TDMA-based exchange policies for multi-robot communication of world information ⋆

**F. Maurelli * Z. Saigol * J. Cartwright * D.M. Lane *
A. Bourque ** B. Nguyen ****

* *Ocean Systems Lab, Heriot-Watt University, EH14 4AS Edinburgh - Scotland, UK (e-mail: f.maurelli;d.m.lane@hw.ac.uk).*
** *Defence R&D Canada Centre for Operational Research and Analysis, 101 Colonel By Drive, 6 CBS, Ottawa, Ontario, Canada, K1A 0K2 (e-mail: alex.bourque;bao.nguyen@drdc-rddc.gc.ca)*

Abstract: The aim of this paper is to present a distributed world model system for multi-robot cooperation. Information is stored in semantic form in an ontology, allowing unlimited new concepts to be added. Automatic exchange of information is driven by the prioritised information needs of each vehicle, and is robust in the face of high latency, low bandwidth, lossy acoustic communications. The performances of different TDMA-based exchange policies were compared to a standard acknowledgement policy, through tests in simulation. Under certain conditions, particularly high packet loss, the matrix-based acknowledgement policies introduced here substantially reduced time to reliable exchange; the results suggest an adaptive exchange policy may perform best under varying conditions.

1. INTRODUCTION

The coming generation of Autonomous Underwater Vehicle (AUV) missions are requiring that vehicles operate without human assistance for extended periods of time (days), in environments that are imprecisely known. Examples of such persistent autonomy needs are present in over the horizon surveillance or mapping (for security and marine science), and in deep-water oilfields (for inspection, repair and maintenance). To accelerate progress of these missions, multiple vehicles may be expected to operate simultaneously, exhibiting collaboration in their execution of tasks.

Equipping vehicles to execute these missions successfully requires that they are able to represent crucial information of the environment or world that can be used as a basis for selecting future actions. The world model typically includes not only the physical state of objects in the environment (including collaborating platforms), but also a degree of internal states, such as the health and intentions of other platforms.

Communication is therefore vital, in order to share information and to allow an effective cooperation on-the-fly, with a mission that can be adapted according the information received by the other robots. Two existing languages for information exchange with AUVs are the Compact Control Language from WHOI (Stokey et al. [2005]), and the Common Control Language from AUSI (Duarte et al. [2005]). These provide good support for the compact encoding of core information relating to AUVs, but application-level exchange must use opaque binary payloads. The DELPHIS system (Sotzing and Lane [2010]) uses broadcast exchange of status messages for AUVs to share mission state and targets, but this relies on constant repetition of a fixed message, rather than acknowledged delivery of arbitrary content.

This paper presents the topic of a decentralised world model service that operates across multiple underwater vehicles, sitting above the communications data link layer. Figure 1 illustrates how the world model fits in a multi-vehicle system. It presents an abstraction away from data packets and acknowledgements, providing each vehicle with a partially shared world view at the semantic information level. This removes the need for vehicle control systems to explicitly plan speech acts, whilst giving the world model the freedom to optimise information exchange across the group of vehicles. The decentralised nature of the system ensures that it is robust to temporary node isolation. We target the most capable underwater communications mechanism, acoustic modems. This paper is organised as follow: Section 2 presents the challenges of

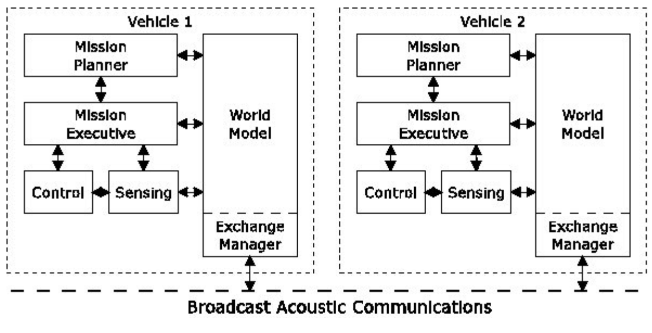

Fig. 1. The world model in a simple three-layer robot control architecture

⋆ This work was supported in part by the Defence R&D Canada Ottawa, under PWGSC Contract #W7714-115079/001/SV

acoustic communication and distributed world modeling; Section 3 briefly presents the main blocks that forms the proposed system; Section 4 focuses on the acknowledgement policies; Section 5 presents the results in simulation. Finally, Conclusions and Future Work are outlined.

