A lightweight protocol suite for underwater communication

Helge Rustad
SINTEF
Trondheim, Norway
helge.rustad@sintef.no

Abstract—**A lightweight MAC and network layer protocol suite for underwater acoustic communication has been developed. The protocol suite covers the MAC layer, hop-by-hop acknowledgement, multi-hop routing and transmission and includes a network discovery and route generation module.**

The protocols are currently implemented in a simulator for verification and performance evaluation. By summer 2009 the protocols will be implemented in real underwater modems as part of a project demonstrator[1].

Keywords: Acoustic underwater communication protocols

I. INTRODUCTION

Underwater acoustic network communication is still a young discipline. It is predicted that this mode of communication will be important in many types of sensor networks for example in the oil industry or for marine environmental monitoring.

Acoustic communication resembles radio communication in many ways, but energy limitations, low bitrate, high attenuation and high delay introduce new challenges and require new or modified protocols compared to radio communication. Both FDMA and TDMA based solutions with random access or reservation schemes are described in the literature, see [1] for a general overview. The IEEE 802.11 CSMA/CA based MAC protocol is a popular choice for adaptation to underwater networks. Random access solutions are generally simple and robust, while more elaborate schemes can offer higher throughput and better energy efficiency, especially in stable high load situations. See for example [2].

Our project focuses on selection and implementation of a set of protocols for our demonstration system rather than doing basic protocol research. We also base our selection on usage scenarios from our project. These include the following:

- The system should be self-configuring when deployed in the water. It should be possible to add or remove nodes. Nodes are stationary and there are up to a few tenths of them.
- The network includes a master node (with link to an outside system) that is the main sink for sensor data and source of commands.
- The network may include both long and short hops. Long hops can use high power, 100W or more with round trip delays of several seconds. Multi-hop routing is required.
- Traffic consists mainly of low volume sensor data and occasional high volume bulk transfers. Bulk transfers can be coordinated by the master node to avoid unnecessary competition among nodes.

Energy limitations are important. A node which is allowed to use 5 kg of Lithium batteries over two years for its 100 W transmitter can transmit around 0.1% of the time. Obviously the mean traffic from such a node must be low.

We find that user scenario constraints are often neglected. Our project will end up in a demonstrator and practical, technical and economical constraints are as important as theoretical performance. For this reason we have chosen a relatively simple CSMA/CA protocol as a basis for our MAC layer. We intend to show that this choice will perform well in our network and traffic scenarios:

- In low traffic situations with a low probability of collisions it is well known that CSMA/CA performs well.
- In high traffic situations we will show that the protocol will not collapse.
- High traffic situations which arise as a result of bulk data transfers can be coordinated by the master using polling and window based protocols that reduce the number of collisions and enhance energy efficiency and throughput.

The simulation work done in this project focuses on MAC and network protocols. As a simplification (that is not always realistic) nodes are considered to be omnidirectional and the water is regarded as an idealistic medium with well defined attenuation and delay as a function of distance.

II. PROTOCOL DESCRIPTION

A. *The MAC protocol*

Our MAC protocol is based on the CSMA/CA protocol used in Wi-Fi networks (IEEE 802.11). The basic rule is: Transmit when the water is silent. The addressed node will return an acknowledgement immediately. If no

[1] This work is part of a government and industry funded Norwegian project called "Nordområdenes Nye Nervesystem" (NNN).

acknowledgement is received wait for a random time (growing exponentially for repeated unsuccessful repetitions) before retransmitting.

Before sending long packets an RTS/CTS (Request To Send / Clear To Send) exchange can be used to reserve collision-free time. Long transmissions are costly in terms of energy and if a short RTS/CTS exchange can reduce the collision probability significantly the total energy budget and the throughput will benefit.

In 802.11 - where propagation delay is negligible compared to transmit time - the backoff interval is calculated as a number of slot times where the slot time is the maximum round trip time in the system. This quantization of the backoff time has the effect of reducing the probability of collision at the next transmit attempt. The slotted nature of the backoff interval has less meaning for us however due to the much longer propagation delays compared to packet transmit time. We still use a slot time as a means for calculating backoff, but it normally much shorter than the maximum roundtrip delay for optimal performance.

Basic CSMA/CA is not an optimal protocol regarding collision avoidance and medium utilization and numerous improvements for underwater communications have been proposed in the literature. We have chosen the basic protocol mainly for its simplicity and since most of our scenarios predict either very low traffic or traffic that can be regulated by a master node.

