A scalable electro-magnetic communication system for underwater swarms

Felix Schill [1] **Uwe R. Zimmer** [2]

[1] *Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: Felix.Schill@epfl.ch)*
[2] *The Australian National University, Canberra, Australia (e-mail: Uwe.Zimmer@anu.edu.au)*

Abstract: Scalable, reliable underwater communication is still a major challenge limiting the deployment of large-scale cooperative underwater systems, such as swarms of AUVs, which generally rely on frequent updates from surrounding units for control and mission objectives. Communication bandwidth is extremely limited in the underwater domain, hence it is crucial to efficiently manage and distribute this scarce resource between nodes. This paper discusses a symmetric, distributed time-slotted channel access algorithm for large-scale, dynamic ad-hoc networks. In contrast to traditional network algorithms which are mostly designed for sporadic point-to-point communication, this algorithm specifically optimises fast, continuous information distribution, locally and globally, while achieving close to 100% channel utilisation. A theoretical analysis of the performance is presented, along with real-time simulation results for dozens of nodes performing rapid reconfigurations. The impact of swarm dynamics on network performance is analysed in detail.

Keywords: Communication protocols, information flows, distributed scheduling, swarm communication, ad-hoc networks, dynamic behaviour

1. INTRODUCTION

Cooperative multi-robot systems (e.g. swarms, formations) rely on scalable real-time communication to distribute their state locally among their neighbours, and also globally among the whole group, for example to find a consensus, change the mission objectives, or aggregating sensor data. While most communication technology focuses on unicast or broadcast, swarms require all-to-all communication, or omnicast (also known as global gossiping). An analysis of omnicast was presented in Schill et al. (2005). Robots in swarms generally exchange their state information regularly among neighbouring nodes to enable collective control strategies. In most robots the energy required for communication is small compared to propulsion requirements. Communication of state information should therefore happen as often and quickly as possible, and the available channel should be used fully.

The requirements for a communication system to enable control strategies for robot swarms or formations are therefore continuous many-to-many communication with low, predictable latency and fast information dissemination among robots, both locally and globally, while being robust towards rapid changes in network topology. However, most communication technology available today is designed for sporadic one-to-one communication, and will not perform well for continuous communication with high channel utilisation. While this mismatch can often be ignored in aerial systems when network utilisation is low, the challenge is far more pronounced in the underwater domain, as bandwidth is severely limited. A common problem with wireless networks is that the network topology is not known in advance and can dynamically change - especially in robotic swarms where nodes are mobile. Medium access therefore has to be adaptive and robust to changes. Furthermore, at start up of a multi-hop radio network, there is no prior communication infrastructure. This poses a bootstrap problem, as information has to be exchanged in order to identify and distribute the current network topology, the number and identity of participating nodes and parameters for the medium access algorithm. Solving these problems is commonly called *ad-hoc networking*, meaning that a network configures and maintains itself automatically as nodes are added, moved or removed, without initial knowledge of the network topology.

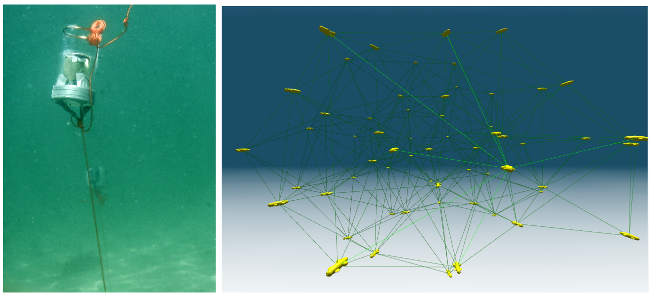


Fig. 1. Left: The longwave radio transceiver module. Right: 3D swarm and networking simulation.

This paper discusses a scalable communication solution, based on longwave radio communication and a time-division multiple access algorithm, for dynamic ad-hoc networks as found in robotic swarms. The solution is optimised for high channel utilisation and deterministic, fair channel access. The scheduling algorithm has been implemented and tested on a low-power embedded radio module, using a 122kHz carrier and binary phase shift keying. The communication system is tested in a real time swarm simulation to analyse performance in fast-changing swarm topologies.