2. CHALLENGES

The aim of this project is to design and implement a framework for a distributed semantic world model to support cooperation between multiple autonomous vehicles. The system should have the properties given below.

(1) Temporal context is incorporated into the world model (lacking in existing systems).
(2) Vehicles may exchange information with their peers at different levels of abstraction, significantly increasing on the capabilities of current systems.
(3) Selection of information to exchange is performed automatically, with minimal prior input from the vehicle subsystem designers.
(4) Robust to communication failure between individual nodes.
(5) No central communications point.
(6) Vehicles may still operate in isolation, with reference to their local world model.
(7) New vehicles and missions can be incorporated in a simple, modular fashion.

The combination of temporal ontological representation (1) with robustness to communication failure (4,5,6) will be the main novel contribution to the field.

Evaluation of the framework will necessarily involve creating several ontologies to describe the information representing the vehicles under test and the world they inhabit.

Upon creating a list of goals or requirements for a system, it is equally important to list issues that will *not* be addressed, and any assumptions that will be made. Thus, for this project:

(1) Communication will be point to point or broadcast, with no facility to route information to hidden nodes.
(2) The cooperating vehicles share a common purpose, without conflicting motivations.
(3) Existing subsystems from the Ocean Systems Laboratory will be integrated to provide robot control, sensing and planning functionality.

3. PROPOSED SYSTEM

The proposed system, outlined in Fig. 2, is composed by the following modules:

- **World Modeling**
  Ontologies allow richer data structure than databases, with the description of hierarchies of classes that can posses both data values, and relations to other classes. Thus they are the obvious choice for providing structure in a complex world model (Guarino [1998]). Unlike standard ontologies, we plan to address temporal context and uncertainty linked to the world modelling and the ontology construction.
- **Communication System**
  A key part of the architecture is the Data Exchange Manager, which will orchestrate the exchange of information between the local world model for this vehicle, and those of the other vehicles in the collective. Tasks performed by the Data Exchange Manager include the selection of information to send to other vehicles based on utility and communications availability, and the merging of received data into the local world model whilst maintaining consistency.

4. WORLD MODELING AND ONTOLOGIES

Several ways have been explored in order to represent world data in a connected system, which is able to express relations among different entities. Recently, there has been raised interest in using semantic frameworks, which can provide a hierarchical distributed representation of knowledge for multidisciplinary agent interaction (Miguelanez et al. [2011]). They provide a common machine understanding representation of knowledge between embedded agents that is generic and extendible. They also include a reasoning interface for inferring new knowledge from the observed data and knowledge stability by checking for inconsistencies. These frameworks improve local (machine level) and global (system level) situation awareness and context for mission and trajectory behaviour. They can therefore act as enablers for autonomy and on-board decision-making. There are currently several institutions developing standards for knowledge representation under these frameworks. Particular attention is taking the effort describing the concepts and relationships for the domains of unmanned platforms (Miguelanez et al. [2011]) and the underwater environment (Foundation [2004-2012]).

Ontologies are a way of representing data in a structured manner, such that it is easy for both humans and machines to interpret the data contained in them. Conventionally the contents of an ontology are divided into the `TBox` (terminological box) and `ABox` (assertion box), where the

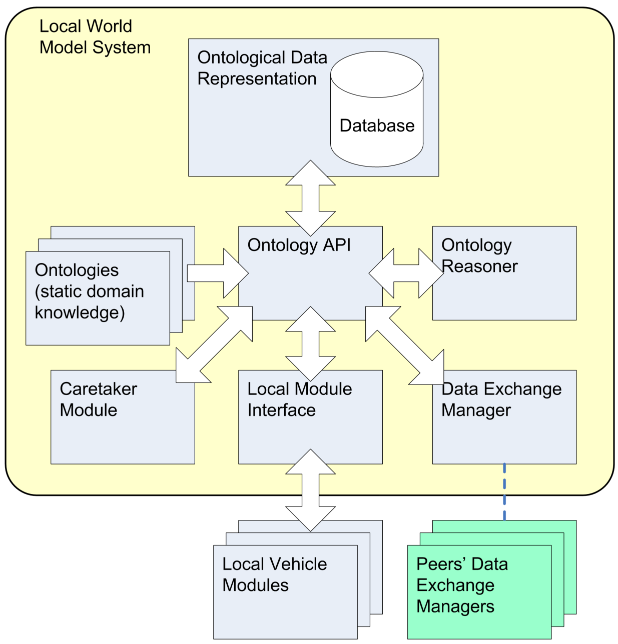

Fig. 2. Proposed world model architecture

TBox is equivalent to a class hierarchy in object-oriented programming, and the ABox would be a container for all instances of the classes.