We have implemented some enhancements to the basic protocol.

- The nodes know in advance (from the network discovery phase) what transmission power level is needed to reach each neighbour and they use a power level slightly higher than the minimum required to reach the intended destination. For very long hops more powerful coding (lower data rate) can be chosen to extend the reach still further.
- RTS/CTS packets can be sent with increased power to ensure that all possible disturbers can hear these packets.
- After network discovery the nodes know the distance and delay to the destination and can calculate the exact time to wait for an acknowledgement.

Retransmission after missing acknowledgement is an integral part of our protocol.

B. *Multi-hop routing*

Any network of some size requires multi-hop routing. In underwater communication multi-hop is even more important than in radio networks. The high attenuation and high transmitter power levels that are required make two short hops much more energy efficient than one long hop. If a higher data rate can be used on the short hops even the throughput can be higher if a long hop is split in two.

Our protocol suite includes two modes of multi-hop routing.

- The network discovery protocol creates as a by-product a "cheapest path" from every node to the master node. This path is stored as a next hop pointer in each node. This path can be used for data reporting from the nodes directly after the network discovery.
- In the other routing mode each packet is provided with a complete route (list of nodes) in the header. This routing mode allows routing between any nodes via any possible route. After the network discovery phase the master node has acquired the information required to use or distribute tables for this kind of routing.

The multi-hop routing uses MAC layer hop-by-hop acknowledgement and retransmission. There is no end-to-end acknowledgement integrated with the routing protocol.

We have implemented two main queuing schemes. With *FIFO queuing* there is one outgoing queue in each node. With *fair queuing* the transmit queue is ordered so that each flow (start/end-node pair) is allowed to transfer the same number of packets (a slight simplification since we do not take packet size or transmission time into account)

C. *Network discovery protocol*

Our protocol suite includes network discovery. It is based on the well-known flooding principle. The master node initiates the algorithm by sending a FLOOD packet. Every node hearing it repeats the FLOOD packet with a random delay. Every message includes information on when it is sent (using the sender's clock) and which other nodes the sender has heard with information on signal quality and timing. After a while every node will know its neighbours and the delay and signal quality parameters (attenuation) to each of them. Each node will transmit a FLOOD packet enough times for this information to be generated.

This algorithm does not depend on synchronized clocks in the nodes. After the packet exchange A->B->A node A can calculate the net round trip time by subtracting the delay in B (reported by B) from the total round trip time (measured with A's clock). When A has retransmitted FLOOD once more B can also calculate the round trip delay. As a by-product the network discovery protocol can be used to synchronize all nodes to the master's clock.

The FLOOD packets include an accumulated routing cost enabling each node to calculate the best (cheapest) neighbour for routing data towards the master node. This information is stored locally and later reported to the master node. Typically the least cost will be a weighted sum of hop lengths and hop count optimizing for either throughput or energy efficiency.

The next phase in the network discovery protocol is to report the result of the FLOOD back to the master node using acknowledged transfer. Each node reports to its "cheapest cost" neighbour identified earlier. Information is aggregated on the way to the master node to reduce the amount of information that will be sent.

The algorithm can easily be extended to handle new nodes entering the system. When a new node appears it transmits a FLOOD message. Nodes hearing this will take up again and continue the previous FLOOD session. The neighbours to the new node will be detected and reported. The new node will not necessarily be incorporated in the least cost routes immediately but the master node has the opportunity to calculate and distribute new routes or start a new FLOOD. The FLOOD algorithm can also be rerun regularly or when there is indication of outdated routing data or missing links or nodes.

An algorithm like this is by nature non-deterministic. There is a chance that a link or node will not be detected due to collisions. With suitable configuration settings and automatic collision avoidance rules this risk can be reduced to an acceptable level. The selection of configuration parameters will always be a compromise between risk of missing links or nodes, algorithm execution time and energy consumption.

The routing information reported to the master node can be used as a basis for calculating and distribution of more advanced routing tables with alternate routes for load sharing and reliability but this is not yet a part of our project.

III. Implementation and simulation

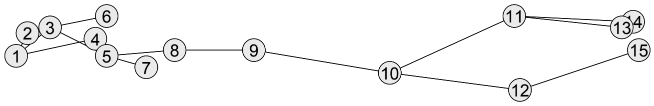
The protocols have been implemented in a simulation environment implemented in the Python programming language using the simulation extension SimPy and plotting routines in Pylab. The simulation environment allows simulation of a network of any topology and size. The protocol modules have been implemented in a way that is close to a real system implementation. This allows easy porting of the implemented algorithms to the demonstration system.