2. MEDIUM ACCESS WITH MULTIPLE TRANSMITTERS

When there is more than one transmitter accessing the same medium, the problem of interference occurs. There are several possibilities to access a medium avoiding interference - transmissions can be separated by time, space, frequency, polarisation, medium, modulation or other means. In swarm networks it is desirable that all nodes within range of a transmitter are able to receive the message. Separation on different channels (e.g. frequency, medium, polarisation) is therefore not ideal as it would require multichannel receivers, and the number of available channels would limit scalability. Nodes are generally already separated in space, which means that the the total available medium can be split into separate regions that do not interfere. This still requires separation in time as well though if the network is to be connected. Separation in time is achieved by scheduling transmissions so that no two transmissions happen in overlapping time intervals. This method is employed in Time Division Multiple Access protocols and, in its non-deterministic form, in ALOHA or CSMA/CA type protocols which are very common in today's communication hardware (e.g. Ethernet, WiFi, Zigbee, etc.). In multi-hop networks, spatial and temporal separation are linked - less spatial overlap of commnication ranges means that transmissions can be packed more densely in time. It has been argued before that short range communication links improve performance and scalability of multi-hop networks [Schill (2007); Frater et al. (2006)].

Due to the severe bandwidth limitations and real time requirements in underwater swarms, Time Division Multiple Access (TDMA) is a good choice. TDMA scheduling algorithms are known in literature, which are mostly tailored for sensor networks or applications with sporadic communication [Gasieniec and Lingas (2002); Hakimi and Schmeichel (1993); Tavli and Heinzelman (2006); Xu (2003); Herman and Tixeuil (2004); Kulkarni and Arumugam (2004); Gronkvist (2006); Gaber and Mansour (2003)]. These algorithms often make many assumptions, such as that the nodes are arranged in a certain way, that the number of nodes is known in advance, or that the node position is known. These algorithms are also not optimised for continuous communication, and only consider sporadic broadcasts or convergecasts. Lastly, it is not assumed that the network topology changes rapidly.

The *Distributed Ad-hoc Omnicast Scheduling* TDMA algorithm (DAOS) was presented by Schill and Zimmer (2006) and Schill (2007). Transmissions take place in discrete time slots; nodes dynamically synchronise and compute local schedules which determine in which time slot they may transmit. The algorithm assumes a strongly connected network and that nodes can transmit all relevant information in a single time slot. Prior knowledge of the network topology is not required, and there is no central coordinator - all nodes behave identically. The algorithm can adapt to rapidly changing network topologies, reaches a very high channel utilisation and disperses information throughout the whole network very efficiently. It has been demonstrated that an all-to-all information exchange between n nodes takes generally less than n time slots. The algorithm is fully distributed and symmetric, and converges to dense, collision-free communication schedules. It is therefore suitable for communication channels where a receiver can decode a message if and only if exactly one transmitter within range sends. An example for such a channel is a pulse-modulated optical transceiver which does not distinguish pulse amplitudes. Theoretical and simulation results revealed that global information exchange is faster for networks with low density, i.e. where communication ranges are short compared to the network size, and nodes are only connected with their immediate neighbours.

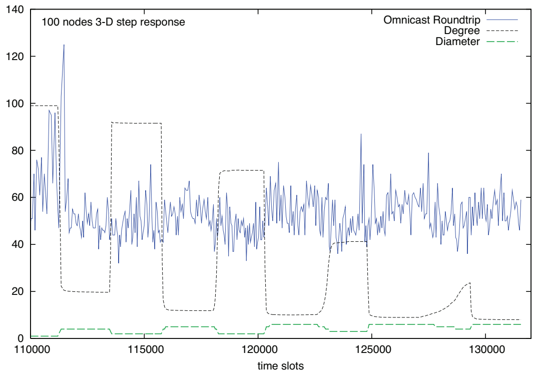


Fig. 2. Network performance of the PDAOS algorithm during rapidly changing network densities, in a swarm with 100 robots.

