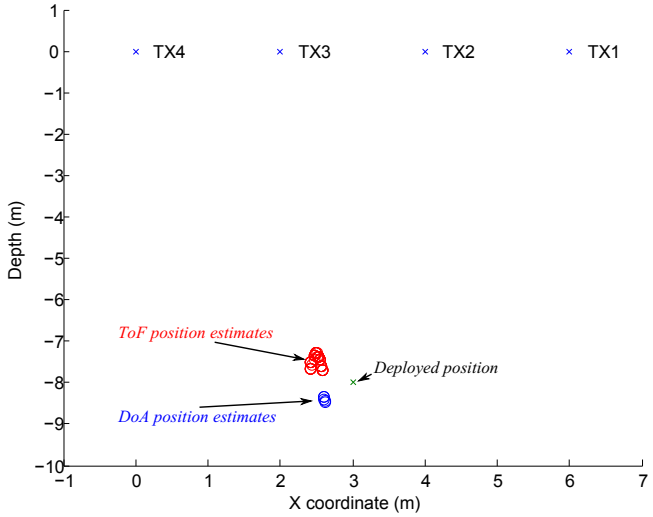


Figure 5.8: Result DoA and ToF-based localization. ToF underestimates the depth of the array and places the positions above the array, DoA overestimates the depth of the array and places the positions under the array. Results indicate an average position error of $71cm$ for ToF and $56cm$ for DoA positioning.

the ToF calculated positions and the deployed position of the array, we can see that the calculated ToF position is placed on average $71cm$ from the deployed position of the array. The calculated positions from the DoA position are placed $56cm$ from the deployed position of the array.

The error distribution of the ToF position is $\mathcal{N}(0.71, 0.12)$. The error distribution of the calculated positions for DoA is $\mathcal{N}(0.56, 0.03)$. This indicates that in this setup the DoA position calculation is more accurate than the ToF position.

We tried to position the array centered underneath the tube with reference transducers. However, this task turned out to be fairly difficult under water. A slight mis-alignment was visually observed during the experiment and the array was indeed actually more positioned to the left. Therefore, in reality the calculated positions were actually more accurate than the $71cm$ and $56cm$ originally calculated. This is in contradiction with results we have obtained before, so we cannot conclude whether the DoA or ToF approach is the most accurate. From these results we can conclude, however, that both methods achieve sub-meter accuracy. Note that ToF-based localization seems to underestimate the actual depth while DoA-based localization overestimates the depth. An explanation of the conflicting results may be that the actual depth of the array was different (shallower) than where we tried to position the array.

Results for combined ToF and DoA localization can be seen in Figure 5.9. The combined localization approach pulls completely to the ToF result. This is the result of the measured σ_{ToF} and σ_{DoA}. The standard deviation of the ToF error distribution is significantly smaller than the standard deviation of the DoA distribution, weights for the ToF measurement are orders of magnitude larger than the DoA weights.

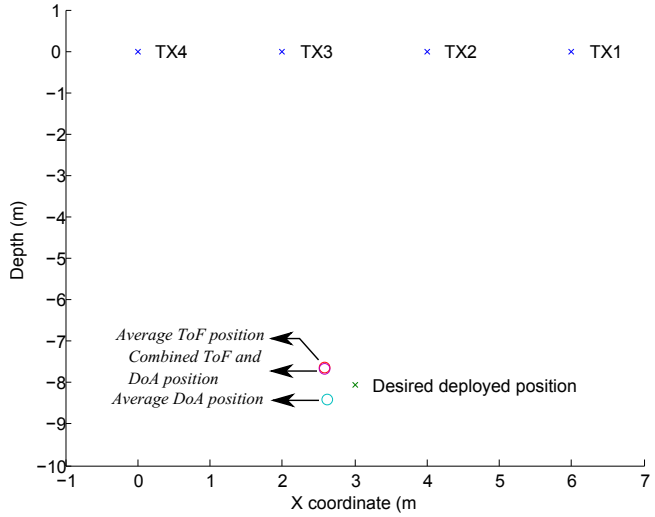


Figure 5.9: DoA, ToF and combined localization. Calculated position for ToF and combined ToF and DoA is $(2.51, -7.49)$, calculated position for DoA localization is $(2.61, -8.40)$. The position of the array is $(3.00, -8.05)$.

5.5 Simulation results

Using simulations we will show the expected performance of our combined ToF and DoA localization approach in a more controlled setup. To perform simulation we use the ToF and DoA error that we derived from the experimental setup.

In the simulations we use the same setup as the experimental setup. The 4 reference nodes are placed at the surface and the blind node is placed at the same position as in the experimental setup. Rather than performing measurements, we use the derived error distributions from the experiment to simulate measurements errors. We run the simulation for 100 different measurements, the results are shown in Figure 5.10.

	μ	σ
ToF $\mathcal{N}(0.0, 0.1219 \cdot 10^{-5})$	0.0522m	$1.8391 \cdot 10^{-4} m$
DoA $\mathcal{N}(0.0, 1.4775 \cdot 10^{-2})$	0.4174m	$0.3157 m$
Combined	0.0522m	$1.8391 \cdot 10^{-4} m$

Figure 5.10: Simulated localization accuracy, ToF, DoA and combined ToF and DoA are performed in a simulated deployment similar to the setup of the experiment. Shown are the average positional error and standard deviation of the error.

Similar results as from the experiment can be seen in the simulation, the result of the combined solutions pull toward the ToF result. This is the result of the large difference in measured error distributions of the measurements. The ToF is weighted more significant because of the small error deviation. Different from the experimental results, the ToF is more accurate than the DoA positioning. We think that in the

experimental setup the ToF is also actually more accurate than the DoA result, and the position of the array is actually positioned more towards the ToF calculated position. Similar to the experiments, the simulation achieves sub-meter accuracy for all localization approaches.

5.6 Conclusion

We consider one-way ranging important for localization under water because it improves the energy-efficiency and scalability. Both ToF and DoA localization can estimate a position using one-way ranging only. By combining ToF and DoA, the number of reference node required to perform localization can be reduced. Therefore in this work we propose a combined ToF and DoA localization approach.

The goal of our localization approach is to provide localization and sound-source tracking in short-range shallow water environments such as a harbor. These environments are highly reflective and may contain many multipath signals. To simulate such an environment the experiments were performed in a dive-center. ToF measurements were performed by measuring the propagation time of transmissions between moored reference transducers and a transducer array at the bottom of the tank. The DoA measurement was performed by determining the DoA angle of the signal impinging on the transducer array. The cross-correlation of the received signal shows that the environment is indeed a highly reflective environment.

However, looking at the results of the position calculation, we conclude that in our setup DoA outperforms ToF localization. However looking at the error distribution of the measurements, the angular measurements show a greater deviation than the ToF measurements, indicating the ToF localization to be more precise. What we can conclude is that both approaches and the combined localization approach achieve sub-meter accuracy.

We have used the error distributions derived from the experiment to perform simulations. From the simulations it becomes clear the ToF performes better than DoA positioning. The conflicting results of the experimental setup (DoA performing better than ToF) are likely the cause of a misplacement of the array. Similar to the experimental results, the position of the combined approach in the simulation pull completely towards the ToF calculated result. This is a result of the big difference in the deviation of the error distributions of DoA and ToF measurement. In simulation again, all approaches achieve sub-meter accuracy.

Bibliography

[1] R. Eustice, L. Whitcomb, H. Singh, and M. Grund, "Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4257–4264.

[2] P. Miller, J. Farrell, Y. Zhao, and V. Djapic, "Autonomous underwater vehicle navigation," *Oceanic Engineering, IEEE Journal of*, vol. 35, no. 3, pp. 663–678, 2010.

[3] J. J. Moré, "The levenberg-marquardt algorithm: Implementation and theory," in *Numerical Analysis*, ser. Lecture Notes in Mathematics, G. Watson, Ed. Springer Berlin Heidelberg, 1978, vol. 630, pp. 105–116.

[4] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks." in *SECON*. IEEE, 2006, pp. 374–382.

[5] K. Blom, R. Wester, A. Kokkeler, and G. Smit, "Low-cost multi-channel underwater acoustic signal processing testbed," in *The Seventh IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2012.

[6] W. Liu and S. Weiss, *Wideband Beamforming*, X. Shen and Y. Pan, Eds. Wiley, 2010.

CHAPTER 6

Cooperative Combined Localization and Time-Synchronization

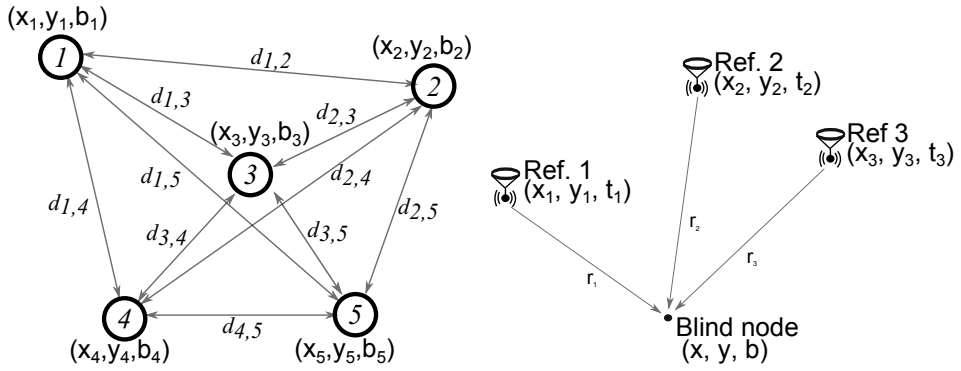
Traditionally, underwater localization and time-synchronization are performed separately. This, however, requires two-way ranging between nodes to determine propagation delays resulting in high power consumption and communication overhead. One-way ranging can be used by using a combined time-synchronization and localization approach. While such an approach exists for non-cooperative networks, to the best of our knowledge no such approach exist for cooperative networks. A cooperative approach has significant benefits in terms of number of reference nodes required, flexibility of placing reference nodes, and accuracy of localization and time-synchronization. Therefore, in this chapter we propose a cooperative combined localization and time-synchronization for UASNs called aLS-Coop-Loc.

Using real-world experiments we evaluate the performance of our cooperative combined localization and time-synchronization approach called aLS-Coop-Loc and a non-cooperative approach. We perform experiments using the SeaSTAR node and a COTS node from Kongsberg Maritime at a lake and at Strindfjorden in Norway. These experiments provide realistic insight into ranging performance in real-world environments.

Both simulations and real-world experiments show that aLS-Coop-Loc outperforms non-cooperative approaches in terms of accuracy of localization and time-synchronization: about 23% better position accuracy and 27% better time-synchronization in simulation and about 2% to 34% better position accuracy in results from real-world experiments. Moreover, results show cooperative localization is able to operate with fewer number of reference nodes, a scenario in which non-cooperative localization can not operate. In simulation this results in reduced accuracy compared to non-cooperative localization, while in results from the experiments indicate cooperative localization is able to outperform non-cooperative localization with reduced number of reference nodes.

6.1 Introduction

When measuring, it is important to know *where* and *when* the measurement was taken. Therefore, localization and time-synchronization play an important role in wireless sensor networks. Existing work on time-synchronization [6] and localization [3]



(a) Cooperative

(b) Non-cooperative

Figure 6.1: Example of cooperative and non-cooperative localization. Cooperative localization uses all pair-wise measurements available, while non-cooperative localization has only measurements between reference nodes and blind-nodes.

consider these aspects separately. However combined localization and time-synchronization, similar to what is already done by GPS [5], allows the position and time to be simultaneously estimated using one-way ranging only. The significant benefit of one-way ranging is its lower communication overhead compared to two-way ranging. With one-way ranging, broadcasts can be used, through which the number of packets required to be sent before localization is performed, is reduced from quadratic to linear complexity in number of network nodes (N), which we show in Section 6.3.2. Because bandwidth is very limited in UASNs and data rates are therefore very low, localization and time-synchronization using one-way ranging becomes important. Another advantage offered by one-way ranging is reduction of energy consumption due to lower communication overhead offered by one-way ranging.

One of the distinguishing factors between cooperative and non-cooperative localization is the separation between the unlocalized and unsynchronized blind-nodes and the synchronized reference nodes with known positions. In non-cooperative localization there is a clear separation between reference nodes and blind-nodes. In cooperative localization, on the other hand, this separation is less exact as the nodes may only have partial reference information, i.e. only a single dimension or only time.

Figure 6.1 shows an example of cooperative and non-cooperative localizations. In cooperative localization, all nodes cooperate to determine their position, using all pair-wise distance measurement. This, potentially, increases the accuracy of cooperative localization and allows more flexible selection of the reference nodes.

GPS is a well known example of non-cooperative localization. There is a clear distinction between reference nodes and blind-nodes, satellites and GPS receivers. While a cooperative approach, such as MDS [2] localization and our aLS-Coop-Loc approach do not have this clear separation.

To allow localization and time-synchronization using only one-way ranging, a combined localization and time-synchronization approach is required. Such a combined localization and time synchronization method already exists for non-cooperative localization approaches like GPS [5], however to the best of our knowledge such approach has not yet been proposed for cooperative networks.

The contributions of this work are twofold:

- We propose a cooperative combined localization and time-synchronization approach for underwater networks called Asynchronous Least-Squares Cooperative Localization (aLS-Coop-Loc). Using both simulation and real-world experiments we evaluate the performance of aLS-Coop-Loc in a small-scale underwater network in two different locations, het Rutbeek, i.e., a recreational lake in city Enschede in the Netherlands and in Strindjorden in Norway near the city of Trondheim. We use two hardware platforms for the experiments, i.e., (i) the SeaSTAR node, a node designed withing the SeaSTAR project [1], and (ii) the Mini underwater acoustic nodes from Kongsberg Maritime. The former, as being a commercial solution, does not allow us to fully control the ToA estimation approach, but does however allow us to perform tests over a larger area.

- Our real world experiments also provide insight into the performance of underwater acoustic ranging in a realistic environment. In underwater research, work has been mostly limited to simulation. Performing real world experiments with underwater communication is difficult and time-consuming. Therefore researchers in the field of underwater communication mostly tend to limit themselves to simulations or theoretical work only. Input for such simulations, derived from experimental work, is not readily available. Moreover, traditional acoustic communication and experiments focus on long-range communication, while we are interested in short range communication.

This chapter is organized as follows: First we review related work regarding localization and time-synchronization in Section 6.2. In Section 6.3 we introduce our cooperative combined localization and time-synchronization algorithm (aLS-Coop-Loc). We compare the performance of non-cooperative and cooperative localization in simulation in Section 6.4. Real-world experiments are described in Section 6.5, including the experiment setup and results.