To limit the complexity of the simulation system several simplifications to real world situations have been made. The simulator models sound propagation in an idealistic medium. Delay and attenuation are calculated as a function of distance. Attenuation is modelled according to equations set forth in [3] taking absorption and spherical spreading into account. The nodes can adjust transmit power and data rate to optimize reach and energy use. Our simulations are done with rather short packet lengths of 200-400 bytes to emphasize the effects of long propagation delays. The nominal bitrate used is 4000 bits/s but the modems can lower the data rate to obtain a longer reach.

We have simulated many network topologies but most simulations have been done with a network topology representing a set of nodes more or less randomly placed along a line with the master node at the end as shown in Fig. 1. This topology includes both short single-hop connections and long multi-hop connections.

The simulated network is 9 km long, a length that will emphasize the effect of long propagation delays. This network has been used for all results shown in this paper. Other topologies we have simulated give comparable results.

The simulator visualizes many details of the protocol behaviour. Fig. 2 shows a snapshot of signal propagation in the water. The plot shows a situation where two messages are transmitted at the same time in the network without interference. The long delays also allow messages transmitted simultaneously to cross paths in the water and be received at different places without interference.



Routes shown are least cost routes towards the master node (1) when energy cost dominates.

Figure 1. Network topology used in most simulations

IV. Simulation results

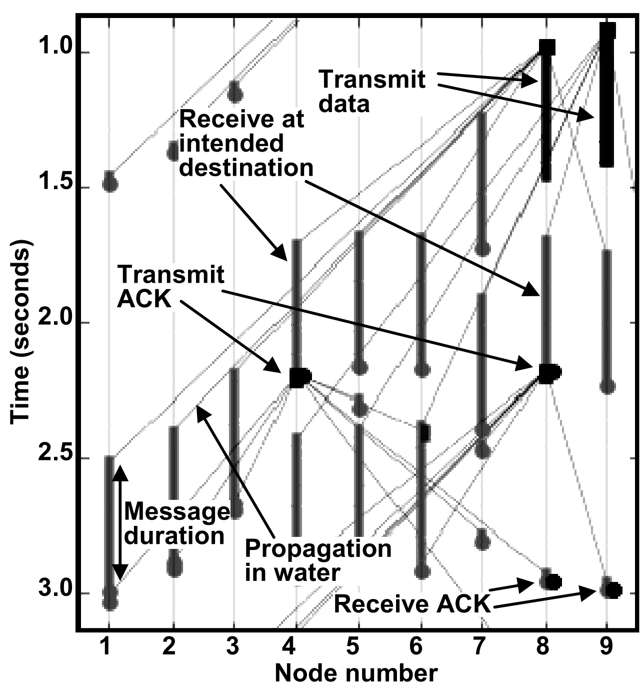
Our simulator can simulate the node discovery protocol. We can vary parameters like delay before resending FLOOD and the number of FLOOD repeats to find settings that give an acceptably low probability of missing nodes and links while consuming a reasonably low amount of energy and time. We have not included results from these simulations here as they are quite straightforward.

A. Protocol stabilit

We will continue by showing that the MAC protocol is well-behaved and does not break down as the offered load rises above the network capacity.

Fig. 3 shows simulation results where every node sends one-hop messages to every neighbour node within reach. There is no other flow control than the MAC backoff mechanism. We see that there is no breakdown as the network reaches saturation. In the saturated situation retransmission ensures that there is still no packet loss, but the transmit queue in every node is always full.

The seemingly small glitch in the curve if Fig. 3 as it enters saturation has been investigated in detail and shown to result from some regularity that appear in the transmission patterns as all queues become constantly filled.



Clock time (seconds) downwards. Node numbers (1-9) horizontally. The diagram shows message transmission, duration, propagation and reception. Original colour coding not visible here shows collisions and special events.