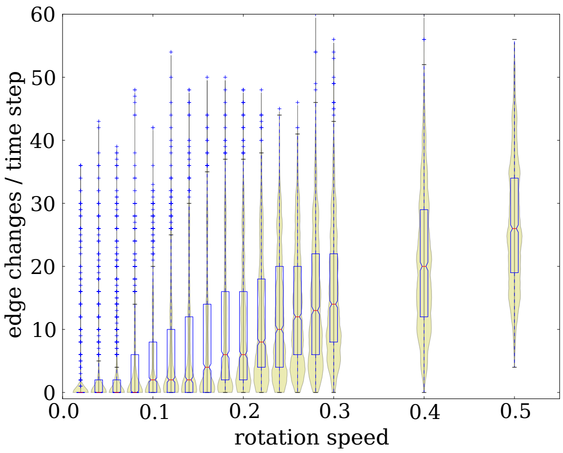
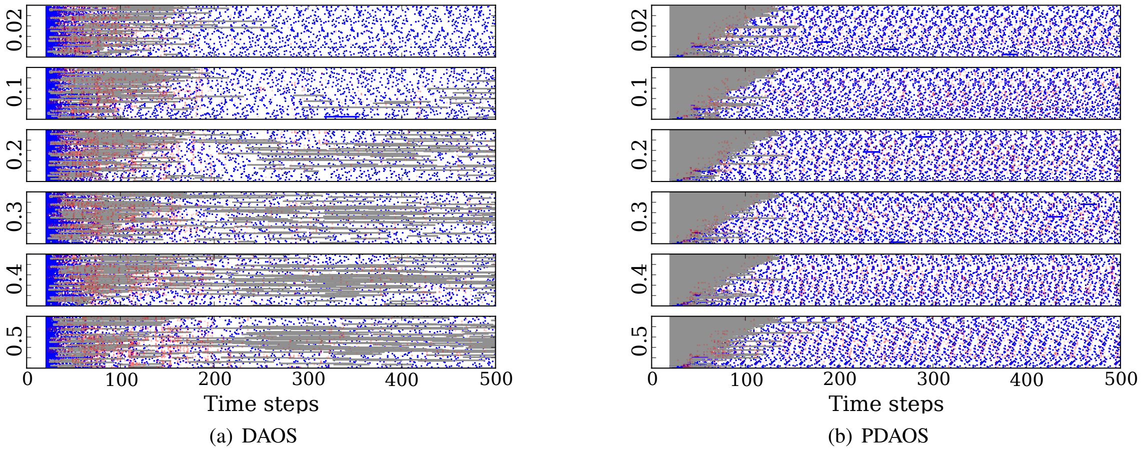


Fig. 3. Graph edge changes per time slot for varying rotation speeds. These values are calculated as the absolute sum of differences in the graph connectivity matrix for each time slot (standard box plot shown with violin plot indicating data distribution).

Experiments with custom-developed phase-modulated long-wave radio modules (shown in 1) revealed that the graph-theoretical network model commonly assumed in literature is too conservative [Schill and Zimmer (2006)]. In fact, if two or more transmitters send within the range of a receiver, the receiver will only observe a collision if the closest nodes have very similar incoming signal strength. Otherwise the receiver will reliably receive the message with the highest signal strength. This insight can be exploited to pack schedules more densely and achieve faster information dissemination in high-density networks. An improved algorithm, *Pruned Distributed Ad-hoc Omnicast Scheduling*, or PDAOS, was presented in



(a) DAOS

(b) PDAOS

Fig. 4. Network startup and resulting schedules. For each time slot, transmitting nodes are shown in blue. Nodes not in the schedule (waiting to be integrated) are shown grey. Collisions are marked in red. Schedules for different rotation speeds are shown (y-axis).

Schill and Zimmer (2007) and Schill (2007) This paper will investigate and compare the performance of these two algorithms when exposed to rapid dynamic reconfiguration of of the swarm. Experiments are carried out in a real-time simulation with an experimentally derived propagation and collision model.

3. EXPERIMENTS

Experiments were carried out in a multi-threaded real-time simulator, developed in Ada2005. Simulated robots can move in 2D and 3D based on predefined patterns, or based on decentralised swarming rules. Unless otherwise noted, predefined patterns were used in this paper. The network topology is derived continously based on robot positions and simulated signal strength; i.e. if a message is received is determined in real-time based on the collision model, node distances and message timings. Both collision models can be tested (i.e. (a) collision if two transmissions within range overlap, or (b) collision if signal strengths are similar). All tests with the DAOS algorithm were conducted with collision model (a), and for PDAOS with collision model (b). The communication range was set to 10 meters. The maximum schedule length for DAOS was set to 32 time slots - this limits the maximum density the algorithm is able to work with, but is a compromise to keep packet overheads low. For PDAOS the schedule length was set to 16 time slots, as the density limits to not apply as strictly here. PDAOS can virtually reduce the network density, which generally improves performance - a shorter schedule length is therefore beneficial.