The world modelling system proposed can be used with any TBox ontology, provided the same ontology is supplied to all platforms running the system. To enable efficient transmission of data, a unique but short UID will be created for every class in the TBox, and these UIDs used to instantiate or refer to instances of the classes.

While the TBox is fixed, the ABox can be supplemented at run-time by recording new observations or objects, and these will be automatically shared between all nodes in the system. Observations should be formatted according to a supplemental TBox which defines a compact way of storing data about objects that may change over time. This TBox is based on situation awareness ontology work by Matheus et al. [2003], and defines and Attribute class that can be associated with an object of any class. An Attribute can represent any property, for example "name" or "longitude". Rather than storing a single value for an Attribute, we allow an Attribute to have many PropertyValues, where a PropertyValue stores both a value and a timestamp (which is taken to be the time the value was observed at). A PropertyValue can store either a data value (such as an integer or string) or a link to another object in the ABox.

There are widely available and proven tools for building ontologies in languages such as OWL (McGuinness and Harmelen [2004]). Ontology reasoning engines are able to perform logical consistency checking of ontologies, and infer new information based on ontology axioms and rules. They can be key for multi-AUV communication, as the different vehicles exchange values with references to ontological concepts.

5. ACKNOWLEDGEMENT POLICIES

Acknowledgements are naturally performed at the packet level; packets are either fully received or not received at all. Information contained within a successfully received packet is always added to the local world model, or skipped if it is already present. Therefore, if a packet sent by node A is acknowledged by node B, another node receiving that ACK (either A or C) will know that the ACKing peer now holds the information in the ACKed packet. To make use of this, each node in the system maintains a set of instance UIDs known by each peer.

Three packet acknowledgement policies of increasing complexity are considered below; but first, the problem definition.

5.1 Problem Definition

Consider an N node underwater wireless (acoustic) network with nodes indexed by $j \in \mathbb{Z}^+,\ j \leq N$. Let the probability of packet loss on a link from i to j be p_{ij}. We will treat the acoustic modem as a black box, and assume that any corruption of a received packet will lead to the loss of the entire packet, thus the packet error rate PER is equal to p_{ij}. The acoustic medium is broadcast in nature, and we assume the use of single channel acoustic modems that permit only one node to transmit at once in order for there to be no collisions.

5.2 Policy: Standard ACK

This acknowledgement policy is a straightforward packet ACK applied to the round-robin TDMA schedule. An $(N-1)$ bit vector is included with every transmitted packet, acknowledging any messages received within the last $(N-1)$ TDMA slots. The probability of a successfully acknowledged delivery in the first TDMA slot from node i to node j is $p_{ij}.p_{ji}$. If we consider exchange of a single item of information, and state that automatic retransmissions are performed in the sender's next slot until an acknowledgement is received, the expected number of frames required for delivery and acknowledged delivery follow simple geometric distributions. Where T denotes the number of TDMA frames required:

$$E(T_{delivered}) = \frac{1}{1-p_{ij}}, \quad E(T_{acked}) = \frac{1}{(1-p_{ij})(1-p_{ji})}$$

Now consider the scenario where a single node has the same item of information to send to the $(N-1)$ other peers in the network. If we assume that the random packet corruption is independent for each receiver, then the expected number of frames until delivery and acknowledged delivery are the same for any N. Of course, while the number of TDMA frames remains the same, the actual time occupied by each frame increases linearly with the number of slots N when using a fixed slot period.

Where recipients are instructed to retransmit information to other nodes, but only until they believe all interested nodes have taken delivery, an analytical solution is less forthcoming. The solution becomes more complex still when we consider more advanced acknowledgement policies, and such derivations are considered to be outside of the scope of this paper. We thus employ repeated randomised tests to evaluate the policies.