6.2 Related work

In this section we review time-synchronization and combined localization and time-synchronization for non-cooperative and cooperative networks. Regarding cooperative localization, we will review the MDS localization approach. This approach is a commonly used approach for cooperative localization. It, however, does not incorporate time-synchronization.

6.2.1 Time-Synchronization

Time-synchronization is the process of synchronization of the clocks at different nodes such that an agreement is reached on what the current time is. This notion of 'time' does not necessarily have to be global (world time), as the nodes can agree on a local time for the complete network.

Let us consider a network of N nodes. Every node has a clock and all nodes are assumed to have the same frequency increment. The clock is modeled as a $\mathbb{R}$ variable, which increases continuously over time. We denote the clock of node i as ϕ_i. Every clock has a bias, this is an offset of the clock compared to another clock. We denote the bias of the clock as b_i. To synchronize a network of N nodes all biases of all the clocks should be calculated according to:

$$\phi_1 - b_1 = \phi_2 - b_2 = \cdots = \phi_N - b_N \tag{6.1}$$

Owing to the availability of low-drift clocks, we assume that the clock drift during range measurements is negligible and we therefore do not take clock drift into account in our model.

6.2.2 Non-cooperative Localization

A number of non-cooperative combined localization and time-synchronization already exist, an example of which is the GPS [5]. Figure 6.1(b) shows a non-cooperative localization and time-synchronization setup. All reference nodes have a known position (x_i, y_i) and are assumed to be synchronized, i.e. their clock biases are known. The reference nodes send out their position information (x_i, y_i) and the time when a message was sent (t_i). A blind node records the arrival time of the message (r_i) and is able to calculate the ToF of the message with a clock-bias of its local clock (b). The blind-node should estimate both its position (x, y) and clock-bias (b). This is done by minimizing the following cost function:

$$\min_{x,y,b} \sum_{i=1}^{N} (\sqrt{(x - x_i)^2 + (y - y_i)^2} - v \cdot (r_i - t_i - b))^2, \tag{6.2}$$

where v is the propagation speed of the signal, which is commonly approximated to $1500\,\mathrm{m/s}$ in water. Several methods exist to minimize the quadratic cost function (6.2). We use the Levenberg-Marquardt iterative optimization technique [4], also described in Appendix B, which is well-behaved and a proven scheme.

6.2.3 Cooperative Localization

The MDS approach is a commonly used approach to cooperative localization, however it requires prior time-synchronization or costly two-way ranging.

An MDS localization algorithm calculates relative positions and works by measuring the distance between all pair of nodes in the network. The distance measured between a pair of nodes is placed in a dissimilarity matrix δ.

$$\delta = \begin{pmatrix} \delta_{1,2} & \delta_{1,3} & \delta_{1,4} & \dots & \delta_{1,N} \\ & \delta_{2,3} & \delta_{2,4} & \dots & \delta_{2,N} \\ & & \delta_{3,4} & \dots & \delta_{3,N} \\ & & & \dots & \dots \\ & & & & \delta_{N-1,N} \end{pmatrix}$$

The dissimilarity matrix is used to find the position vectors of the nodes $x_1 \dots x_N \in \mathbb{R}^3$. This is done by minimizing the following cost function [2]:

$$\min_{x_1 \dots x_N} \sum_{i=1}^{N} \sum_{j=i+1}^{N} (\|x_i - x_j\| - \delta_{i,j})^2, \tag{6.3}$$

where $\|x_i - x_j\|$ is defined as the Euclidian distance between vector x_i and x_j. Note that the upper triangle of the dissimilarity matrix is used, MDS assumes the dissimilarity matrix is symmetric.

MDS is a relatively straight-forward way of determining relative positions, however does require two-way ranging or prior time-synchronization, which introduces significant communication overhead.

6.3 aLS-Coop-Loc

In this section we introduce a combined localization and time-synchronization approach for cooperative networks. Our algorithm follows a similar approach as MDS. However, rather than just calculating the position (x, y) of nodes, also the unknown clock bias (b) is calculated. Although we describe our localization for a 2-dimensional setup, it can be easily extended to three dimensions.

Before we describe the localization algorithm, first we review how range measurements are calculated. Let us consider nodes are positioned in a D dimensional space, where D = 2 or 3. Let $\vec{x}_{i=1\dots N}$, $x_i \in \mathbb{R}^D$ be the vector or coordinates of node i and assume $b_i \in \mathbb{R}$ is the clock bias of node i.

Let us consider two nodes with index i and j out of a network of N nodes. If we define the start of transmission of a message on node i as t_i and the reception time of the message as r_i and also consider the clock bias of both nodes b_i and b_j, we can measure the ToF between two nodes as follows:

$$(r_i - b_i) - (t_j - b_j) = tof_{i,j} \tag{6.4}$$

From the $tof_{i,j}$ we can calculate the pseudo-distance between the two nodes ($\tau_{i.j}$) by using the propagation speed $v \approx 1500\,\mathrm{m/s}$. The measured pseudo-distance between nodes (denoted by $\tau_{i,j}$), measured during the operation of an underwater network, and the estimated distance between the nodes, calculated during the process of iterative optimization, should converge to the same value:

$$(||\vec{x}_i - \vec{x}_j|| - v(b_i - b_j)) \rightarrow \tau_{i,j} \tag{6.5}$$

Where $\tau_{i,j} = v((r_i - b_i) - (t_j - b_j))$. We measure these pseudo distances ($\tau_{i,j}$) between nodes using acoustic communication during the operation of the network and place them in a dissimilarity matrix similar as being done in MDS:

$$\tau = \begin{pmatrix} \tau_{1,2} & \tau_{1,3} & \tau_{1,4} & \cdots & \tau_{1,N} \\ & \tau_{2,3} & \tau_{2,4} & \cdots & \tau_{2,N} \\ & & \tau_{3,4} & \cdots & \tau_{3,N} \\ & & & \cdots & \cdots \\ & & & & \tau_{N-1,N} \end{pmatrix}$$

However rather than measuring the actual distance between two nodes, we measure the distance between nodes with a clock bias error of the sender and a clock bias error of the receiver. After measuring the propagation delay between the nodes with the unknown clock biases, for every node we estimate the position ($\vec{x}_i \in \mathbb{R}^D$) and clock-bias ($b_i \in \mathbb{R}$) by minimizing the following cost function:

$$cost = \min_{x_1 \ldots x_N, b_1 \ldots b_N} \sum_{i=1}^{N} \sum_{j=i+1}^{N} ((||\vec{x}_i - \vec{x}_j|| - v(b_i - b_j)) - \tau_{i,j})^2 \tag{6.6}$$

Note that the upper triangle of the dissimilarity matrix is used, hence the approach uses one-way ranging.

We minimize this cost function using Levenberg Marquardt Algorithm (LMA) [4]. Similar to the MDS described in Section 6.2.3, this approach uses the upper triangle of the dissimilarity matrix.

6.3.1 Adding references and resolving ambiguity

The positions that are calculated using the cost function from Equation 6.6 fit the measured distance, but can be rotated or flipped in all dimensions. This means that they can be rotated in the space dimension ($\mathbb{R}^D$), which is similar to the ambiguity of the MDS approach, but it can also rotate or flip in the time dimension ($\mathbb{R}$). We call this problem the space-time ambiguity.

To add meaning to the results some reference measurements are required. For example to resolve the space-time ambiguity a measurement of the clock bias needs to be added. This is done using a single round-trip measurement between only two nodes in the whole network. This is a significant improvement compared with existing approaches, which require at least round-trip measurements between a reference nodes and all other nodes in the network.

Consider a set of nodes called ref_{time} for which we have the measured clock bias denoted as β_i. The cost function can now be extended with the following cost:

$$cost_{time} = \sum_{i \in ref_{time}} (\beta_i - b_i)^2 \tag{6.7}$$

The same can be done if we know or are able to measure the position or part of the position of a node. For example if the depth can be measured, the cost function can be extended with a cost for the depth. The set of nodes for which we have a reference

is called $ref_{space,3}$ and we denote the measurement of the depth as $\vec{r}_{i,3}$, we can now extend the cost function with:

$$cost_{space,3} = \sum_{i \in ref_{space,3}} (\vec{r}_{i,3} - \vec{x}_{i,3})^2 \tag{6.8}$$

This can be done for any dimension d as follows:

$$cost_{space,d} = \sum_{i \in ref_{space,d}} (\vec{r}_{i,d} - \vec{x}_{i,d})^2 \tag{6.9}$$

If we know the x, y or $depth$ of any node, we can extend the cost function with reference measurements. The set of reference nodes for dimension d is called $cost_{space,d}$. The complete cost function is now the basic cost function for fitting the distance measurements (Eq (6.6)) and all space (Eq (6.9)) and time (Eq (6.7)) cost functions:

$$cost_{overall} = cost + \sum_{d=1}^{D} cost_{space,d} + cost_{time} \tag{6.10}$$

To provide the absolute space-time coordinates some reference measurements are needed. To be able to at least split the results into space and time dimensions and to resolve the space-time ambiguity, the relative clock bias of two nodes in the network is required. To calculate an absolute position in 2 dimensions, 3 reference nodes in the network are required. Because the reference measurements can be split up into separate dimensions it is possible for example to measure the x, y position at surface nodes and assume the depth is 0, and to measure only the depth of a node that is submerged. This will allow the complete network to be absolutely positioned in 3 dimensions.

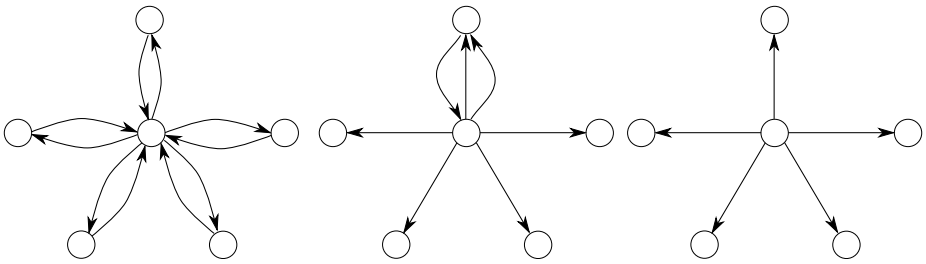
6.3.2 Communication overhead and power consumption

By looking at the amount of communication required to perform measurements we can say something about the time required to perform ranging measurements and the power consumption of nodes during the measurement phase. Figure 6.2 shows the complexity of different localization and time-synchronization approaches and the communication required by a single node. To perform localization and time-synchronization all the nodes in the network need to perform this communication.

Considering that an unsynchronized MDS approach requires two-way ranging between all pair of nodes (illustrated in Figure 6.2(a), the communication complexity of an MDS approach given the number of nodes in the network (N) is:

$$O\left(2N\frac{N-1}{2}\right) \tag{6.11}$$

A better approach (requiring less communication cycles) would be to first synchronize all the nodes in the network to a single master and then performing ranging using broadcast. Such an approach is illustrated in Figure 6.2(b), a node first performs



(a) MDS (b) MDS + Time Sync. (c) aLS-Coop-Loc

Figure 6.2: Example of communication required for a single node to perform localization and time-synchronization. With MDS two-way ranging between a node and all his neighbors is required, with MDS with prior time-synchronization two-way ranging between a node and a time master is performed and ranging can be performed using one-way ranging. With aLS-Coop-Loc only broadcasts and one-way ranging is used to perform time-synchronization and localization thereby reducing the amount of communication before localization and time-synchronization can be performed.

two-way ranging with a clock master and then broadcast a message to its neighbors (which are also time-synchronized) to perform ranging. The complexity of such an approach is:

$$O(2N + N) \tag{6.12}$$

The aLS-Coop-Loc approach (illustrated in Figure 6.2(c) requires broadcast messages only, and therefore the amount of communication required during the ranging phase of the localization and time-synchronization increases linear with the number of nodes in the network:

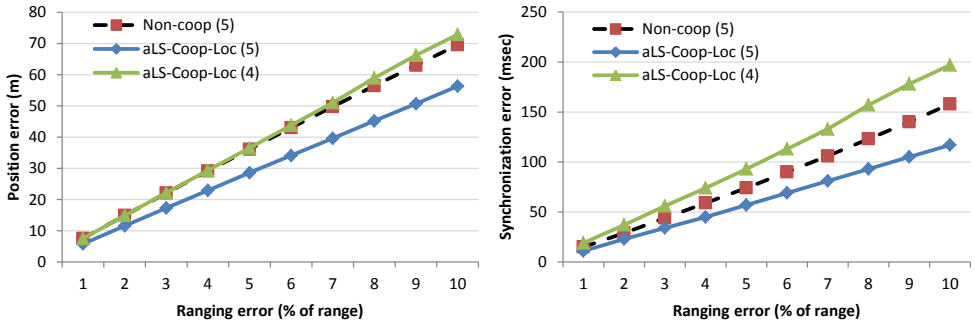
$$O(N) \tag{6.13}$$

The power consumption required during the measurement phase is a linear function of the amount of communication required and therefore follows the same complexity. This shows aLS-Coop-Loc outperforms existing approaches in number of communication cycles required. This reduces the energy consumed during the measurement phase, decreases the time required to perform measurements and improves reliability because fewer messages need to be received.

6.4 Simulation

6.4.1 Setup

We evaluate the performance of the aLS-Coop-Loc and non-cooperative localization approaches using simulation. The non-cooperative approach (non-coop) is the



(a) Position error

(b) Time-synchronization error

Figure 6.3: The simulated time-synchronization and position error as a function of the ranging error. Average error of different networks with 8 nodes deployed in an area of 500x500x500m. Ranging error is guassian distributed in range $\mathcal{N}(0, 0 \ldots 0.1 \times range)$. For the non-cooperative approach always 5 reference nodes are used, for the aLS-Coop-Loc approach the results are calculated with 4 and 5 reference nodes in the network.

approach described in the related work in Chapter 2.

In the simulation eight nodes are randomly deployed in an area of 500x500x500 meters (our targeted deployment size for the system). Five reference nodes are added to the network and the position accuracy is evaluated for different ranging errors between 0% and 10% of the range. The reference nodes are placed at the corners of the deployment area and the calculated positions are restricted to the deployment area. The ranging errors are distributed following a Gaussian distribution $\mathcal{N}(0, 0 \ldots 0.1 \times range)$. We run this simulation for at least 100 deployments and evaluate the average accuracy of time-synchronization and localization. For aLS-Coop-Loc we run the algorithm with 4 and 5 reference nodes in the network.