Two experiment series were simulated. Firstly, static network performance was tested for a static network with varying numbers of nodes from 10 to 200, and different densities. Nodes are arranged in a 2D square grid with an inter-node grid distance of 2 m, 4 m, 6 m or 8 m. The following table shows the resulting network graph degree for 100 nodes:

Grid dist.(m)	avg. degree	min. degree	max. degree
2	43.1	17	68
4	15.8	5	20
6	6.8	2	8
8	3.6	1	4

In the second experiment, robots are arranged in 3D in two layers. The nodes in each layer are again arranged in a horizontal grid as before, and the two layers are separated vertically by half the grid distance. The layers are then shifted slowly along the horizontal axis in opposite directions. Robots reaching the grid boundary change layers and proceed in the opposite direction, so that the two layers are always aligned and connected. The result is a rotating, closed loop moving pattern. The “rotation” speed is given relative to the grid distance - i.e. a value of 1 means that nodes move at half the grid distance per communication time slot. This means that a nodes on top and bottom layer that were horizontally separated by one grid unit are aligned horizontally after one time step. At the start of each dynamic test, the robots are positioned beyond the communication range, after which they proceed to their respective grid positions within the first 10 seconds. The network therefore has to bootstrap in a dynamic environment were links are created and removed dynamically. Simulation runs were carried out for networks with 100 nodes, grid distance of 6 meters, and rotation speeds between 0.02 and 0.5. Due to the 3D arrangement, the graph degree is slightly different in the dynamic tests, with an average degree of 11.2, minimum of 4 and maximum of 17. Figure 3 shows the number of links that are continuously created and destroyed per time slot.

The network performance will be evaluated by determining the average time for global information exchange (omnicast roundtrip), and the average and minimum frequency of transmissions for each node. Initial startup of the network, and joining of two swarms, will be used as step-response-like scenarios. Dynamic performance is evaluated in a scenario where robots move through the swarm at various relative velocities; the average frequency at which robots move into and out of range of other nodes is used as a measure for network dynamics.

4. RESULTS

Network performance indicators were chosen based on relevance to swarm networks, such as the frequency at which nodes can transmit, and number of collisions. Of particular importance in swarm control is the speed of local and global information exchange between nodes. Local information distri-

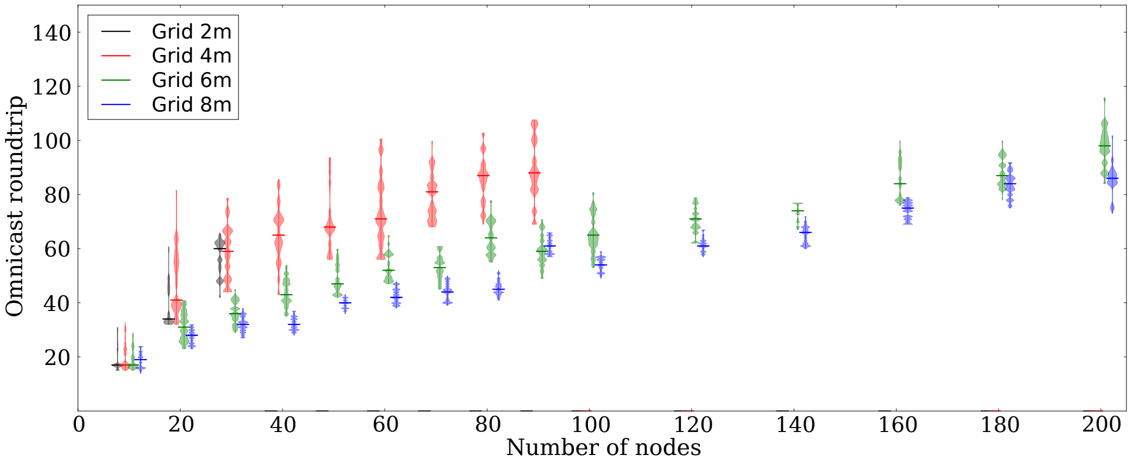


Fig. 5. Global information exchange for varying density and swarm size (DAOS, schedule length: 32 slots). The plot shows the distribution and median of the 0.25-0.75 quantile. All simulation runs had multiples of 10 as number of nodes - plot groups are slightly offset for visibility only.