5.3 Policy: M-ACK

Named M-ACK for 'matrix acknowledgement', this policy involves each peer transmitting a matrix A_{ij} of $N \times N$ ACK bits with every packet; in practice this can actually be reduced to $N(N-1)$ bits by eliminating meaningless self-ACKs (the matrix diagonal). The ACK matrix states that node i received the message sent by j before the transmission from i; the ACKs are relative to the transmission slot of the peer that is said to have received the packet. By using an ACK matrix, nodes are able to repeat ACK bits sent by other nodes, increasing the probability that ACKs for a transmission are seen by the source. To implement this scheme, each node keeps a record of the $(N-1)$ ACK matrices received from its peers since it last transmitted, as well as the information content of the last two packets transmitted by each peer.

Preparation: Just before the beginning of every TDMA slot k, each peer clears the bits in the ACK matrices it holds where the ACKed peer $j = k$, as well as the entire ACK matrix last received from peer k. Clearing old ACKs is necessary to prevent them from propagating past the point in time at which they are valid.

Transmitting: When node s comes to transmit, it constructs its own ACK matrix by performing a boolean OR operation over the $(N-1)$ peer ACK matrices it is holding, then replaces the row $i = s$ with its own ACKs for the packets it received directly.

Receiving: On receipt of an ACK matrix, the ACK bits are associated with either the last or second-last packet received from a peer, depending on the relative TDMA slot positions of the ACKing node i and the ACKed node j.

The M-ACK policy has the potential to reduce the amount of redundant information retransmission across the network, by increasing the probability of ACKs propagating back to the information sender. The bandwidth cost of the matrix used by this policy is $\mathcal{O}(N^2)$, whereas a simple ACK vector is only $\mathcal{O}(N)$. The additional per-packet bandwidth cost of M-ACK versus using a standard vector of ACKs is $(N^2 - 2N)$ bits, assuming the matrix diagonal elements are omitted.

5.4 Policy: MPS-ACK

Named MPS-ACK for 'matrix pseudo acknowledgement', this policy is based on M-ACK. However, it adds what we have termed a 'pseudo ACK'. Suppose that peer j transmits a packet and it is received by peer k. The packet may or may not have been received by peer i, but no acknowledgement to this effect is received by peer k. Under this policy, peer k is able to consult its internal state to determine if peer i already has all the information contained in the packet. If this is the case, peer k is permitted to set the acknowledgement bit A_{ij} – a pseudo ACK – in its next transmission.

This mechanism has the potential to reduce redundant retransmission of information in the network when an intended recipient of a packet has previously received the information, but due to packet loss the genuine acknowledgement would not seen by the source. Note that MPS-ACK requires that the acknowledgement system is aware of the information content of packets and existing peer knowledge, and would not be possible if packets were considered as opaque items to be acknowledged. There is no bandwidth penalty for implementing this policy, as the MPS-ACK matrix is the same size as that for M-ACK. It will incur an $\mathcal{O}(N^2M)$ computation penalty for an N node network, with an average packet capacity of M information instances.

6. RESULTS IN SIMULATION

Let the term 'exchange policy' describe the combination of acknowledgement policy with the transmission candidate selection scheme. All exchange policies tested here use the transmission selection scheme as a base; some variants allow repeating of information generated by another node. As an initial test of the performance of the different policies, a stepwise iterative simulation was constructed to evaluate the distribution of a single information item from one to $(N-1)$ other nodes. Boolean variables were used to represent the presence and transfer of the information item, with no actual data transmitted or received. In order to compare the different exchange policies as closely as possible, the same sequence of random packet corruption was applied to all policies, for a given test run; 2000 test runs were performed for each (PER, policy) data point.

Note that this simple test scenario describes an unequal distribution of observations, where only one node is generating. Later tests cover an equal distribution of observations. In the discussions below, we focus on the performance in terms of time to acknowledgement of delivery, rather than time to delivery itself. The reasoning being that until delivery is acknowledged, the information will be retransmitted, wasting bandwidth and slowing delivery of all information on average.