Figure 2. Simulator sound propagation plot

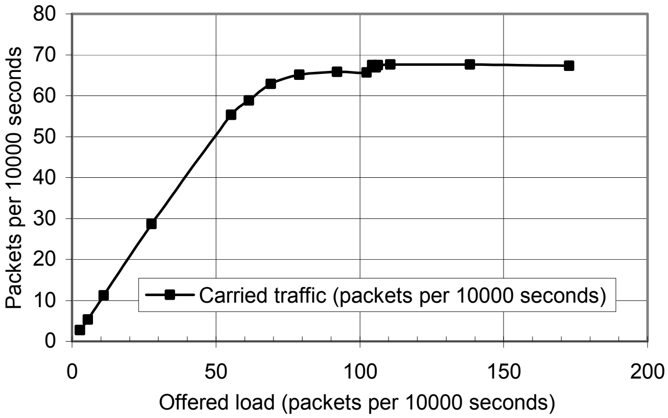


Figure 3. Carried versus offered load

In the next scenario simulated a window based transport layer flow control protocol has been added. One data flow is set up from each node towards the master. The master uses the window mechanism to constrain all flows to the same rate. This mechanism is equivalent to polling from the master with several active polls in the network at the same time.

Fig. 4 shows that the network still behaves nicely. As expected the number of retransmissions due to collisions grows as the traffic rises but we note that the rise is smooth up to the saturation points at the end of the curves. As expected large window sizes (more aggressive load) results in more collisions, but also slightly larger maximum throughput.

B. *Configuration parameters and fairness*

The minimum and maximum backoff times are in 802.11 controlled by the slot time and the CWmin and CWmax parameters (minimum and maximum backoff range). In a WLAN network transmit power is negligible, delays short and networks are optimized for throughput with rather large retransmission rates. In underwater communication energy is costly and delays long. Therefore parameter selection becomes a compromise between throughput and energy cost. In very low load situations long backoff times are favourable, but collisions are rare anyway and optimal parameter selection is not very important.

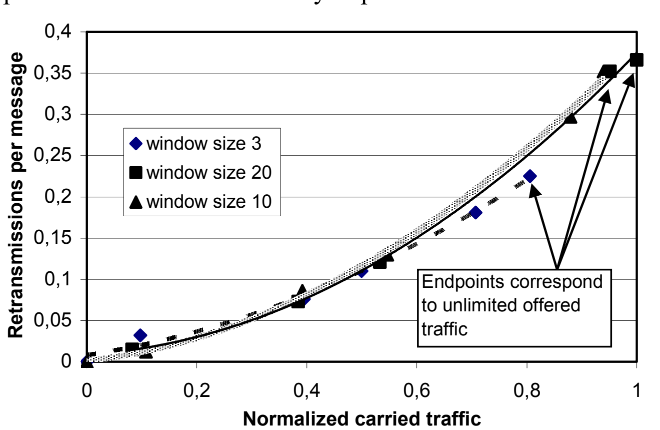


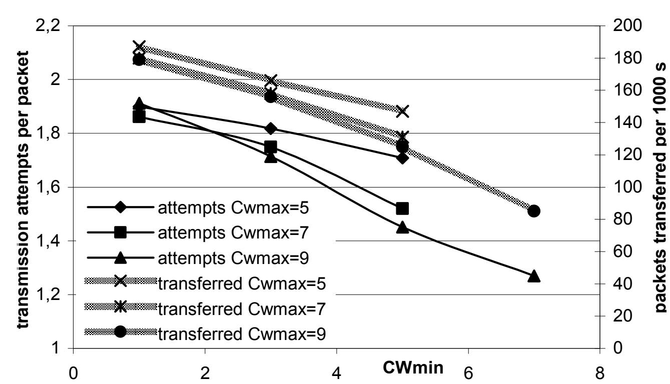
Figure 4. Retransmissions versus load

In high load situations parameter selection is more important. We have studied a high load scenario that is a combination of fixed load trickle traffic combined with one bulk transfer that uses all remaining capacity. Fig. 5 shows that CWmin is more important than CWmax. Actually the important factor is slot-time^{CWmin} since the slot time (chosen arbitrarily as 0.1s) is shorter than typical delays and has no physical relevance in itself.

Fig. 5 shows a simulation where trickle flows from all nodes consume around 40% of the capacity while a single bulk transfer uses a long multi-hop connection. The fair queuing scheme ensures that the small trickle flows are not much affected by the bulk transfer. Not unexpectedly a longer minimum backoff time results in fewer collisions but also lower throughput. The optimal values will therefore be a compromise between throughput and energy efficiency.