6.4.2 Results

The result of the localization accuracy and time-synchronization accuracy are shown in Figure 6.3(a) and Figure 6.3(b), respectively. It can be seen from the results that aLS-Coop-Loc outperforms the non-cooperative localization with the same number of reference nodes. This is due to the fact that the cooperative localization uses more measurements compared with non-cooperative localization. The cooperative localization approach uses all measurements available between all pair of nodes, while non-cooperative localization uses only measurements between reference nodes and blind nodes. On average the aLS-Coop-Loc(5) approach has an improved position accuracy of 23% and performs 27% better time-synchronization. When using only 4 reference nodes, the aLS-Coop-Loc algorithm is still able to calculate positions, although the accuracy of the position estimate is decreased. Position accuracy is decreased by 2% on average while time-synchronization is about 26% worse compared

to non-cooperative localization and time-synchronization.

6.5 Real world experiments

To better evaluate performance of cooperative and non-cooperative localization approaches in more realistic environments, we perform ranging measurements in three different underwater environments as described below. These experiments were performed using two underwater node platforms, i.e., (i) the SeaSTAR node and (ii) commercially available Mini node of Kongsberg Maritimes. Appendix A contains a more detailed description of these hardware platforms.

The SeaSTAR node is an underwater node developed within the SeaSTAR project [1] as an inexpensive open platform for underwater communication experiments. The node is based on an off-the-shelf ARM Cortex-M3 which was extended by a custom analog acoustic transceiver front-end and a HopeRF RFM22B 433 MHz RF wireless link module. A three-cell 2200 mAh LiPo battery was used to power each node. The radio module can be used to provide a connection to the shore when the node is deployed at the surface, allowing for an easy to use high bandwidth connection to gather experimental data. This greatly simplifies performing experiments because it allows accurate time-synchronization and data collection during the experiment using the radio, rather than an acoustic underwater link.

Using these hardware platforms we have performed three different test:

- **het Rutbeek short-range**: 6 SeaSTAR nodes were deployed in a recreational lake in city of Enschede. Distances between the nodes were small, ranging between 12 and 55 meters. Two different deployments were tested, one of which is shown in Google Earth as illustrated in Figure 6.4. The only difference between the first and the second setup are the shifted position of the deployed nodes. The nodes are deployed at the surface allowing the antenna to have a radio connection with the shore. The data was collected using this radio connection to the shore. At the shore a laptop was collecting all data from all nodes to log files. Using sandbags and ropes the nodes were fixed to the lake bottom.

- **Strindfjorden short-range**: In this short-range setting, nodes were deployed from a pier close to the shore of Strindfjorden in Trondheim Norway. Distances between the nodes were small, ranging between 7 and 25 meters. This test was performed using both the SeaSTAR node and the commercial Mini nodes of Kongsberg Maritimes.

- **Strindfjorden long-range**: In the long-range setup, nodes were deployed from a research vessel and were fully submerged at a depth of about 200 meter and with distances between nodes ranging from 200 meters up to 1.5 kilometers. This test was performed using the commercial Mini nodes of Kongsberg Maritimes.

In the above described setups we first measure the position of the nodes using a GPS receiver. We then determine the distance between the nodes using acoustic ranging. We compare the acoustic ranging results with the distances calculated based

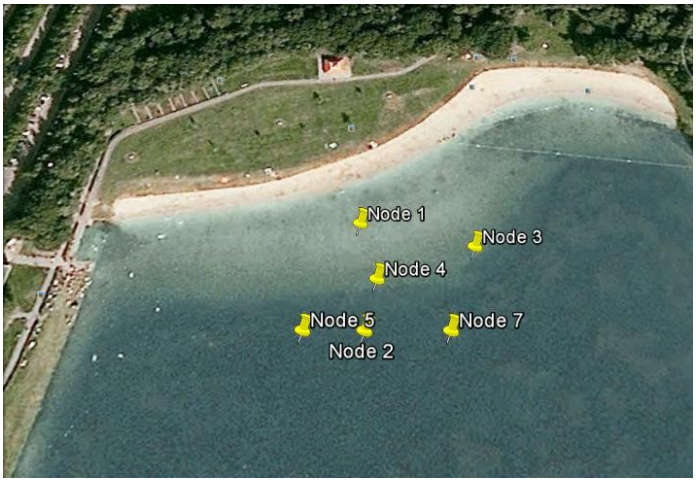


Figure 6.4: Example of network deployment on Google Earth, nodes were deployed up to a distance of 50 meters from each other

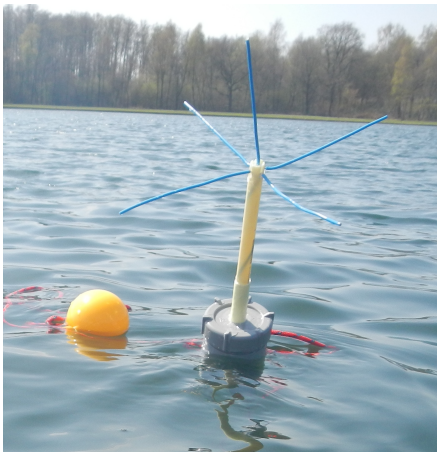


Figure 6.5: Node deployed at het Rutbeek, visible are the node antenna for radio connection to shore, the grey PVC container for the electronics and a bouy

on the GPS positions. The error of the GPS distance compared to the acoustic ranging is expressed as a percentage of the range. From these errors, we calculate the normal distribution of the links denoted by $\mathcal{N}(\mu, \sigma)$.

6.5.1 het Rutbeek short-range

Using the experimental setup we determine the ranging performance of the SeaSTAR node in short-range setups. The ranging results from the different experiments vary from reasonably accurate (error of about 10%) to very inaccurate (error up to

Method	Position error	Time error
Non-Cooperative	11.8 m (1.00)	4.7 msec (1.00)
aLS-Coop-Loc (5)	8.2 m (0.69)	2.2 msec (0.47)
aLS-Coop-Loc (4)	9.8 m (0.83)	2.5 msec (0.53)

(a) Rutbeek deployment 1 $\mathcal{N}(11.53, 16.48)$

Method	Position error	Time error
Non-Cooperative	26.3 m (1.00)	16.4 msec (1.00)
aLS-Coop-Loc (5)	24.2 m (0.92)	6.4 msec (0.39)
aLS-Coop-Loc (4)	25.4 m (0.97)	6.2 msec (0.38)

(b) Rutbeek deployment 2 $\mathcal{N}(27.85, 42.22)$

Figure 6.6: Results performance evaluation with 16 nodes, table shows the positioning error, time-synchronization error for different localization approaches run with different error parameters obtained from real measurements performed in Rutbeek. Results between brackets are the results relative to the non-cooperative approach.

100% of the range). Using the measurements we were able to derive ranging error distributions.

The average distributions for the Rutbeek setups are:

$$\mathcal{N}(11.53, 16.48), \mathcal{N}(27.85, 42.22)$$

The results of the ranging show a significant ranging error. What the impact of the GPS error is on the measurements is unclear. The GPS receiver we used was a Garmin Geko using WAAS, indicating a positioning error of 1-3 meter. The absolute positions of GPS may have an error, but the relative distances between the GPS measurements seem to be quite accurate when we compare the actual deployment with the GPS derived deployment on Google Earth. We noticed that nodes were somewhat drifting because of the wind and as such they were not static. Therefore an error will always be present in the system if the positions are compared to a single GPS measurement. Because there is a large uncertainty about the position of the nodes in the network, due to the GPS error, we use the derived ranging error distribution to evaluate the performance of the localization algorithms rather than calculating on the measurements directly.

Using the ranging error distributions derived from the experiments, we make an estimation of the performance of the localization and time-synchronization approaches in such an environment. For every error distribution we run a simulation, generating link range estimations using the error distributions derived from the experiments. This simulates an environment which is similar to the environment in which we performed the experiment, yet allows us to vary other settings such as number of reference nodes and total number of nodes in the network.

The evaluation was performed for a network of 16 nodes, with 5 reference nodes

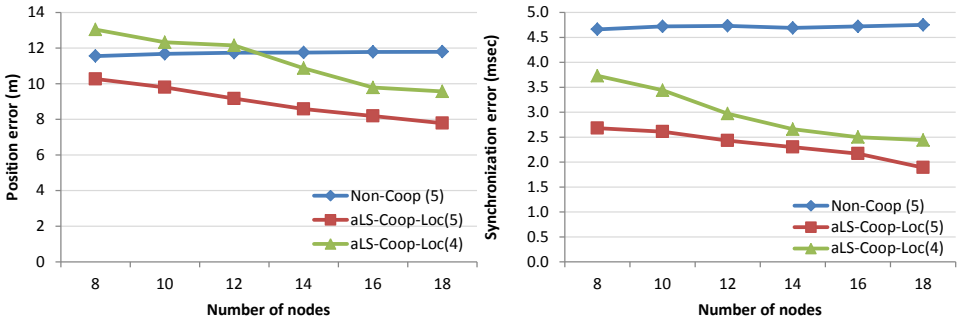


Figure 6.7: Performance results using the parameters obtained in Rutbeek deployment 1, run with increasing number of nodes in the network.

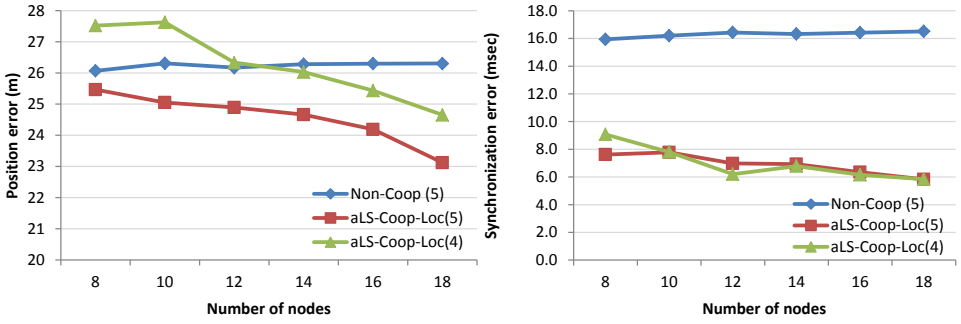


Figure 6.8: Performance results using the parameters obtained in Rutbeek deployment 2, run with increasing number of nodes in the network.

for the non-cooperative localization and 4 and 5 reference nodes for cooperative localization, with nodes randomly positioned on an area of 50x50x50m. The calculated positions are limited to the bounding box of the localization area and the position of the reference nodes are at the corners of the localization area. The results of this evaluation is shown in Figure 6.6.

From the results illustrated in Figure 6.6 it can be seen that cooperative localization using 4 reference nodes outperforms non-cooperative localization: position accuracy is 31% and 8% more accurate and time-synchronization is about 53% and 61% more accurate. In the case of having 4 reference nodes, cooperative localization is still able to calculate the position and perform time-synchronization, and does so with a slightly improved accuracy in terms of position and a much better time-synchronization accuracy compared to non-cooperative localization.

We have also evaluated the performance of the localization algorithm with different number of nodes in the network. Results are shown in Figure 6.7 and Figure 6.8. The aLS-Coop-Loc approach was evaluated for networks with 4 and 5 reference nodes. What can be seen from the results is that the non-cooperative performance is independent of the number of nodes in the network. The aLS-Coop-Loc results are dependent on the number of nodes in the network. The aLS-Coop-Loc(5) approach outperforms the non-cooperative localization on average approach by 2% to 27% for

Link	Distance	μ	σ
Node 2 - 3	6.94 m	-21.54	6.97
Node 2 - 4	7.46 m	-29.11	4.23
Node 2 - 5	18.03 m	13.46	1.07
Node 3 - 4	6.79 m	-34.88	1.91
Node 3 - 5	11.09 m	-10.99	3.02
Node 4 - 5	15.19 m	-7.61	2.43
Overall		-15.18	16.05

Figure 6.9: Results ranging short-range experiments performed using the Kongsberg Mini. Ranges were derived from the measured positions using GPS. Errors are shown as percentage of the range.

.

position accuracy and about 50% for time-synchronization.

From the results it also becomes clear that using fewer reference nodes (4 in this case) for aLS-Coop-Loc has in some cases worse performance than non-cooperative localization. The average position accuracy for aLS-Coop-Loc(4) ranges from 17% worse to 19% improved performance. From these results it also can be seen that the number of nodes has an effect on the performance for cooperative localization and time-synchronization.

6.5.2 Strindfjorden short-range setup

In the Strindfjorden short-range setup we have performed a ranging performance test using both the Kongsberg Mini and the SeaSTAR node in the same near-shore short-range setup. For the near-shore experiments, again, a GPS receiver was used.

For the SeaSTAR nodes the average ranging error distribution is:

$$\mathcal{N}(18.65, 20.92)$$

The error distribution for the Kongsberg Mini nodes is:

$$\mathcal{N}(-15.18, 16.05)$$

Both the SeaSTAR as well as the Kongsberg Mini nodes show significant range errors in this short-range setup, in the next section we evaluate a long-range setup and we will see that these large errors are not present in a long-range setup.

The results of the individual link ranging errors using the Kongsberg Mini are shown in Figure 6.9. Interesting is that the μ of the ranging error is for most links negative. This indicates that there may be a systematic error in the ranging system. When performing a range measurement, the nodes compensate for the processing delay of a packet. We suspect that an error in the estimate of the processing delay causes the ranges to be underestimated.

Using the derived error distributions we have calculated the position accuracy of both the SeaSTAR as well as the Kongsberg Mini node in this environment. Results are shown in Figure 6.10. Again, the cooperative localization approach outperforms a non-cooperative localization approach.

Method	Position error	Time error
Non-Cooperative	15.0 m (1.00)	7.3 msec (1.00)
aLS-Coop-Loc (5)	11.5 m (0.77)	3.3 msec (0.45)
aLS-Coop-Loc (4)	13.9 m (0.93)	3.4 msec (0.47)

(a) SeaSTAR $\mathcal{N}(18.65, 20.92)$

Method	Position error	Time error
Non-Cooperative	10.8 m (1.00)	3.9 msec (1.00)
aLS-Coop-Loc (5)	7.0 m (1.03)	1.7 msec (0.59)
aLS-Coop-Loc (4)	8.1 m (0.77)	2.1 msec (0.57)

(b) Kongsberg Mini $\mathcal{N}(-15.18, 16.05)$

Figure 6.10: Performance result with the parameters derived from the Strindfjorden short-range test setup.