Figures 3 and 4 show results of a five node simulation with one node distributing a single observation. The trends in figure 3 show that while having other nodes repeat the information is beneficial, the extra information provided by the matrix ACKs has a substantial positive effect, even without repeating. At high PER, MPS-ACK slightly outperforms standard ACK-repeat on that metric. However, this plot doesn't show the whole picture, as even after the source is aware of successful delivery to all nodes, repeating nodes may continue to unnecessarily transmit the information.

For the all-node view, we look to figure 4, which presents the time until all senders are aware of successful delivery. Interestingly, with this metric the standard ACK-repeat policy performs worse than the standard ACK policy; it appears that the benefit of repeating information is offset by the requirement for the additional senders to successfully receive acknowledgements. Of course, purely

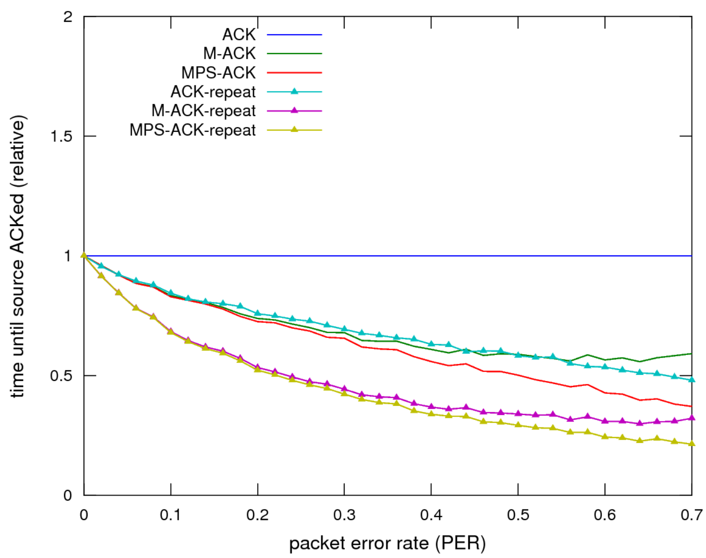

Fig. 3. Results in simulation with $N = 5$ and one information item distributed by node 1. Graph shows mean time until information source has confirmed delivery to all nodes, relative to baseline ACK policy (smaller is better). M-ACK-repeat and MPS-ACK-repeat consistently beat the other policies here, but at higher PER, MPS-ACK is not far behind. Baseline ACK policy performs worst. Note that this metric doesn't take account of time required for other nodes to have confirmed delivery, if repeating.

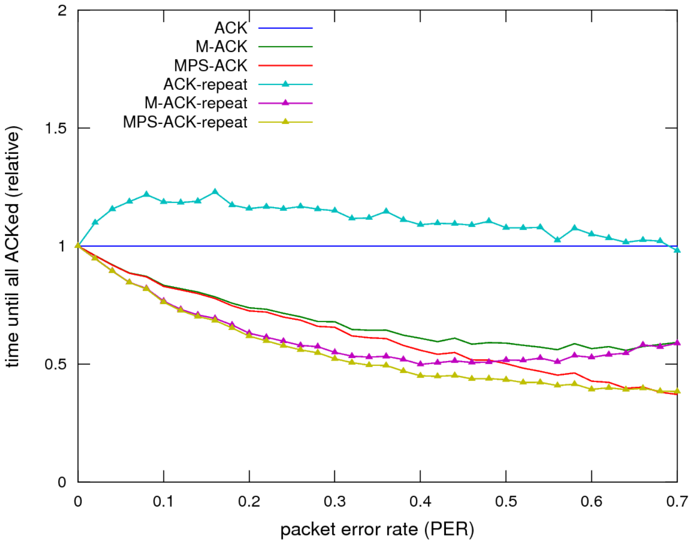

Fig. 4. Results in simulation with $N = 5$ and one information item distributed by node 1. The graph shows mean time until all senders (1 or 5, depending on the chosen policy) have confirmed delivery to all other nodes, relative to baseline ACK policy. Naive repeating with ACK-repeat generally performs worse than ACK. Overall, this figure suggests that at lower PER, the best results are achieved by using repeating with either matrix-based policy. However, at higher PER, the benefits of the MPS-ACK policy dominate.

in terms of information delivery, having other nodes repeat the information will increase overall distribution speed, if their transmission slots are otherwise free and there is non-zero packet loss.