Other simulations have shown that if the bulk transfer uses a short single-hop connection low CWmin values (1 to 3) result in unfairness that impairs the trickle flows. The reason for this is that if CWmin is low nodes close to a transmitting node are favoured for the next transmission. Low delay allows them to start before the competitors and low attenuation reduces the probability of destructive collision.

We will next look at a scenario where there is one flow from each node to the master node offering unlimited traffic. We will look at different ways to regulate the traffic to minimize the unfairness between flows. As a measure of unfairness we have chosen the throughput on one of the one-hop flows divided by the throughput of a selected 7-hop flow. Ideally both flows should get the same resources but typically the one-hop flow will get a higher throughput. In Fig. 6 the unfairness of simple *FIFO queuing* and *fair queuing* are compared as a function of window size when a transport layer end-to-end protocol is used. These curves are compared to *sequential polling* where each node is polled for a large bulk of data at a time and *enforced fairness* where the master node regulates the traffic (statistically) so that each node can send the same amount of data. As we see the unfairness is high except when enforced.



Bulk transfer of data from node 14. Fixed rate unacknowledged traffic (~40% of capacity) from all other nodes. Slot time=0.1s. Fair queuing.

Figure 5. Effect of CWmin and CWmax

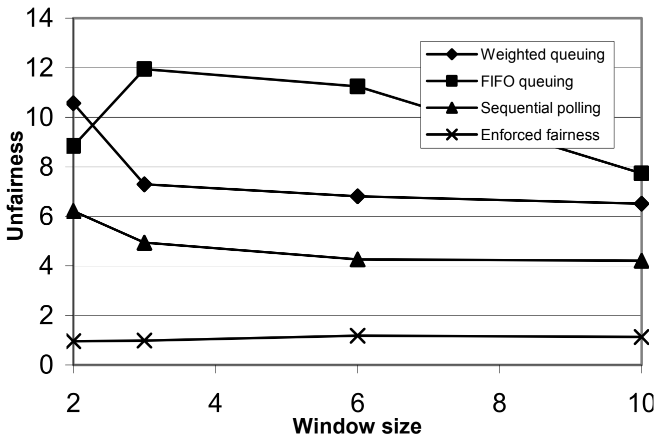


Figure 6. Unfairness versus window size

There are several causes to the unfairness:

- The MAC protocol grants access to nodes, not to flows. Multi-hop flows must normally share medium access with many flows at several nodes while a single-hop flow might originate at a node with no other traffic.
- Multi-hop messages consume more medium resources. At transit nodes the outgoing and ingoing links need non-overlapping access time.

Acknowledgements consume costly resources, especially on long hops where propagation delays can dominate over packet length. Reducing the number of end-to-end acknowledgement packets frees resources and Fig. 7 shows some effects of this. An unexpected result is that the unfairness grows dramatically when reducing the number of acknowledgements. A closer inspection of the simulation results show that with one end-to-end acknowledgement packet per forward packet the output queue in the master node is a bottleneck that enforces some fairness through fair queuing. When this bottleneck is reduced the single-hop nodes close to the master gain a large advantage and seize most of the capacity of the communication medium.

C. *Mixed loads*

A typical scenario might be that many nodes send small amounts of sensor data regularly to the master node (trickle traffic) while a bulk transfer of data goes on between one node and the master node. The trickle traffic could also be sent without end to end acknowledgement and flow control (except for the MAC layer acknowledgement and retransmission). A loss of some measurements might not outweigh the cost of end-to-end acknowledge on all of them. The bulk transfer would need an end to end protocol however. We have studied the impact of the bulk transfer on the trickle loads.