Link	**Distance**	μ	σ
Node 1 - 3	0.189 km	-3.09	5.50
Node 1 - 4	1.090 km	-2.18	0.51
Node 1 - 5	1.551 km	-2.23	0.45
Node 1 - 6	1.270 km	-1.48	0.44
Node 3 - 5	1.473 km	-1.42	0.30
Node 3 - 6	1.208 km	-1.41	0
Node 4 - 6	0.238 km	-8.86	0.32
Node 5 - 6	0.302 km	0.55	0.13
Overall		-3.12	3.60

Figure 6.11: Results long-range ranging experiments performed using Kongsberg hardware. Ranges were derived from the measured positions using SSBL. Errors are shown as percentage of the range.

.

6.5.3 Strindfjorden long-range

Tests were performed over longer ranges in an off-shore test in 200 meter deep water. The positions of the nodes underwater in the long-range test were determined using the Kongberg HiPAP system, a Super Short Base Line (SSBL) dynamic positioning system.

Results for the experiments are shown in Figure 6.11. For every link where measurements were available the error between the pre-determined range and the acoustically measured range was determined. These errors were expressed as a percentage of the range. From these errors the normal distribution of the link error was determined, denoted by $\mathcal{N}(\mu, \sigma)$.

What can be seen from the results show in Figure 6.11 is that the ranging error of

Method	Position error	Time error
Non-Cooperative	28.05 m (1.00)	8 msec (1.00)
aLS-Coop-Loc (5)	17.04 m (0,61)	4 msec (0.50)
aLS-Coop-Loc (4)	19.91 m (0.71)	5 msec (0.63)

Figure 6.12: Performance result for the long-range experiments with the parameters derived using Kongsberg hardware, simulation was performed on an area of 500x500x500m.

the long-range experiments were quite low. The results of the near-shore experiment show much greater errors than the short-range experiment.

We have performed the performance evaluation again with the measured error distributions of this experiment:

$$\mathcal{N}(-3.12, 3.60)$$

The evaluation was performed for a network of 16 nodes, with 5 reference nodes for the non-cooperative localization and 4 and 5 reference nodes for aLS-Coop-Loc. Results are shown in Figure 6.12. This time, the deployment area of the nodes was changed to 500x500x500m. Again, results obtained from the long-range experiment using the Kongsberg Mini match the previous obtained results, i.e. cooperative localization and time-synchronization outperforms the non-cooperative approach for the same number of reference nodes.

6.5.4 Evaluation of ranging errors

The tests involving shallow water short-range ranging experiments show significant errors. To get a better estimation of where these errors come from, we identify the following possible source of these errors:

- **GPS and SSBL error**. The measurement of the positions using the GPS (and the SSBL) result in incorrect estimation of the actual ranges. The GPS receiver indicated a positioning error of 1-3 meter, this is however an absolute error. The relative positioning error appears to be much smaller. Placing the GPS calculated positions on Google Maps and using the pier as reference the relative distances seem to match the reality quite correct. In future work it would be better to use a more accurate GPS receiver and validate some of the ranges with a laser rangefinder.

- **Dynamics of the nodes**. During the Rutbeek experiment, the nodes were not fixed and were moving with the wind. This causes the nodes to deviate from the GPS measured positions. In the Strandfjorden experiment the nodes were deployed from a pier and the position of the nodes during the experiment was therefore much more stable. A fixed deployment of the nodes is therefore preferred, deploying the nodes close to the bottom may result in a more fixed placement of the node but the radio interface provided by the SeaSTAR node can then not be used.

- **Incorrect ToA processing delay**. In software the modem needs to compensate for the processing delay of a packet to determine the accurate ToA. If this processing delay is incorrect or the processing delay is not constant, the ToA is incorrectly estimated. In the estimations performed with the Kongsberg Mini, the ranges are almost always underestimated. We suspect incorrect large processing delay results in the underestimation of the actual range.
- **ToA estimation error**. The last error we can identify is the ToA measurement performed by the node. As we have seen before in Chapter 5, a highly reflective environment such as a shallow water environment contains many multipaths. As can be seen in Figure 5.6 of the dive-center test, the LoS path is not always the strongest path. If the modem decodes the strongest path, the measurement will contain an error. Channel sounding of the environment is therefore recommended.

6.6 Conclusion

Localization and time-synchronization are important aspects of UASNs, as they give meaning to sensor measurements by adding information on where and when measurements are taken. Localization and time-synchronization, however, are traditionally done separately.

When time-synchronization and localization are performed separated, two-way ranging is required. Two-way ranging poses a significant overhead in terms of communication and energy consumption. With very limited data-rates and battery capacity available in UASNs this is not practical. One-way ranging is preferred because the amount of communication required to perform rangaing is significantly reduced.

Combined localization and time-synchronization existed before only for non-cooperative networks, for cooperative networks an approach also exist now. In this chapter we have shown a combined localization and time-synchronization approach for cooperative networks which we call aLS-Coop-Loc. We have formulated a cost function which can fit the position and clock bias of the nodes with the measured propagation delays in a cooperative network. The cost function can be used to calculate the positions and clock biases, but still reference nodes are required to remove space-time ambiguity.

We have shown how to include additional information such as depth, position and clock synchronization by extending the cost function. The algorithm is flexible in the reference positions and reference measurement can be used even if it is only one dimension. For example the depth information of a single node or the clock bias of two nodes.

Using simulation we have shown the aLS-Coop-Loc cooperative combined localization and time-synchronization approach outperforms a non-cooperative approach when the same number of reference nodes are used. When four reference nodes are used, cooperative localization and time-synchronization is still able to calculate position and time, and does so at a slightly better position accuracy and a much better time-synchronization.

We have performed real-world tests with the SeaSTAR sensor nodes and with COTS underwater nodes from Kongsberg. These tests provide insight into the performance of underwater acoustic ranging in realistic environments. Such results are not readily available, experiments in underwater communication are generally considered very hard and many researchers in the field limit themselves to simulations and theoretical work. Tests were performed at a recreational lake near the campus of the university and in at Strindfjorden near Trondheim. The measurements from the test were used to derive an error model for the acoustic ranging. Using this error model we evaluated again the different localization approaches. From this evaluation we can see that the experimental results match the results from the simulation: aLS-Coop-Loc outperforms non-cooperative localization with improved position accuracy of about 2% to 34% when the same number of reference nodes are used.

Using the error models derived from the experiments, we have performed performance evaluation of the different approaches with increasing number of nodes in the network. The performance of the non-cooperative approach is unaffected by the number of blind nodes in the network. The performance of cooperative localization improves with increasing number of nodes in the network in such environments.

We have also performed tests with COTS nodes from Kongsberg Maritime. Using this hardware we have performed a nearshore short-range measurement and an offshore long-range measurement. For the long-range measurement, the relative ranging errors are considerably smaller than for short-range measurements. We have identifier several causes of these errors.

Our approach on combined localization and time-synchronization provides a solution for cooperative networks which was not available before. Moreover, our experiments in realistic environments provide insight into the performance of ranging in real-world environments.

Bibliography

[1] Seastar project. *www.seastar-project.nl, 2010.*

[2] I. Borg and P. Groenen. Modern Multidimensional Scaling: Theory and Applications. *Springer, 2005.*

[3] K. Y. Foo and P. R. Atkins. A relative-localization algorithm using incomplete pairwise distance measurements for underwater applications. EURASIP J. Adv. Signal Process, *2010:11:1–11:7, January 2010.*

[4] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics, *11(2):431–441, 1963.*

[5] B. W. Parkinson, A. I. for Aeronautics, Astronautics, GPS, and NAVSTAR. Global positioning systems : theory and applications. Vol. 2. *American Institute of Aeronautics and Astronautics, 1996.*

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

Part III

MAC Protocols

CHAPTER 7

BigMAC: large-scale localization and time-synchronization

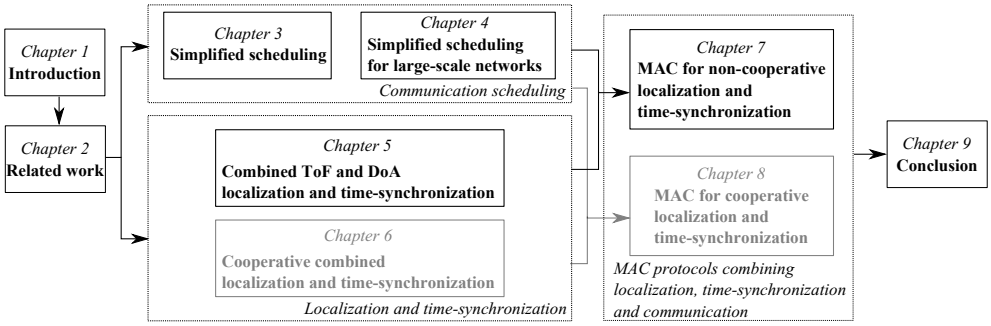


Figure 7.1: The MAC protocol uses the scheduling introduced in Chapter 4 and uses the localization algorithm with combined ToF and DoA from Chapter 5.

In this chapter we introduce a MAC protocol designed for large-scale underwater localization and time-synchronization networks. The MAC protocol assumes a network of static reference nodes and allows blind nodes to be localized by listening-only to the beacon messages. Such a system is known to be very scalable. We use the non-cooperative localization approach as described in Chapter 5 to provide a position estimate for the blind-nodes in this system. To allow efficient communication the transmission of beacon messages we extend and use the scheduling approach of Chapter 4 with broadcast scheduling. Figure 7.1 shows the relation of the MAC protocol with techniques introduced in previous chapters. We combine the localization approach from Chapter 5 with efficient communication scheduling introduced in Chapter 3 and Chapter 4. This system uses the efficient communication scheduling approach to provide a localization service.

We evaluate scheduled and unscheduled communication for sending beacon messages using simulation. Our experimental results show that when energy-efficiency of the beacons is of concern or when the modulation rate is low ($\leq 1kbps$), scheduled communication is preferred. For systems which are not concerned with the energy-efficiency of the beacons, unscheduled communication delivers more messages to the blind-nodes and localization and time-synchronization are performed faster in case

of high modulation rates ($5kbps$ and $10kbps$).

7.1 Introduction

In this chapter we look at the design of a MAC protocol for a scalable and energy-efficient underwater localization system. Our design uses a static network of beacons, which periodically send out beacon messages. Blind nodes can listen to the beacon messages and use the information to calculate their position and do time synchronization. In such a system it makes sense to look at scheduled communication for sending beacon message because requirements of scheduled communication and localization beacons overlap, beacons should be static, have a-priori knowledge of their position and should be time-synchronized. A localization approach using ToF, DoA or combined ToF and DoA as described in Chapter 5 can be used in such a system to provide a position estimate.

For the design of the MAC we will evaluate the performance of scheduled and unscheduled communication mechanisms to send the localization beacon messages. We consider ALOHA and scheduling communication for sending localization beacon messages and evaluate the performance of both approaches in terms of time required for localization, localization accuracy, and energy-efficiency.

The rest of the chapter is organized as follows: Section 7.2 describes our localization system design. In Section 7.2.1 we show how broadcast messages can be scheduled by extending the scheduling approaches we have introduced in Chapter 3 and Chapter 4. In Section 7.3 we will evaluate scheduled and unscheduled communication for sending localization beacon message. This allows us to determine which communication method is best for what circumstances. In Section 7.4 we look at how modulation rate and distance between the beacons effect the performance of ALOHA and scheduled based transmission of beacons.

7.2 Localization system design

For our approach we assume a static network of reference beacons for which the position of all beacons is known. The deployment of the beacon nodes is irrelevant for our approach, but as an example we assume a grid deployment. An example deployment is shown in Figure 7.2, representing a grid of reference nodes at the sea surface and a reference node at the sea floor. The reference nodes send out periodic beacon messages which can be used by blind nodes to determine their position. These beacon messages are the messages we aim to schedule and send without collisions.

The beacon messages contain the following information:

- Position of the beacon (x, y, z coordinates)
- Time-stamp of when the beacon was sent

Using this information the blind node can perform time-of-flight localization and time-synchronization by only listening to the beacon messages. The reference nodes can be equiped with GPS receivers for position and time information. Another

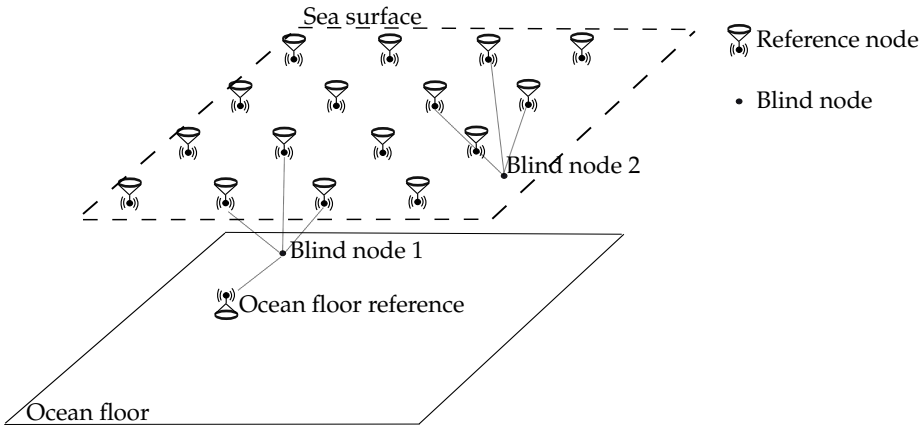


Figure 7.2: Example of a localization network with surface and ocean floor references and blind nodes listening to beacon message to localize and synchronize.

possibility is to preconfigure the position of the reference node and listen to other reference nodes beacon messages to remain synchronized.

7.2.1 Broadcast message scheduling

For scheduling the beacon messages we will use the simplified scheduling approach introduced in Chapter 3. To schedule the beacon messages we will first need to extend the set of simplified scheduling constraints with a constraint for beacon messages. A beacon message should be scheduled in such a way that on all positions within the network the message can be correctly received. In other words, no collision should occur at any position in the network.

This can be done as follows: when node A broadcasts its beacon message node B will have to wait until the message of node A passes, then node B can start transmission. When node B transmits immediately after the message from node A has passed node B the propagation circle of the message from B will always stay within the propagation circle of node A. This means that on any position within the network both messages can be received without any interference. An example of this is shown in Figure 7.3. This is similar to how OCSMA [1] orders its transmissions, however rather than creating a chain of transmissions as OCSMA does, we schedule the transmissions of the broadcast messages. In [2] another opportunity is exploited that arises when scheduling broadcasts from nodes which are outside of the interference range.