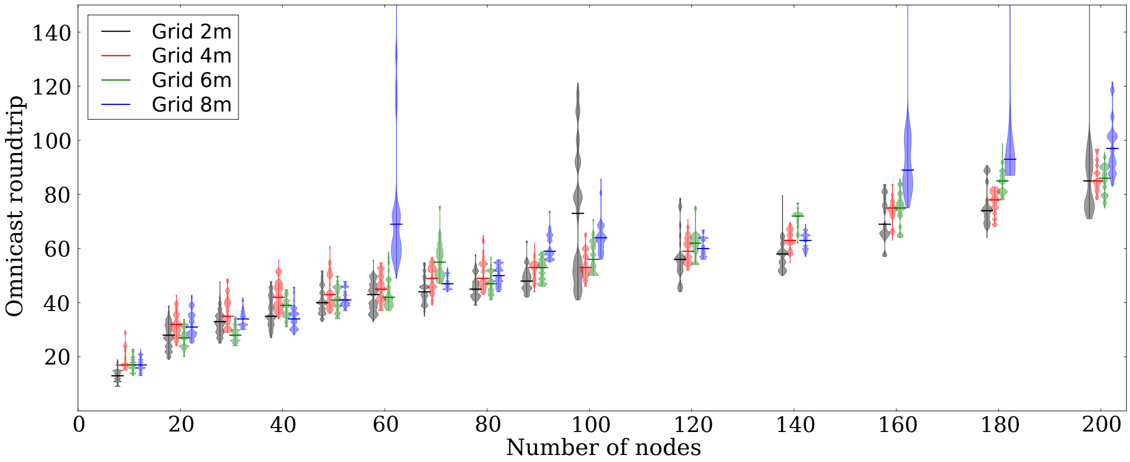
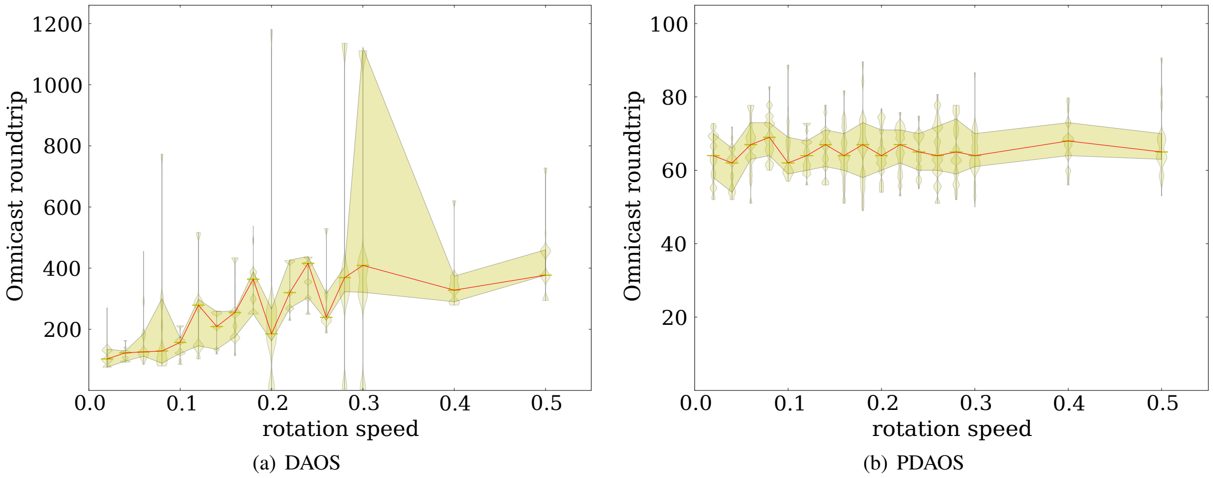