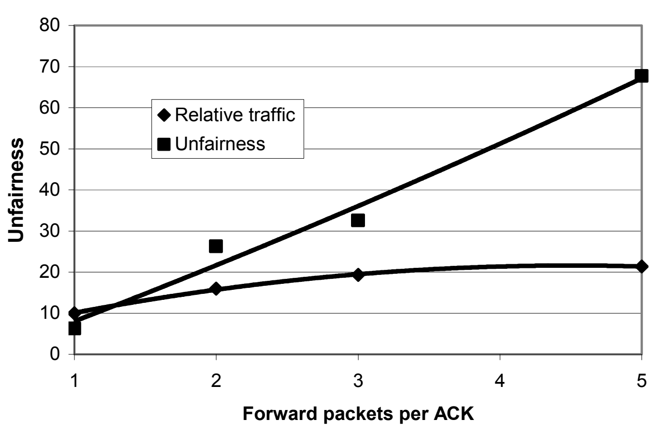
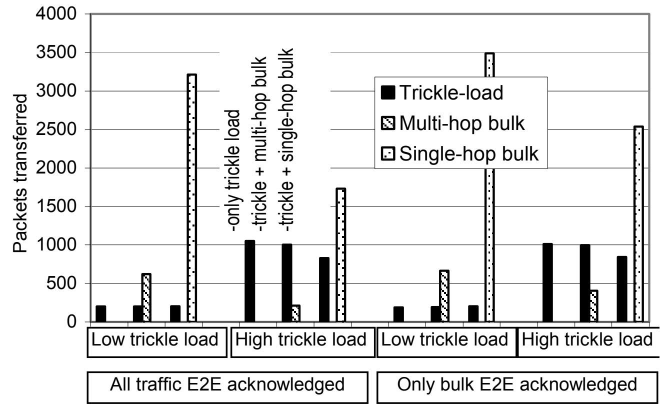


Figure 7. Unfairness versus ACK frequency

Fig. 8 shows that the trickle throughput is relatively unaffected by the bulk load whether the trickle load is end to end acknowledged or not.

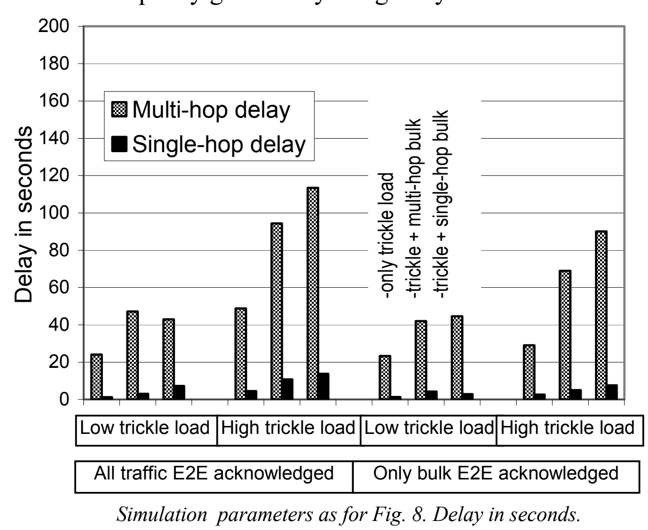


Low trickle load is around 7% of capacity (sum of all sources). High trickle load is 45% of capacity Trickle traffic shown alone, combined with one multi-hop bulk transfer and combined with one single-hop bulk transfer.

Figure 8. Trickle load and bulk transfer

Fig. 9 shows the impact of the bulk transfer on the delay of single-hop and multi-hop trickle flows. We see that the delays for high trickle loads grow quite much, especially when also the trickle flows are end-to-end acknowledged. This simulation has been run with a low window size of three packets per flow.

We know from other simulations that using a larger window size results in even larger delays for all flows while the traffic capacity grows only marginally.



Simulation parameters as for Fig. 8. Delay in seconds.

Figure 9. Bulk transfer influence on trickle flow delay

V. Conclusions

We have implemented and simulated a set of communication protocols for the demonstrator in the Norwegian NNN project. The protocols include MAC layer, network discovery and multi-hop routing. The protocols will later be implemented in the physical demonstrator nodes.

Simulations have shown that the protocols are well-behaved in overload situations and perform as expected. In high load situations fairness must be enforced by higher layer protocols if it is required since the MAC layer can be very unfair.

Our MAC layer is for simplicity based on basic CSMA/CA. Modifications to this protocol that can offer better throughput and energy efficiency have been proposed in the literature and may be added to our protocol suite in the future.

VI. References

[1] [1] I. F. Akyildiz, D. Pompili, and T. Melodia, "State of the Art in Protocol Research for Underwater Acoustic Sensor Networks," ACM Mobile Computing and Communication Review, vol. 11, no. 4, 2007.

[2] [2] Affan Syed, W. Ye, and John Heidemann. T-Lohi: A new class of MAC protocols for underwater acoustic sensor networks. In Proc. of IEEE Infocom, Phoenix, AZ, USA, April 2008, pages 231—235.

[3] [3] J. M. Hovem, “Underwater acoustics: Propagation, devices and systems”, POLECER International Conference, Piezoceramics for end-users II, March 508, 2006, Hafjell, Lillehammer-Norway.