To put this in a scheduling constraint, node B will have to delay its transmission until the message from A has propagated to the position of B. Assuming the position of A (denoted as $\delta_i.src$) is the source of transmission δ_i, and position of B (denoted as $\delta_j.src$) is the source of transmission δ_j and assuming we can calculate the propagation time between two positions using the unspecified function T (For example T calculates the Euclidean distance between the two positions divided by a estimate of the sound speed underwater), the minimum delay between transmission δ_i and δ_j can now be calculated as:

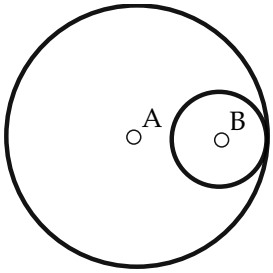


Figure 7.3: An example of how two broadcasts can be transmitted without collisions

$$\delta_i.duration + T(\delta_i.src, \delta_j.src)$$

Where $\delta_i.duration$ is the duration of transmission δ_i. This also shows that no exclusive access is needed, not even for broadcasting messages. The constraints we will add to the set of simplified scheduling constraints places a restriction between the transmission task start time $\delta_j.start$ and the transmission task start time $\delta_i.start$. Transmission task δ_j will wait for the message from δ_i to propagate from δ_i to δ_j (e.g. $T(\delta_i.src, \delta_j.src)$) and wait for it to completely pass the node ($\delta_i.duration$). This can be expressed using Equation (7.1).

$$\delta_j.start \geq \delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_j.src) \tag{7.1}$$

The complete set of simplified scheduling constraints is given in Figure 7.4 and now consists of the following four scheduling rules:

1. If two transmissions are scheduled from the same node, the first transmission should finish before the second one can start.
2. If any of the two transmissions is a broadcast, the broadcast scheduling constraint should be used.
3. If both transmissions are unicasts, the interference rule should be used.
4. If both transmissions are unrelated, both can be scheduled at the same time.

We can now use the set of scheduling constraints to determine the delays between transmissions and use the scheduling approach of Chapter 3 to find an efficient schedule.

7.2.2 Position estimation and time-synchronization

Position estimation in the proposed localization system design can be performed using ToF or a combined ToF and DoA approach. In Chapter 5 we have described ToF and combined ToF and DoA localization and time-synchronization. Such an approach can be applied to the proposed system. When using only ToF measurements, more reference nodes are required to be in range of the blind node to perform a position

$$\begin{cases} \delta_j.start \geq \delta_i.start + \delta_i.duration & \text{if } \delta_i.src = \delta_j.src \\ \delta_j.start \geq \delta_i.start + \delta_i.duration + T(\delta_i.src, \delta_j.src) & \text{if } (\delta_i.src = broadcast \text{ or } \delta_j.src = broadcast) \\ & \quad \text{and } Interfer(\delta_i.src, \delta_j.dst) \\ \delta_j.start \geq \delta_i.start + \delta_i.duration + max(& \\ \quad T(\delta_i.src, \delta_i.dst) - T(\delta_j.src, \delta_i.dst), & \text{if } \delta_i.src \neq \delta_j.src \text{ and } Interfer(\delta_i.src, \delta_j.dst) \\ \quad T(\delta_i.src, \delta_j.dst) - T(\delta_j.src, \delta_j.dst)) & \\ \delta_j.start \geq \delta_i.start & \text{otherwise} \end{cases} \tag{7.2}$$

Figure 7.4: Extended set of simplified scheduling constraints allowing broadcast scheduling.

estimate. When combining ToF and DoA, the number of reference nodes that need to be in range of the blind node is decreased. We assume the blind node has both ToF and DoA information and a position estimate can be performed with 3 reference nodes in range.

7.3 Performance evaluation

For the performance evaluation we compare a scheduled based approach with an ALOHA approach. To do this we deploy 16 beacons in a grid. The maximum communication range for the nodes is set to 500m and all beacons are spaced at a distance of 500m from each other. We uniformly deploy 100 blind nodes in such a way that all blind nodes are within communication range of at least 3 beacons. We are interested in the following performance parameters:

Time required for localization: which is the time it takes until all blind nodes are localized. For a node to find its location it needs to receive a beacon message from 3 distinct beacons.

Accuracy: We run the simulation for 30 seconds and measure the number of beacon messages that are received by the blind nodes. When more messages are received a localization algorithm using the MAC protocol may get a more accurate distance and position estimate because of having more information.

Energy-efficiency: In underwater communication most energy is consumed by sending messages. The sending of data consumes several factors more energy than receiving, sometimes as large as 100 times more. We will therefore focus only on the amount of data that is being sent. As a measure of the energy-efficiency we will look at the ratio between number of beacon messages sent versus number of beacon messages received ($\frac{N_{received}}{N_{send}}$). The higher this fraction the more effective the MAC protocol is in terms of effectively delivering the beacon messages to the blind nodes.

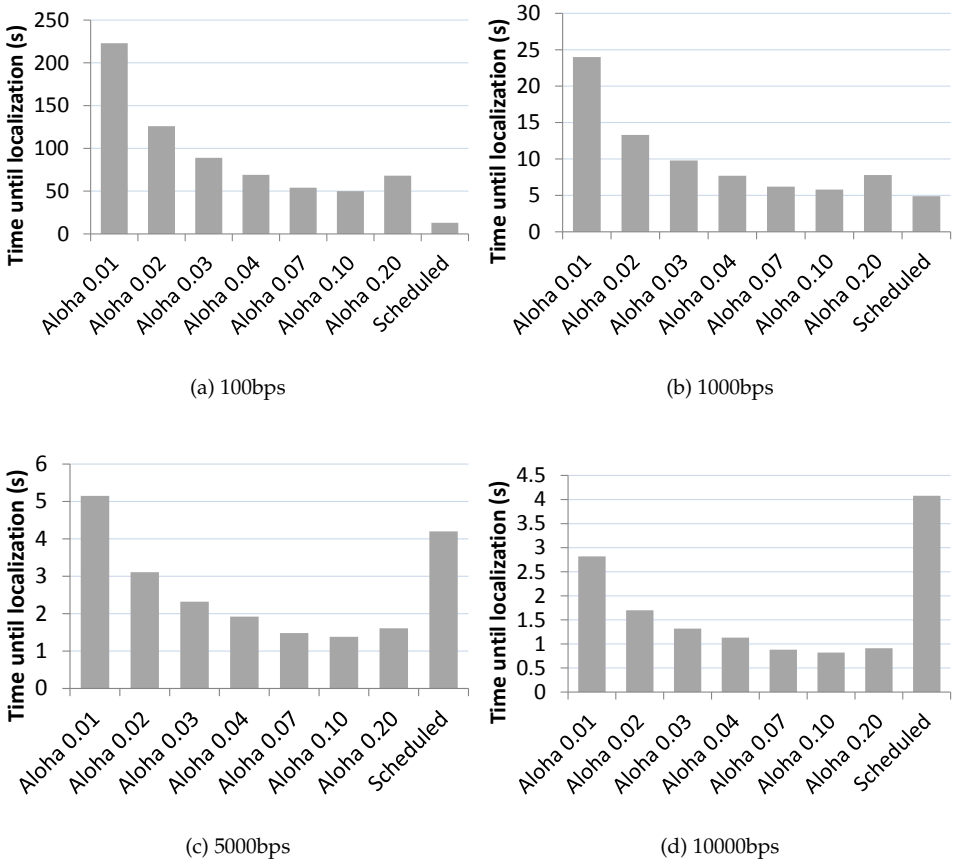
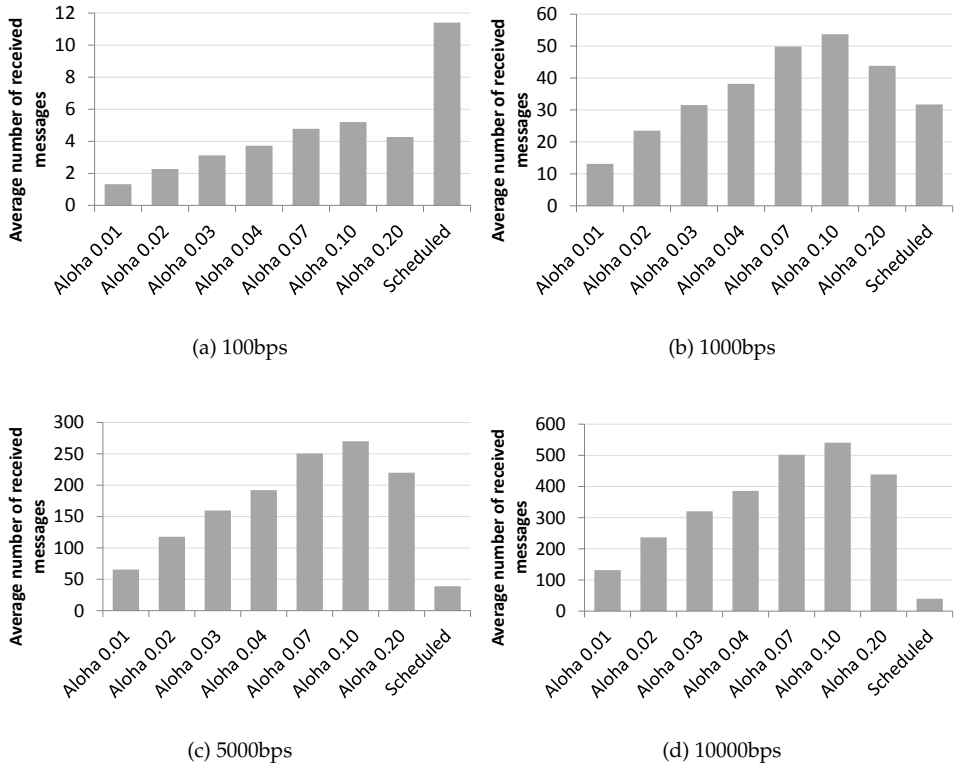


Figure 7.5: Average time required for localization at different bitrates for 100 nodes. This the time it takes for all nodes in the network to receive beacon messages from 3 distinct reference nodes. When measurements are received from 3 distinct reference nodes, the node is able to calculate its position. The simulation was run for different sending probabilities for the ALOHA approach and using a scheduled approach.

7.3.1 Time required for localization

We run the MAC protocols for 500 different deployments and four different modulation rates ($100bps$, $1kbps$, $5kbps$ and $10kbps$). In Figure 7.5 the average time required for localization is plotted for the different modulation rates. What can be seen from these results is that the scheduling approach is largely independent of the sending rate. For the lowest modulation rate ($100bps$) the time until localization for the scheduled approach is 13 seconds, but for the higher modulation rates this time is around 4 to 5 seconds. We see that the scheduling approach outperforms the ALOHA approach in the case of using the lowest modulation rate (100bps). Not shown in the graphs but observed during simulation is that the scheduling approach has very predictable



(a) 100bps

(b) 1000bps

(c) 5000bps

(d) 10000bps

Figure 7.6: Accuracy expressed in average number of received beacons by every blind-node. More beacon messages received means more measurements are available for performing localization and therefore localization accuracy will increase.

localization time while ALOHA shows large variations between simulation runs.

7.3.2 Accuracy

In Figure 7.6 the average number of received beacon messages by every node over an interval of 30 seconds is plotted. When more messages are received the blind node is able to average the ToF estimation over more measurements. This should increase the accuracy of localization. We can see that in the case of lower modulation rates the scheduling approach significantly outperforms the ALOHA based approach. Only for the higher modulation rate the ALOHA approach delivers the most messages to the blind nodes.

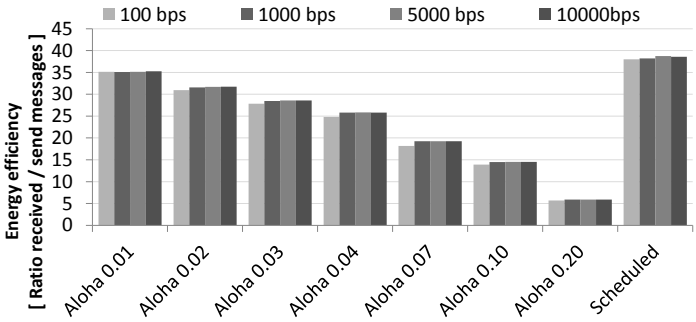


Figure 7.7: Energy-efficiency as defined by ratio of received messages by the blind-nodes for every beacon message transmitted by a reference node. When more messages are successfully received by the blind-nodes for every transmission, the energy-efficiency of the transmission is considered better.

7.3.3 Energy-efficiency

In Figure 7.7 the ratio between number of sent and received beacon messages is plotted. It can be seen that in this case the scheduled based approach works best. The scheduled approach has the best goodput, the highest number of successfully delivered messages for every transmitted beacon. This is the result of the fact that the transmission of the beacon messages are scheduled to be received free of collisions. From the energy-efficiency point of view a scheduled approach is therefore preferred for all modulation rates.

7.3.4 Evaluation

From the results of the simulation we can conclude that if energy-efficiency is not of concern only for the higher modulation rates ($5kbps$ and $10kbps$) the ALOHA approach is preferred. The schedule based approach is too conservative in sending beacon messages in this scenario, which results in fewer beacon messages being received by the blind nodes. For the lower modulation rates ($< 1kbps$) the scheduled approach outperforms the ALOHA approach in terms of time required for localization and performs similar in terms of number of received beacons. If energy-efficiency is of concern, then the scheduled based approach is preferred in all cases. To allow an ALOHA based approach achieve the same level of energy-efficiency a very low send probability should be selected which will result in long time required for localization.