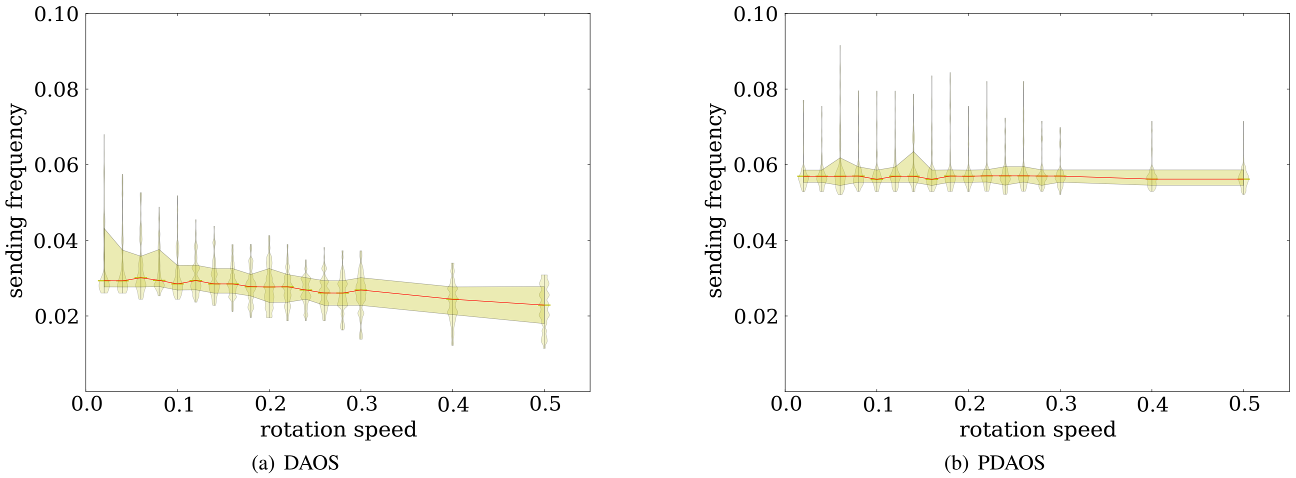
Fig. 6. Global information exchange for varying density and swarm size (PDAOS, schedule length: 16 slots). The plot shows the distribution and median of the 0.25-0.75 quantile. All simulation runs had multiples of 10 as number of nodes - plot groups are slightly offset for visibility only.



(a) DAOS

(b) PDAOS

Fig. 7. Omnicast performance in the dynamic tests for varying rotation speeds. Refer to fig. 3 for a measure of network reconfiguration for each speed. The PDAOS algorithm maintains a consistently good performance. Median performance shown in red with 0.25-0.75 quantile envelope. Actual distribution for each data set is shown as violin plots.



(a) DAOS

(b) PDAOS

Fig. 8. Node transmission frequency during dynamic tests for different speeds, in transmissions per time slot.

bution within a 1-hop neighbourhood is driven by frequency of sending events per node and collisions, and is important for low-level swarm control. Global information distribution is more difficult to measure (*omnicast*), but it greatly matters for higher-level tasks of the swarm, such as finding a consensus, computing a maximum, and calculating gradients. The method to measure the speed of global information exchange was described in detail in previous publications. In summary, it is the average time it takes for each node to receive information from all other nodes, and is measured through a distributed counting mechanism where each node keeps track of and distributes the counter value of all other nodes. A node can only increment its counter if all counter values have been distributed between all nodes. The time between counter increments is the performance value referred to in this paper as *omnicast roundtrip*, and is the time the network requires to exchange information from all to all nodes. As this is only possible if the network is connected and if all nodes are able to transmit, it is a good indicator of network functionality and performance. It has been shown before that global information exchange can be performed in less time slots than the number of nodes under certain conditions. The results in this paper further confirm this.