Comparing the five node results from figure 4 with ten node results in figure 5, we see similar trends as expected. For brevity, a plot of results for the three node case is not included here. It is very similar to the other two cases, but with the crossing of plots for MPS-ACK and M-ACK-repeat occurring at a much lower PER. From these results, the strongest performing policy so far appears to be MPS-ACK-repeat.

7. CONCLUSIONS AND FUTURE WORK

This paper has presented a distributed world model system for multi-AUV cooperation, with an analysis in simulation of different TDMA-based exchange policies. The matrix-based acknowledgement policies introduced substantially reduced time to reliable exchange, particularly under high packet loss. The results suggest an adaptive exchange policy may perform best under varying conditions. Future work will address a more realistic information generation scenario. Integration with ROS (Quigley et al. [2009]) will be key to run the code in the vehicles. The next step will be the use of acoustic modems to test the core principles of the work and soon after that, field trials are foreseen in Scotland, with the AUV fleet of the Ocean Systems Lab.

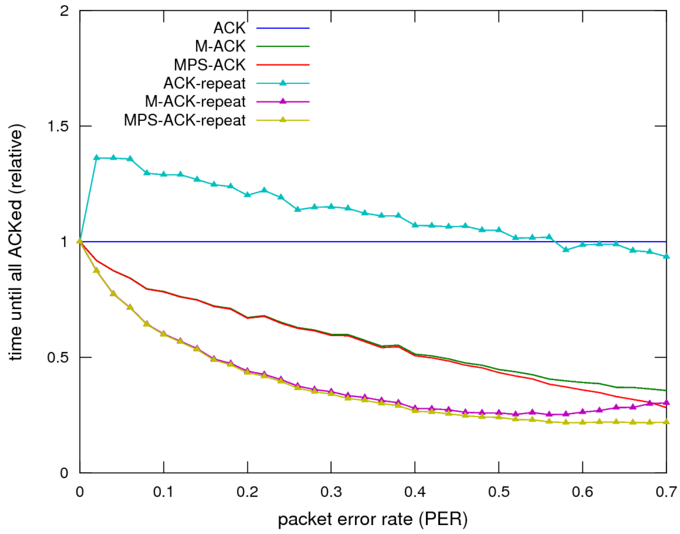

Fig. 5. Results in simulation with $N = 10$ and one information item to be distributed by node 1. Graph shows mean time until all senders (1 or 10, depending on the chosen policy) are aware of delivery to all other nodes, relative to baseline ACK policy. Compared to the $N = 5$ test case in figure 4, here there is a slightly more pronounced difference between policies with and without repeating.

REFERENCES

C.N. Duarte, G.R. Martel, C. Buzzell, D. Crimmins, R. Komerska, S. Mupparapu, S. Chappell, D.R. Blidberg, and R. Nitzel. A Common Control Language to Support Multiple Cooperating AUVs. In *Proceedings of the Fourteenth International Symposium on Unmanned Untethered Submersible Technology*, 2005.

National Science Foundation. Marine metadata interoperability. In *www.marinemetadata.org*, 2004-2012.

N. Guarino. Formal ontology in information systems. In *Proceedings of Formal Ontology in Information Systems*. IOS Press, 1998.

Christopher J. Matheus, Mieczyslaw M. Kokar, and Kenneth Baclawski. A Core Ontology for Situation Awareness. In *Proceedings of the Sixth International Conference on Information Fusion*, pages 545–552, 2003.

DL McGuinness and F Van Harmelen. Owl web ontology language overview. *W3C recommendation*, 2004.

Emilio Miguelanez, Pedro Patron, Keith E. Brown, Yvan R. Petillot, and David M. Lane. Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles. *IEEE Transactions on Knowledge and Data Engineering*, 23: 759–773, 2011. ISSN 1041-4347.

M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. 2009.

Christopher C. Sotzing and David M. Lane. Improving the coordination efficiency of limited-communication multiautonomus underwater vehicle operations using a multiagent architecture. *Journal of Field Robotics*, 27 (4):412–429, 2010. doi: 10.1002/rob.20340.

RP Stokey, LE Freitag, MD Grund, and W.H.O. Instn. A Compact Control Language for AUV acoustic communication. *Oceans 2005-Europe*, 2, 2005.