7.4 Efficiency of broadcast scheduling

The efficiency of broadcasting for scheduled communication is affected by different parameters than for ALOHA communication. To evaluate what parameters influence the efficiency of scheduled and ALOHA broadcasting we measure a small setup with increasing distance between the beacons and with increasing modulation rate. In this setup we placed four beacons on a surface and transmit broadcast messages. On the

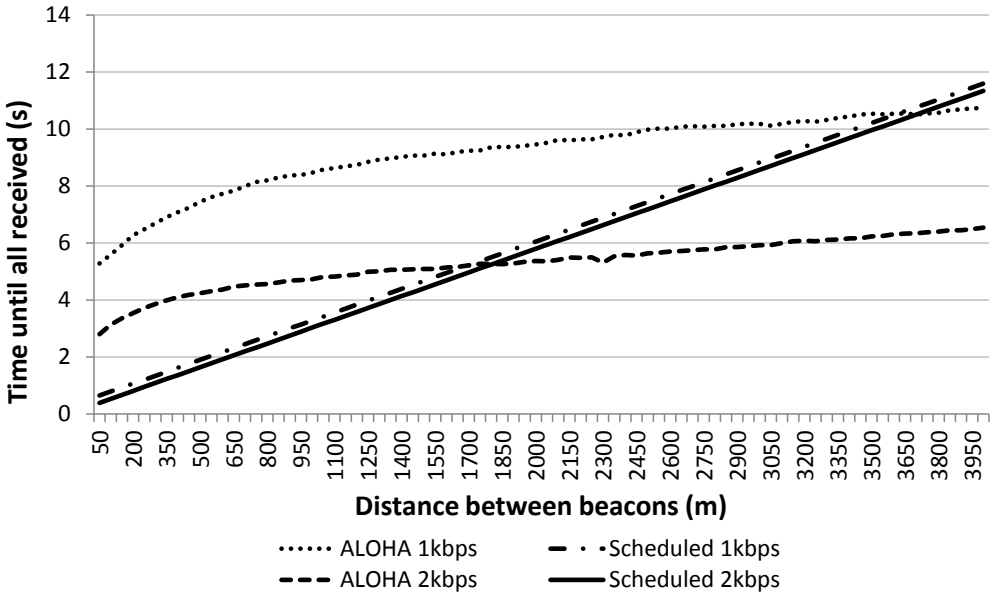


Figure 7.8: Result of broadcast scheduling at different beacon distances and different modulation rates. Performance of the scheduled broadcasting is largely dictated by the distance between the beacons and performance of the ALOHA approach is largely dictated by the physical layer throughput.

surface there are ten nodes which should receive the broadcast message. We measure the time it takes until all ten nodes have successfully received all four broadcast messages.

We compare the performance of scheduled communication with ALOHA [3]. We set the sending rate (G) of all the beacons combined to $G = \frac{1}{2}$ which is the optimal sending rate for pure ALOHA [3]. We run the simulation at different distances between beacons and at two different modulation rates. The results are shown in Figure 7.8.

What can be seen from the results is that scheduling broadcasts is not optimal. This is because when scheduling broadcasts the messages are scheduled very pessimistically, while in reality collisions occur much less often. The graph shows therefore a cut off point where ALOHA communication performs better than scheduled communication.

From the results it also becomes clear that the performance of scheduled communication is very dependent on the distance between beacons. This is because the delays for transmitting a packet are calculated based on the distance between beacons. The modulation rate, however, does not have a significant impact on the performance. The graph shows two lines for scheduled communication where the higher modulation rate only shows a slight improvement in performance (the 2kbps simulation is the lower line while the 1kbps simulation is the top line).

The ALOHA performance is not very dependent on the distance between the

beacons, i.e. the distance is a factor in the performance but the modulation rate is the biggest factor. When the modulation rate doubles the performance of ALOHA almost doubles. The biggest factor in the performance of ALOHA is the modulation rate.

7.5 Conclusion

We have designed a MAC protocol for an underwater localization system which uses static reference nodes and allows blind nodes to be localized by listening-only to the beacon messages. The system is scalable as the blind nodes do not send out any messages.

In such a system ToF, DoA or a combined ToF and DoA approach can be used to calculate the position and time-synchronization of the blind-node. In Chapter 5 we have shown how combined ToF and DoA can be performed, such a localization approach can be applied to this system.

This system shows how our proposed communication scheduling algorithm and proposed combined ToF and DoA localization and time-synchronization approach can be combined in a MAC protocol for large-scale localization systems.

For the design of the MAC protocol we have looked at both scheduled as well as unscheduled communication. We have compared the performance of a scheduled MAC to ALOHA for sending beacon messages. We evaluated the performance of the MAC in terms of time required for localization, accuracy, and energy-efficiency. Our results show that the when energy-efficiency is not of concern ALOHA MAC is the best choice only for high modulation rates ($5kbps$ and $10kbps$). It has the best results in terms of time required for localization and number of received beacons. The scheduled approach is the best choice for low data rates ($100bps$ and $1kbps$) and energy constrained systems. The scheduling of beacons guarantees collision free reception, maximizing the successful reception rate of the beacon messages. Also it has the most predictable performance and shows little variations in performance between different simulation runs.

Bibliography

[1] Y.-J. Chen and H.-L. Wang, "Ordered csma: a collision-free mac protocol for underwater acoustic networks," in *OCEANS 2007*, Sept 2007, pp. 1–6.

[2] H. Ramezani and G. Leus, "L-mac: Localization packet scheduling for an underwater acoustic sensor network," in *Communications (ICC), 2013 IEEE International Conference on*, June 2013, pp. 1459–1463.

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

CHAPTER 8

LittleMAC: small-scale cooperative underwater clusters

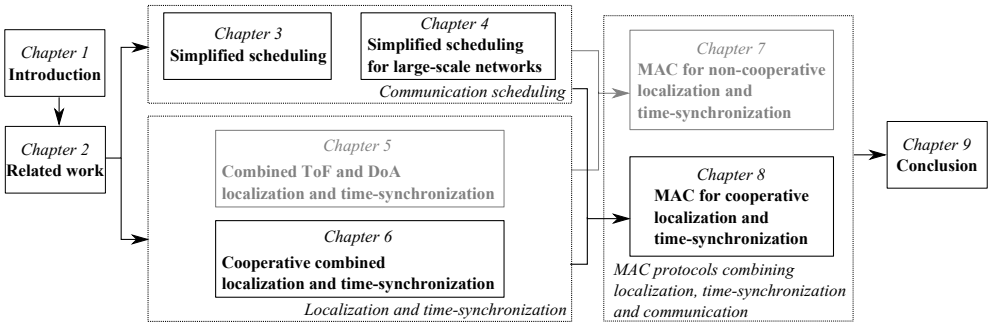


Figure 8.1: The proposed MAC protocol combines scheduling techniques from Chapter 3 and Chapter 4 with the cooperative localization approach proposed in Chapter 6.

In this chapter we propose a design for an underwater MAC protocol which combines localization, time-synchronization and communication. The design combines different techniques from previous chapters, Figure 8.1 shows the relation of this chapter with the algorithms described in previous chapters. In this chapter we evaluate how localization and time-synchronization can be performed during the deployment phase of the network, to allow efficient scheduled communication to be used during the operation phase of the network. Different than the mac proposed in Chapter 7, this MAC protocol is designed for small-scale networks but does not require the usage of reference to provide (relative) positioning and time-synchronization. Moreover the localization and time-synchronization is performed to allow efficient communication during the operation phase of the network, while the MAC of Chapter 6 is designed for a continous available service of localization and time-synchronization.

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-synchronization and communication important because scheduled communication, localization and time-synchronization require estimation of propagation delays between nodes to operate accurately and efficiently. For localization we use cooperative localization because it does not require the use of reference nodes

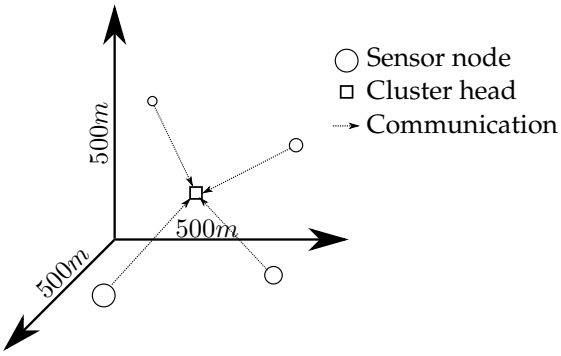


Figure 8.2: Example of a 3d deployment of an underwater cluster

to calculate relative positions. Because no reference nodes are required, deployment of such a network is simpler and cheaper because no external positioning system is required to determine the position of the reference nodes. However, because range measurements are required between all pair of nodes, scalability of the approach is limited.

Our MAC design consists of two phases, i.e. an unscheduled coordination phase and a communication phase. During the first phase, packet arrival times are measured and propagation delays are estimated. Relative positions and time-synchronization are performed using the approach described in Chapter 6. During the communication phase, sensor-data is transmitted using scheduled communication. Scheduling is performed using the simplified scheduling approach described in Chapter 3 and Chapter 4.

Using simulation we evaluate the feasibility of such a design, by measuring the time required for the coordination phase at different modulation rates.

8.1 Introduction

The focus of this chapter 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. An example of such a cluster is shown in Figure 8.2. Our aim is to provide scheduled communication, position estimation of nodes and time-synchronization simultaneously through a design of an integrated MAC.

Regarding localization, we provide relative position estimation using the approach described in Chapter 6. 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 UASNs. Determining the positions of reference nodes under water can be difficult and requires the use of external positioning systems such as an USBL system attached to a boat or use of surface beacons equipped with GPS. When only relative positions are required, such as for communication scheduling,

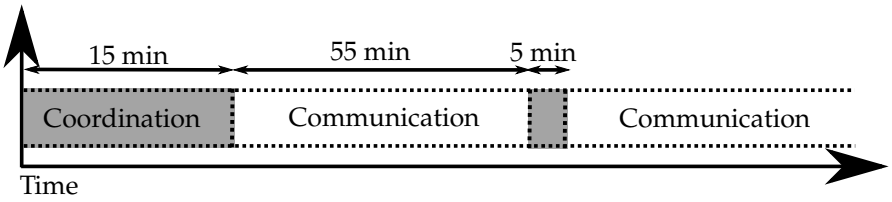


Figure 8.3: Timeline of a network in operation. Initialy the network is in the coordination phase. In this phase the positions of the nodes are determined. The cluster-head switches the network into the communication phase. After the communication phase, the network is switched back to a coordination phase to update the positions of the nodes.

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

The proposed MAC protocol combines localization, time-synchronization and communication in a way not done before. Traditionally underwater MAC protocols have focused on a single aspect only. In this chapter we will show a design for a MAC protocol which combines all three aspects. 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-synchronization to be given, we consider the overhead of an unscheduled coordination phase required for determining these positions and time-synchronization. We have evaluated this overhead in simulation and will show the required data-rates for an acceptable overhead.

The rest of this chapter is organised as follows: Section 8.2 describes the proposed design of the MAC protocol followed by Section 8.3 in which we evaluate the performance of the MAC protocol to finish the chapter with the conclusion in Section 8.4.

8.2 Design

The operation of our designed protocol as shown in Figure 8.3 is divided in a coordination and communication phase. Initially the network is in the coordination phase, in which nodes record the arrival time of the messages broadcast by other nodes in the network. The nodes in the network measure the propagation delay with a clock uncertainty in such a way that the measurements can be used by the combined localization and time-synchronization approach from Chapter 6. Once all the pair-wise propagation delays are measured, the cooperative localization and time-synchronization algorithm is able to calculate the relative positions of the nodes and the clock biases.

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

Field	Size	Description
$length$	1 byte	Length of packet
$type$	4 bits	Type of packet (coordinate)
$source$	4 bits	Unique identifier of source node
$time$	2 bytes	Time in $milliseconds$
$processing.delay$	2 bytes	Processing delay in $milliseconds$
$measured.arrival[16]$	2 bytes	Measured packet arrival time from neighbor
Total	38 bytes	

Figure 8.4: Information contained in measurement broadcast message. Broadcast messages are sent by all nodes during the coordination phase to measure the propagation delay between nodes. Using this information, the position and clock-biases of the nodes can be calculated.

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 fall-back 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-synchronization.

8.2.1 Coordination phase

The coordination phase of the MAC protocol is all about determining the propagation delays between nodes so that the measurements can be used to run the combined cooperative localization and time-synchronization method. The coordination phase of our MAC protocol is done using broadcast messages. Initially 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 8.4. The broadcast message contains the source address of the node and contains the measured arrival time of the messages broadcasted by neighbor nodes. All the nodes measure the arrival times of the neighbor broadcast messages during the coordination phase. They also broadcast these measured arrival times, so that the cluster-head can collect all the arrival times from all the nodes in the network. Using the cooperative localization and time-synchronization algorithm introduced in Chapter 6, the cluster head is able to calculate the time-synchronization and position of all the nodes in the network, using the measured arrival times.

Figure 8.5 shows how the arrival time measurement of the broadcast messages is done. When node 1 transmit a broadcast message, this message will be received by node 2. Node 2 will measure the arrival time of the broadcast message and measures the propagation delay with clock errors. This information is stored in the arrival field

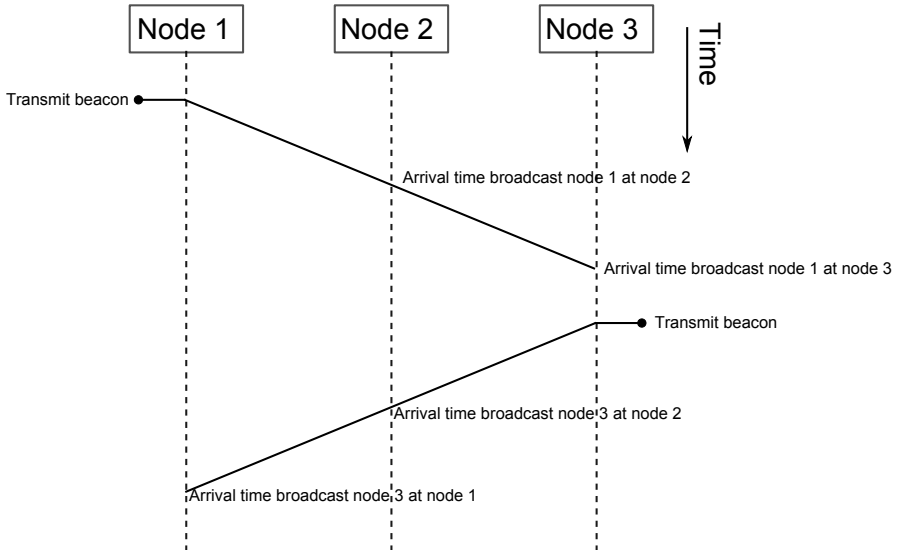


Figure 8.5: Timeline for sending beacons and measuring the arrival time at the receiving nodes. During the coordination phase all nodes measure the propagation delay with all their neigbors.

of the broadcast message of node 2. When node 3 receives the broadcast from node 1, it also records the arrival time of the message using his local clock. For using the cooperative localization and time-synchronization approach from Chapter 6, only the arrival times measured using the local clock are required.

The broadcast message also contains a processing delay field. All the nodes will calculate the processing delay of the message transmitted by the cluster-head. This can be used by the cluster-head to calculate a round-trip time using his local clock. This round-trip time does not contain any unknown clock errors. This is needed to resolve the time-space ambiguity present in the aLS-Coop localization algorithm. Figure 8.6 shows how the round-trip time measurement can be performed.

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

In the coordination phase, we use pure ALOHA, the optimal sending rate (G) can be calculated [1]. From the simulation results (Section 8.3) it becomes clear that the coordination phase can take a significant amount of time. Once the initial coordination is done, the 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 faster.