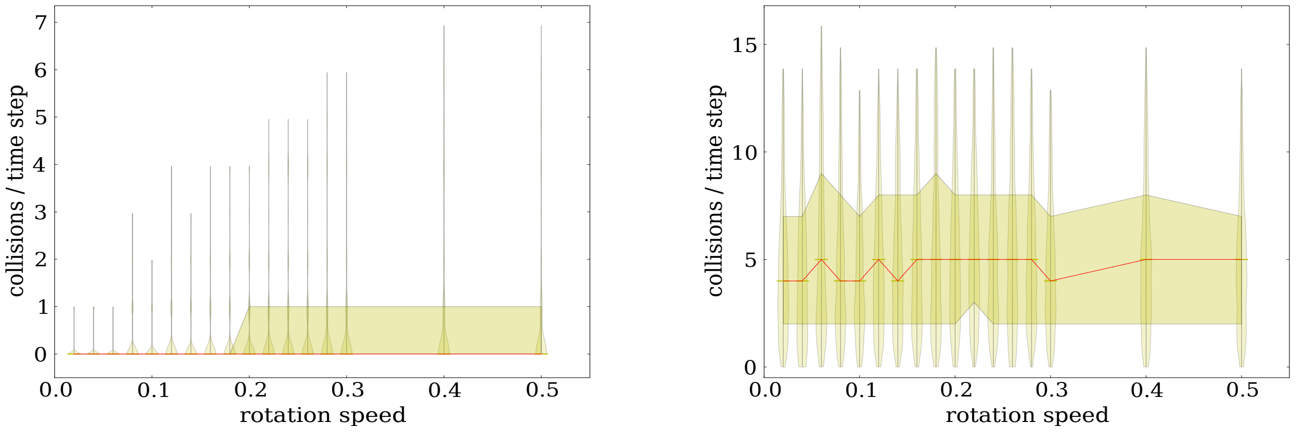
Figure 4 displays the startup behaviour and schedules generated by DAOS and PDAOS for selected rotation speeds. Both algorithms converge very quickly, in less than 200 time slots. DAOS converges to collision-free schedules in static networks, and shows very few collisions in slow-moving networks, but starts to drop nodes at higher speeds (visible as grey lines). While nodes are eventually integrated again, this reduces the network performance. PDAOS is able to reconfigure quickly without dropping nodes, even for high rotation speeds, and creates denser schedules due to pruning. The plots also show clearly the repetetive patterns of the communication schedule. Close inspection reveals how the schedules adapt to the changing network, visible as subtle changes in the patterns.

The results of the static tests are shown in figures 5 and 6. For low network densities (grid distances 6 m and 8 m) both algorithms show similar performance. For higher network densities, DAOS shows longer omnicast roundtrip times, and starts losing nodes in larger networks. This is because the 2-hop neighbourhoods become bigger than the maximum schedule length, making it impossible to accomodate all nodes. Omnicast cannot be performed for 4 m grid distance above 90 nodes, or 2 m grid distance above 30 nodes. This problem can easily be addressed by changing the schedule length to 64 or 128, however this greatly increases the overhead in each packet. PDAOS is unaffected by density changes, as each node prefers nodes with locally measured strong signals and removes nodes that are further away - hence the virtual network density is given by the schedule length, and fairly independent of actual density. The data for different network sizes indicates that omnicast can be achieved in less than n time steps for small networks, and $0.5 \cdot n$ time steps for n nodes for larger networks.

Previous experiments already established that PDAOS is able to adapt to rapid network changes while maintaining performance (figure 2 shows data from an experiment using 3D swarming rules and rapidly changing densities). However, in these previous experiments, due to the nature of the swarming rules, nodes mostly maintained their local neighbourhoods, requiring little reconfiguration for PDAOS, which prefers nodes ordered by signal strength. The dynamic experiments carried out for this paper therefore shift two layers of nodes against each other, which means that half of neighboured nodes change repeatedly, forcing some level of reconfiguration. While DAOS is clearly affected by this, PDAOS performs consistently.

The same result applies to the average sending frequency of nodes (figure 8, which for DAOS reduces from 0.03 to 0.02 for larger speeds, while PDAOS maintains a mostly constant frequency of almost 0.06 transmissions per time slot (approximately once every 16 time slots on average).

Figure 9 shows the number of collisions in the network. For DAOS the collision model is stricter (any two overlapping transmissions within range of a receiver are counted). For PDAOS, a collision is only counted if the signal strengths of two simultaneous receptions are within +/- 5% of each other. Despite the more relaxed collision model, PDAOS shows a significantly higher number of collisions of approximately 5 collisions per time step, compared to a median of 0 for DAOS. Yet this does not seem to affect information dissemination, as there are still enough nodes that do receive the messages, and information flow through the network is maintained.



(a) DAOS

(b) PDAOS

Fig. 9. Number of collisions per time slot (median and distribution)

5. CONCLUSIONS

This paper discusses the performance of two swarm-optimised time division multiple access scheduling algorithms, which were specifically designed for large-scale underwater swarms and have been implemented on a compact underwater longwave radio module. In particular, the information dissemination performance of the network under various dynamically changing network topologies was examined, to obtain how fast the swarm can reconfigure while maintaining good connectivity. Both in static and dynamic tests, PDAOS shows better and more consistent performance than DAOS - it appears to be mostly unaffected by dynamic effects. The better performance can be explained by the reduced schedule length achieved by pruning according to signal strength. A hypothesis for the robustness of PDAOS towards dynamic changes is that the restructuring of schedules based on signal strength allows it to maintain schedules without dropping nodes and letting them reapply. If the underlying communication hardware permits it, PDAOS is therefore a better choice than DAOS, despite the higher number of collisions it produces. While DAOS has been tested on real longwave radio modules, PDAOS still needs to be implemented and tested on hardware in future work. In this simulation it was assumed that signal strength is known accurately - this may not always be the case in the real world. The impact of noise in signal strength measurements still needs to be evaluated. The assumed collision model for PDAOS has been experimentally established for the PLL-based phase-modulated longwave receivers built within this project. Other communciation channels such as pulse-modulated optical communication do not show the behaviour of locking onto the stronger signal, and will mix strong and weak signals. PDAOS is therefore not expected to work for these channels, which makes DAOS the more appropriate choice. In a practical application, PDAOS with a schedule length of 16 would incur an overhead of 20 bytes per packet for node ID, logical clock and the schedule. For 80 bytes of user data, 10 packets per second can be achieved using an 8192 bit/second longwave radio. Based on simulation results, nodes would therefore be able to send at least once every 17 time steps, or once every 1.7 seconds, allowing for responsive swarm control locally. In a network with 100 nodes, global information exchange can be achieved approximately every 5-6 seconds. This could e.g. allow a large swarm to identify and track a plume of pollutants in real-time.

REFERENCES

Frater, M.R., Ryan, M.J., and Dunbar, R.M. (2006). Electromagnetic communications within swarms of autonomous underwater vehicles. In *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, 64–70. ACM Press, New York, NY, USA. doi: http://doi.acm.org/10.1145/1161039.1161053.

Gaber, I. and Mansour, Y. (2003). Centralized broadcast in multihop radio networks. *Journal of Algorithms*, 46(1), 1–20. doi:http://dx.doi.org/10.1016/S0196-6774(02)00292-4.

Gasieniec, L. and Lingas, A. (2002). On adaptive deterministic gossiping in ad hoc radio networks. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, 689–690. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Gronkvist, J. (2006). Novel assignment strategies for spatial reuse TDMA in wireless ad hoc networks. *Wireless Networks*, 12, 255–265.

Hakimi, S.L. and Schmeichel, E.F. (1993). Gossiping in radio networks. *Ars Combinatoria*, 35-A, 155–160.

Herman, T. and Tixeuil, S. (2004). A Distributed TDMA Slot Assignment Algorithm for Wireless Sensor Networks. In *ALGOSENSORS*, 45–58.

Kulkarni, S.S. and Arumugam, U. (2004). Tdma service for sensor networks. In *ICDCS Workshops*, 604–609.

Schill, F. (2007). *Distributed Communication in Swarms of Autonomous Underwater Vehicles*. Ph.D. thesis, Australian National University.

Schill, F., Trumpf, J., and Zimmer, U.R. (2005). Towards optimal TDMA scheduling for robotic swarm communication. In *Proceedings Towards Autonomous Robotic Systems*.

Schill, F. and Zimmer, U.R. (2006). Effective communication in schools of submersibles. In *Proceedings IEEE OCEANS'06*.

Schill, F. and Zimmer, U.R. (2007). Pruning local schedules for efficient swarm communication. In *Proceedings of the International Symposium on Underwater Technology, Tokyo, Japan*.

Tavli, B. and Heinzelman, W.B. (2006). Energy and spatial reuse efficient network-wide real-time data broadcasting in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(10), 1297–1312. doi: http://dx.doi.org/10.1109/TMC.2006.151.

Xu, Y. (2003). An o(n1.5) deterministic gossiping algorithm for radio networks. *Algorithmica*, 36, 93–96.