During the communication phase, the actual sensor data transmission is per-

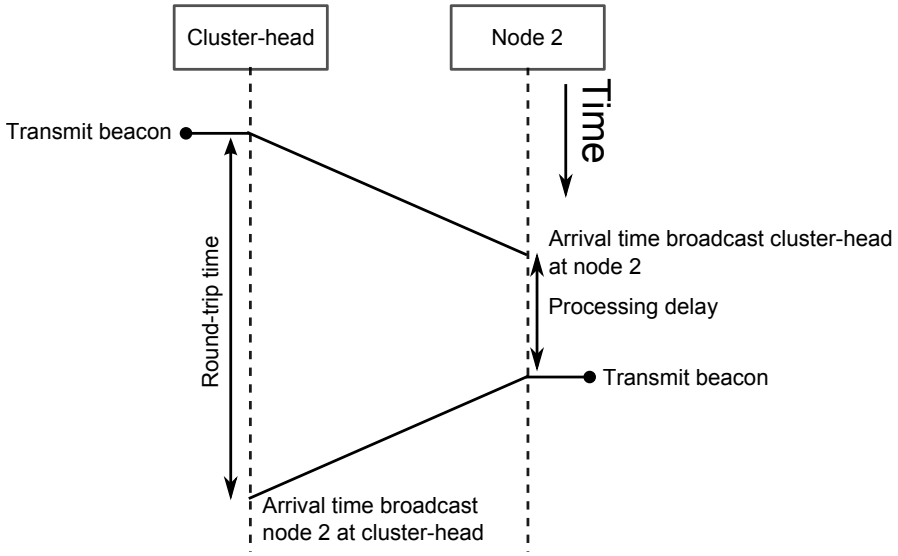


Figure 8.6: Measurement of the round-trip time. During the coordination phase, the nodes measure the processing delay of the broadcast from the cluster-head. This allows the cluster-head to estimate the propagation-delay without a clock error.

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	

Figure 8.7: Information contained in schedule message. Message is transmitted by the cluster-head at the start of the communication phase, to inform the network of the communication that needs to be followed during the communication phase.

formed. 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 8.7.

Before calculating a schedule, the cluster-head will first determine the routing in

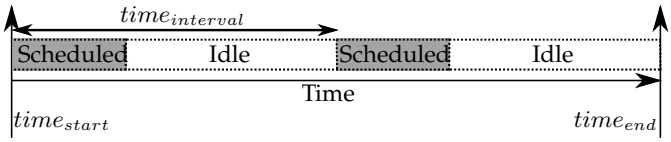


Figure 8.8: Example of how the operation phase of the network can be split up in a scheduled communication phase and an idle sleep phase. During the idle phase nodes can sleep and preserve energy, during the communication phase nodes can use scheduled communication thereby preventing collisions and preserving energy.

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	

Figure 8.9: Information contained in data message

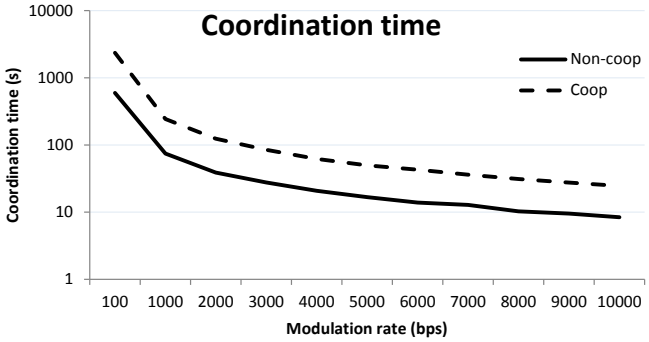
the network. For example in a data-collection network, the cluster-head will schedule all nodes to send their measurements to a gateway node. Other communication patterns can also be scheduled, according to the requirements of the application at hand.

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 Centralized Simplified Scheduling algorithm is used from Chapter 4.

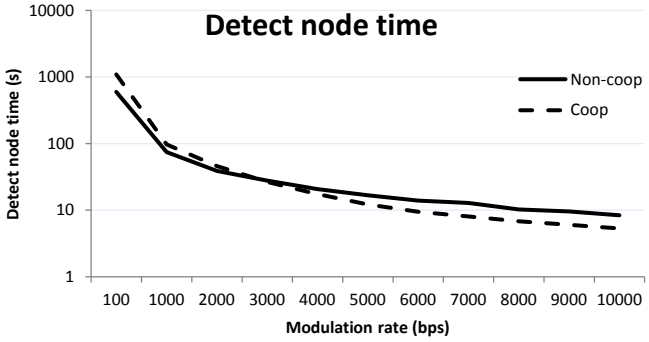
In Figure 8.9 the content of a data message is described. The packet contains the identifier of the source and destination nodes. The source and destination identifiers are the same if the message is a broadcast message and the data field contains the packet data.

The data messages described in Figure 8.9 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.



Rate	Non-Coop	Coop
100	594.91	2355.93
1000	74.57	243.17
2000	38.79	124.26
3000	27.70	84.69
4000	20.76	62.04
5000	16.76	49.82
6000	13.90	42.93
7000	12.83	36.17
8000	10.27	31.13
9000	9.54	27.55
10000	8.38	24.55

Figure 8.10: Time required to perform all time-of-arrival measurements and collect the measurements at the cluster-head. This is required to run the cooperative localization and time-synchronization algorithm.



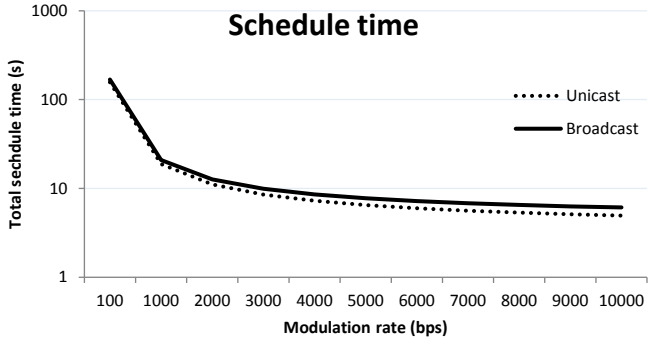
Rate	Non-Coop	Coop
100	594.91	1094.05
1000	74.57	97.18
2000	38.79	45.80
3000	27.70	26.53
4000	20.76	17.38
5000	16.76	12.16
6000	13.90	9.48
7000	12.83	8.04
8000	10.27	6.81
9000	9.54	6.03
10000	8.38	5.31

Figure 8.11: Time required for detection of nodes in the network. This is the time until the cluster-head has successfully received a message from all nodes in the network.

8.3 Performance evaluation

For the performance evaluation, we assume a network of 16 nodes uniform randomly deployed on an area of 500mx500m. We assume nodes are able to communicate with any other node in the network. In this scenario we run the MAC protocol and measure how the long coordination phase takes, the time it takes to measure all pair-wise propagation delays. We also measure the time until the cluster-head detects all nodes in the network. The result are shown in Figure 8.10 and Figure 8.11.

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 used for the non-cooperative MAC protocol in Chapter 7. 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



Rate	Broadcast	Unicast
100	168.30	157.01
1000	20.84	18.77
2000	12.65	11.09
3000	9.92	8.53
4000	8.55	7.25
5000	7.73	6.48
6000	7.19	5.97
7000	6.80	5.60
8000	6.50	5.33
9000	6.28	5.12
10000	6.09	4.95

Figure 8.12: Schedule lengths for all 16 nodes sending 128 bytes. Schedules are calculated in two different ways, by scheduling a broadcast message and by scheduling a directed unicast message.

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. For the scenario of 100bps, the coordination time for the cooperative approach is $\sim$2400 seconds or 40 minutes. Comparing the relative localization approach to an approach in which preconfigured beacons are used, we can see that the approach using preconfigured beacons yields lower coordination phase times. It is possible to use an cooperative approach when the network is static and the coordination only needs to be performed on initial deployment. When the data rates go up, the coordination times quickly go down to the order of minutes. With these data rates it is possible to perform more measurements in the same time. If the coordination time is stretched to 1 hour or 2 hour, more measurements will improve the localization accuracy.

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 8.12. The difference between beacons and unicast sending is only marginal. Therefore it makes sense to always use broadcast scheduling and move the routing decision to the nodes. This way no routing requirements ever need to be sent to the cluster-head, reducing overall communication overhead.

8.4 Conclusion

We have shown a design for a MAC protocol, which combines scheduled communication, localization and time-synchronization. This MAC protocol is designed for small clusters with limited number of nodes. We consider an integrated MAC design to be important because when scheduled communication is used, such is the case in our

MAC protocol, localization and time-synchronization are also needed. Our work is novel because it provides a combined MAC protocol for localization, time-synchronization and communication. Traditional MAC protocols consider these three aspects separately.

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 apply the aLS-Coop-Loc localization approach from Chapter 6 to calculate both the position and time-biases of the nodes in the network. During the communication phase, nodes communicate using scheduled communication. Using the calculated positions and clock-biases a schedule can be calculated.

We have perfomed measurements on the two phases of the protocol. For the coordination phase, 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. At the lowest bit rate we have simulated (100bps) the coordination time for the relative position approach is estimated to be around 40 minutes. We see this as a realistic time as we expect the coordination phase to be performed once when the network is deployed and the network is then switched to an operating mode in which it will run for months and years. Increasing the bit rate will allow more measurements to be performed in the same coordination time, thereby increasing the accuracy of localization and time-synchronization.

Using beacons with known positions outperforms relative positioning in terms of coordination time, 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.

Bibliography

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

Conclusion

CHAPTER 9

Conclusion

9.1 Summary

Existing underwater monitoring solutions use a small number of AUV to monitor large areas, this allows coarse-grained and offline monitoring. On the other hand, there is a need for fine-grained and real-time monitoring in multi-hop static networks of nodes. Such a network should provide the services of communication, localization and time-synchronization. Acoustic communication, the most widely used approach to underwater communication, poses significant challenges for the design of networking and communication protocols. Our hypothesis is that an integrated approach to MAC protocols combining localization, time-synchronization and communication has significant benefits over three separate solutions.

In this work we looked at communication scheduling approaches and combined localization and time-synchronization for UASNs MAC protocols. Communication scheduling provides efficient and reliable communication and overcomes the problem of long propagation delays caused by the slow propagation speed of the acoustic signal. Combining localization and time-synchronization overcomes the overhead of first having to do time-synchronization and then localization or visa versa. It allows localization and time-synchronization to be performed using one-way ranging, reducing the communication overhead required for localization and time-synchronization and reducing the energy-consumption.

Existing scheduling approaches are cumbersome and sub-optimal. Therefore in Chapter 3 we have derived a set of simplified scheduling constraints. Using this set of simplified scheduling constraints we have derived two scheduling algorithms which allow easy scheduling of communication. We have shown that our simplified scheduling approach outperforms existing approaches using timeslot scheduling.

In Chapter 4 we show how simplified scheduling can be used in large-scale multi-hop networks. We have introduced a distributed scheduling approach using clusters which reduces the computational and communication complexity. Comparison of the complexity of centralized and distributed approaches show a distributed approach is needed to allow scheduling of large-scale networks. We have also introduced an approach to allow scheduling with dependencies between transmissions. This allows aggregation of data from the outside of the network to a central data collection node and can be used to reduce the end-to-end delay of packets in a data-collection network.

In Chapter 5 we introduced a one-way ranging combined ToF and DoA local-

ization approach. Such an approach can be used to reduce the number of reference nodes required compared to ToF only approach. Using simulation and experiments in a dive-tank we evaluate the performance of the separate and combined approach. Results show that even in a difficult environment with many multipath present, accurate ToA estimation is possible.

In Chapter 6 we introduce a combined localization and time-synchronization algorithm for cooperative networks. While such an approach exists for non-cooperative networks, no such approach existed for cooperative networks before. The algorithm is flexible in the reference positions and reference measurement can be used even if it is only one dimension. Such a cooperative approach can be used in networks where only relative positions and time-synchronization is required and no or a limited number of reference nodes are available (due to practical constraints or constraints imposed by the application).

Using simulations and real-world tests we have evaluated the performance of the cooperative approach compared to a non-cooperative localization approach. Both simulations and real-world tests show that a cooperative approach outperforms a non-cooperative approach. Using SeaSTAR node and a COTS node from Kongsberg Maritime we have performed tests in real world environments. In these tests we evaluated the short-range ranging performance. The tests show that in short-range shallow water communication environments the relative ranging error is very significant and short-range ranging is difficult.

In Chapter 7 we propose the design of a MAC protocol for a scalable and energy-efficient underwater localization system for non-cooperative localization. Using the beacon scheduling approach from Chapter 4 transmission of beacons is scheduled. We have compared the performance of a scheduled approach to Carrier Sense Multiple Access (CSMA) for sending beacon messages. Our results show that when energy-efficiency is not of concern CSMA is the best choice for high modulation rates ($5kbps$ and $10kbps$). It has the best results in terms of time required for localization and number of received beacons. The scheduled approach is the best choice for low data rates ($1kbps$) and energy constrained systems. The scheduling of beacons guarantees collision free reception, maximizing the successful reception rate of the beacon messages and thereby increasing the reliability and decreasing the power consumptions of the system. Moreover it has the most predictable performance and shows little variations in performance between different simulation runs.

In Chapter 8 we have shown a design for a MAC protocol, which combines scheduled communication, localization and time-synchronization for small-scale underwater clusters. We use the cooperative scheduling approach from Chapter 6 to provide relative positions and time-synchronization even when no reference nodes are present in the network. Simulation shows that such a design is feasible even though the coordination phase may take significant time.

9.2 Conclusion

In this thesis we have shown that communication scheduling can be done in a simple way and improves reliability, througput and energy-efficiency of UASNs

and localization networks by avoiding collisions. Communication scheduling is not only for small-scale networks but can be scaled to large scale networks using distributed scheduling approaches and scheduling dependencies can be used in these large scale networks to improve end-to-end delay. These are important performance parameters to develop real-time and fine-grained monitoring applications which allow long term deployment.

Regarding localization and time-synchronization we have shown that by combining localization and time-synchronization significant benefits in terms of energy-efficiency and accuracy can be gained. Moreover the one-way ranging approaches are much better scalable to large number of nodes. This is important to enable realization of fine-grained large-scale monitoring applications. In UASNs a large number of nodes are present.

The application of communication scheduling and combined localization and time-synchronization in two different MAC protocols has shown that it is not enough to consider communication, localization and time-synchronization separately. When designing MAC protocols for UASNs one needs to consider the impact of all these services to develop an energy-efficient and scalable MAC protocol.

The BigMAC protocol shows that using one-way ranging localization and time-synchronization can be used to develop a scalable system for large-scale localization and time-synchronization. In this system the scheduled communication has a clear benefit in terms of energy-efficiency for all data-rates and time until localization for low data-rates. Because we use a one-way ranging approach to localization, the system is scalable and communication scheduling can be used.

The LittleMAC protocol shows how cooperative combined localization and time-synchronization can be used to develop an autonomous MAC protocol for small-scale clusters. The cooperative localization does not require the deployment of reference nodes, yet relative positioning and time-synchronization can be used to allow efficient scheduled communication during the operation phase of such a network.

In Chapter 1 we have split up the overall research question into two research questions. We will now look at how we have answered these research questions in our work:

- **How can energy-efficient, scalable and reliable communication scheduling be performed in small-scale and large-scale UASNs.** This question is answered in Chapter 3 and Chapter 4. In these chapters we have shown a simple scheduling algorithm which allows energy-efficient and reliable communication. Scheduling communication prevents packet collisions from occuring and thereby improves the reliability and energy-efficiency of the system. We have also shown how this communication scheduling can be scaled to large-scale networks.

- **How can localization and time-synchronization be performed in a energy-efficient, scalable and practical way.** This question has been answered in Chapter 6 and Chapter 5. In Chapter 6 we have shown a cooperative localization and time-synchronization approach. In Chapter 5 we have shown a localization and time-synchronization approach which combines ToF and DoA. Both approaches

use one-way ranging, increasing the energy-efficiency compared to two-way ranging approaches.

We have set the following overall research question:

How can communication, localization and time-synchronization be combined into an energy-efficient, reliable and scalable MAC protocol.

In Chapter 7 and Chapter 8 we have shown two different applications of the introduced communication scheduling, localization and time-synchronization techniques. By considering all three aspects (communication, localization and time-synchronization) a more efficient MAC protocol is designed. Evaluation of these MAC protocols proves our hypothesis:

An integrated approach to Underwater Acoustic Sensor Network MAC protocols, combining localization, time-synchronization and communication has significant benefits over three separate solutions.

9.3 Future work

In the future we would like to apply our techniques to real deployments of UASNs. The communication scheduling techniques described in Chapter 3 and Chapter 4 are not tested in real deployments. Applying the communication scheduling in real networks will give a better understanding of the performance benefits in real deployments and give a better understandig for which applications and environments the communication scheduling is suited.

The localization and time-synchronization experiments performed in Chapter 6 and Chapter 5 were done in a limited setup. The results from Chapter 5 indicate accurate localization is possible, the measured ToA variation in this experiment is very small. With the test performed in Chapter 6 the variation of the measurements were much bigger. We believe that by using the cross-correlation approach for ToA, the accuracy can be improved at the cost of increasing the computational complexity.

We would also like to improve the experiment setup and perform better quantification of the performance of ranging in realistic environments. The ranges supported by our own designed underwater node are limited and it would be interesting to see how well the localization performs over larger ranges. Quantification of more realistic environments such as a harbour would be interesting.

Regarding the localization approach in Chapter 6, we would also like to investigate the possibility of adding constraints to the optimization process. This would allow calculating the position of the nodes with fewer reference nodes. For example it would be possible to identify a node as being a surface node and all other nodes should be positioned below this node, this adds extra information to the reference node allowing even a further reduction in number of reference nodes required.

CHAPTER 10

List of publications

Bibliography

[1] 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.

[2] W. A. P. van Kleunen, N. Meratnia, and P. J. M. Havinga, "MAC scheduling in large-scale underwater acoustic networks," in *Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS 2011), Sevilla, Spain*. USA: SciTePress, July 2011, pp. 27–34.

[3] W. A. P. van Kleunen, N. Meratnia, and P. J. M. Havinga, "Simplified scheduling for underwater acoustic networks," *Journal of Networks*, vol. 8, no. 1, pp. 4–14, January 2013.

[4] 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, pp. 6:1–6:5.

[5] W. A. P. van Kleunen, N. Meratnia, and P. J. M. Havinga, "MDS-Mac: a scheduled MAC for localization, time-synchronisation and communication in underwater acoustic networks," in *The 6th IEEE/IFIP International Workshop on Ubiquitous UnderWater Sensor Network, UUWSN 2012, Paphos, Cyprus*. USA: IEEE Computer Society, December 2012, pp. 666–672.

[6] W. A. P. van Kleunen, N. Meratnia, and P. J. M. Havinga, "aLS-Coop-Loc: Cooperative combined localization and time-synchronization in underwater acoustic networks," in *CyPhy 2014*, April 2014.

[7] W. A. P. van Kleunen, N. Moseley, N. Meratnia, and P. J. M. Havinga, "Experiments with aLS-Coop-Loc cooperative combined localization and time-synchronization," in *WinMee 2014*, May 2014.

[8] W. A. P. van Kleunen, K. C. H. Blom, N. Meratnia, A. B. J. Kokkeler, P. J. M. Havinga, and G. Smit, "Underwater localization by combining time-of-flight and direction-of-arrival," in *Oceans 2014*, April 2014.

[9] D. J. A. Bijwaard, W. A. P. van Kleunen, P. J. M. Havinga, L. Kleiboer, and M. J. J. Bijl, "Industry: Using dynamic wsns in smart logistics for fruits and pharmacy," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, SenSys 2011, Seatle, WA, USA*. New York: ACM, November 2011, pp. 218–231.

[10] K. Zhang, W. A. P. van Kleunen, N. Meratnia, P. J. M. Havinga, and E. Tijs, "Underwater sensing and communication platform," in *OCEANS 2011, Santander, Spain*. USA: IEEE Oceanic Engineering Society, June 2011, pp. 1–6.

Appendices

APPENDIX A

Description of SeaSTAR and Kongsberg Mini node

This chapter describes the hardware that was used during the experiments performed in Chapter 6. For the experiment the Kongsberg Mini described in Section A.1 and the SeaSTAR node described in Section A.2 were used.

A.1 Kongsberg Mini

The Kongsberg Mini node is a COTS node from Kongsberg design for UASNs. In [2] the node is described. The Kongsberg Mini is member of a family of underwater nodes ranging in size from the Maxi to the Mini. The Mini node is the smallest member of the family and contains the smallest battery. The nodes have a stainless steel enclosure allowing operating depth of up to 7000 meters. The node contains a Gumstix embedded Linux computer for easy application development. Nodes can be deployed using a floating collar and an acoustic release, using the remotely controlled acoustic release the node can be released and retrieved from the ocean or sea floor.

The nodes work in the 20khz to 30khz acoustic band and used Direct-Sequence Spread Spectrum (DSSS) signaling with a variable spreading factores. Communication rates range from 200bps to 1710bps. The Kongsberg Mini has demonstrated communication ranges of up to 3800 m horizontal. Figure A.1 shows a picture of the Kongsberg underwater nodes.

A.2 Seastar

This node was developed within the SeaSTAR project. The main motivation for developing this underwater network node is that all available products known to us were either very expensive, i.e. more than a few hundred dollars per node, did not support accurate ToA measurements, or were closed hardware platforms and did not allow us to control modulation and ToA estimation.

The two main design goals for the node were to make a low-cost and light-weight system which is easily adapted to our current and future research activities and to be able to operate the nodes continuously for at least 24 hours. To keep the cost down, many subsystems were implemented using commercially available parts. The basis of the node is an off-the-shelf ARM Cortex-M3 board which was extended by a custom

Figure A.1: Example of the Kongsberg Maxi, Midi and Mini.

analog acoustic transceiver front-end and a HopeRF RFM22B 433 MHz RF wireless link module. A three-cell 2200 mAh LiPo battery was used to power each node.

Most of the battery power is used when the node is transmitting. Good transmit mode power efficiency is ensured by employing a switching-mode power amplifier architecture. This limits the modulation formats to those with a constant envelope, i.e. Frequency-Shift Keying (FSK) and Phase-Shift Keying (PSK). Two-tone FSK was chosen as our modulation format so carrier synchronization at the receiver could be avoided, thereby reducing system complexity. Following the software-defined radio paradigm [3], the majority of the acoustic MODEM functions are implemented in software for maximum flexibility.

A system-level block diagram of the node is shown in Figure A.2.

In receive mode, the transducer picks up acoustic energy in the 20 to 30 kHz band and produces a signal which is then amplified by a variable gain amplifier (VGA) and several fixed-gain amplifiers. The final receive amplifier drives the on-board 12-bit analog-to-digital converter (ADC) of the ARM processor. Further processing of the received signal is performed in software on the ARM microcontroller.

In transmit mode, the ARM processor configures an on-board timer/counter to

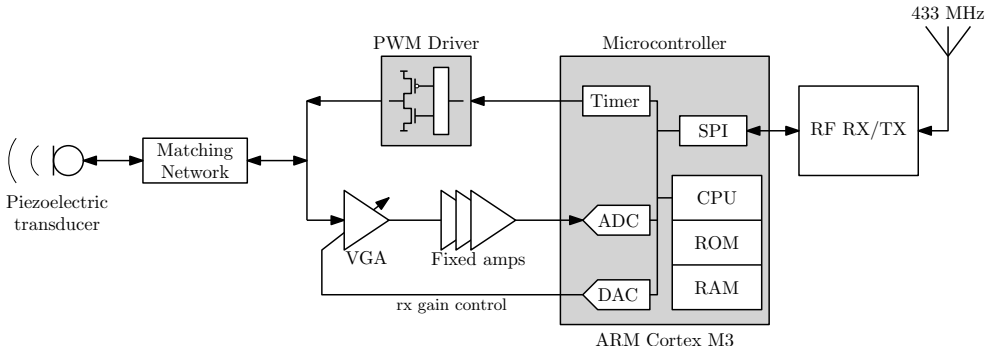


Figure A.2: A system-level block diagram of the underwater sensor node.

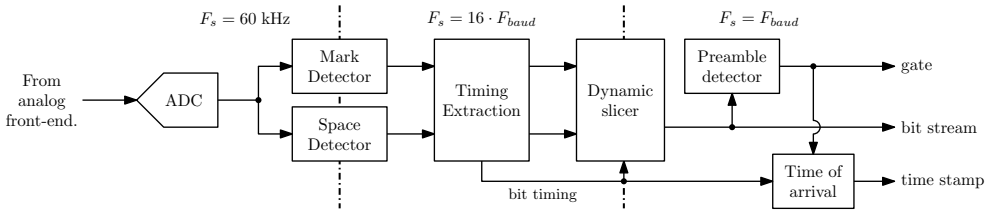


Figure A.3: A system-level block diagram of the FSK decoder with ToA support.

generate a square wave signal of the desired frequency determined by a '1' or a '0' bit. This signal is amplified by the PWM driver and converted into acoustic energy by the piezoelectric transducer.

The power consumption of the node was measured at 6 W in transmit mode and 700 mW in receive mode.

The architecture of the software-based FSK decoder is shown in Figure A.3. Two tone detectors are employed, one for each FSK tone, to determine the energy present around the tone frequencies. From these (changing) energy levels, the timing of the bit stream is extracted using an early-late synchronization loop, which is updated at sixteen times the baud rate. This timing information is used to sample the energy levels at the baud rate and subsequently used by a dynamic slicer to decide whether a '0' or a '1' was received. The bit timing information is also used to determine the time of arrival of the bit. Given that the timing extraction runs at 16 times the Baud rate, the time of arrival has a resolution of $\frac{1}{16 \cdot F_{baud}}$ seconds, where F_{baud} is the Baud rate.

The packet layout used is shown in Figure A.4. To ensure quick timing acquisition,

SYNC	PREAMBLE	LENGTH	PAYLOAD	CRC
10101010	11 bits	8 bits	variable	16 bits

Figure A.4: The layout of a packet as transmitted by the MODEM. The packet contains a SYNC for synchronization purposes and a preamble to allow detection of the start of the packet with high accuracy. A CRC is added to check the validity of the variable sized packet payload.

each packet starts with an alternating synchronization bit pattern, i.e. "10101010", providing maximum timing information to the timing synchronization loop. A fixed preamble based on an 11-bit Barker sequence [1] directly follows the synchronization pattern to allow detection of a valid packet. The packet payload has a variable size of up to 128 bytes and is protected by a cyclic redundancy check (CRC) [4] word of 16-bits.

In addition to the underwater MODEM, each node can also be reached through its 433 MHz RF wireless transciever. This is used for node health monitoring, debugging purposes and collecting measurement data.

During the experiments, the nodes used a transmission rate of 100 Baud, giving the ToA timestamp a resolution of 0.625 ms. This equates to about 0.9 m distance assuming a propagation speed of 1500 m/s.

The total cost of the node electronics, excluding the piezoelectric transducer, is below 100 US dollars.

Bibliography

[1] S. Golomb and R. Scholtz. Generalized barker sequences. *Information Theory, IEEE Transactions on*, 11(4):533–537, 1965.

[2] T. S. Husøy, F. R. Knudsen, B. Gjelstad, and A. Furdal. Product development at kongsberg maritime related to underwater sensor networks. In *Proceedings of the Seventh ACM International Conference on Underwater Networks and Systems*, WUWNet '12, pages 16:1–16:5, New York, NY, USA, 2012. ACM.

[3] J. Mitola. The software radio architecture. *IEEE Communications Magazine*, 1995.

[4] W. Peterson and D. Brown. Cyclic codes for error detection. *Proceedings of the IRE*, 49(1):228–235, 1961.

APPENDIX B

Levenberg-Marquardt iterative optimization

For calculating the positions of the nodes in the network commonly an iterative optimization approach is used. For optimization we use LMA [1], a well known and popular numerical solution to the problem of minimizing a function. The LMA optimizes the parameters x of a function $f(x)$ to find a local and possibly global minimum such that the sum of the squares of the residuals (r) becomes minimal:

$$f(x) = \sum_{i=1}^{m} r_i^2(x) \tag{B.1}$$

In each iteration step, the parameters vector x is updated by a new estimation $x + \delta$. To determine δ the function is approximated by its linearization.:

$$f(x_i + \delta) \approx f(x) + J_i\delta \tag{B.2}$$

where

$$J_i = \frac{\partial f(x)}{\partial x} \tag{B.3}$$

For each iteration step, LMA calculates the update step δ by solving:

$$(J^T J + \lambda diag(J^T J))\delta = J^T f(x) \tag{B.4}$$

Where J is the jacobian matrix whose i^{th} row equals J_i for every i. The damping factor λ is updated at each iteration. A smaller value of λ will bring the algorithm closer to the Gauss-Newton algorithm, and a large λ will result in the algorithm operating similar to the gradient descent approach.

Bibliography

[1] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.