A Framework for Marine Sensor Network & Autonomous Vehicle Interaction

Matthew Dunbabin
Autonomous Systems Laboratory
CSIRO ICT Centre
PO Box 883, Kenmore QLD 4069, Australia
Email: matthew.dunbabin@csiro.au

Peter Corke
School of Engineering Systems
Queensland University of Technology
Brisbane, Australia 4001
Email: peter.corke@qut.edu.au

Abstract— **We wish to exploit the complementary characteristics of marine sensor networks and autonomous vehicles through interaction and cooperation to achieve tasks that neither can accomplish alone – large-scale, long duration perception of the environment and response to changes in that environment. We outline and experimentally evaluate a new programming and communications framework that supports the integration of autonomous surface and underwater vehicles and sensor networks. The proposed programming framework is RPC-based and allows robots and static sensors to consider each other as service providers and consumers. The framework allows service forwarding in which services that cannot be provided by one fixed or mobile node are forwarded to another over a local communications link. This allows the functionality of the system to be extended in a way that is transparent to the sensor network.**

I. Introduction

Many scientifically relevant marine and inland water environments are too large and dynamic for efficient and cost effective data collection by robotic systems alone. Autonomous Surface Vehicles (ASVs) and Autonomous Underwater Vehicles (AUVs) offer continuous data collection capabilities at a fine spatial scale (along transects). Wireless surface and underwater sensor networks allow larger scale data collection at coarse spatial resolution but high temporal resolution. To date these have typically been considered as completely separate systems with separate data streams that are often post processed at a centralised facility. A method for improving capture of relevant dynamic properties in these environments is to integrate the technologies in a way that they complement each other in real-time and at distributed locations.

The field of robot and sensor network integration has been approached by researchers from two directions. One is the addition of mobility to sensor nodes, creating mobile sensors which are essentially robots based on sensor network technology [1], [2]. The second, is adding ad-hoc networking to robotics, for example [3] which uses robot motion as a primitive to maintain network connectivity, and [4] which evaluates different routing protocols for robots. In this class the robots are typically based on high capability processors, with UDP over WiFi networking [5], high power consumption and relatively short lifetime between charging. In all these examples, the integration is not tight, that is, the sensor node is merely a passive passenger on the robot, or the robot is simply logging messages that were picked up by the sensor node it was carrying.

There are currently few long-term deployed marine sensor networks with examples including the Swedish “Klimat” network [6], the “ReefGrid” network on the Great Barrier Reef [7] and the “OASIS” and “LOBO” networks [8] operated by the Monterey Bay Aquarium Research Institute, and the CSIRO “TasMAN” [9]. There are also a small number of sensor networks utilizing underwater acoustic communications (for example, NetALS and SeaWeb [10]). However, to the author’s knowledge these systems are not currently being used to directly integrate with robotic systems in real-time.

Pereira et.al [11] describe a communication framework particularly suited to Autonomous Underwater Gliders using radio modems instead of the usual Iridium phone to increase data throughput and reduce surface time and data transmission cost in coastal environments. Preliminary results show the system is applicable in regions where there is receiver/base station coverage. However the data is centrally aggregrated and there is no demonstration of re-tasking over the communications link. Sukhatme et. al. [12] describe a 10 node sensor network with a small autonomous boat to measure plankton assemblages in a small lake. The system demonstrates distributed mapping and the authors mention that the boat path was guided by information from the sensor network however no details are given on the communication and tasking frameworks.

In this paper we present and experimentally evaluate a programming framework that allows a robot to become a member of a wireless sensor network. It bridges the two technologies by integrating autonomous control of robotic vehicles with multi-hop message routing in ad-hoc networks. It allows a sensor network to be used by robotics engineers without needing to know arcane details of sensor networks such as serial forwarders, MAC or routing protocols. Engineers are able to easily query and change the state of the robot or a remote sensor node, and add new services to the robot without any change to the sensor network software.

The remainder of this paper is structured as follows: Section II outlines the RPC-based programming framework for integrating autonomous vehicles and sensor network nodes. Section III presents experimental results using the framework applied to an Autonomous Surface Vehicle on a large

978-1-4244-5222-4/10/$26.00 ©2010 CROWN

water quality monitoring sensor network. A discussion of the increased functionality that integrated robot and sensor network interaction can have on operation and science in remote environments is given in Section IV, with Section V concluding the paper.

II. Integrated Robot & Sensor Network Framework

In order to allow marine robotic systems and sensor networks to integrate and cooperate at levels that improve overall system functionality and scientific discovery, a common yet flexible communication framework is required. This framework must allow interaction between the mobile and static entities, but also cater for their individual sensing, processing and mobility capabilities. Furthermore, as these network systems become very large (both in terms of coverage area and number of nodes) the ease with which we can create, program, interact with and manage the entire network becomes paramount in the framework design.

A. Integration Requirements

Sensor networks and robotic systems typically have disparate programming frameworks making direct integration and sharing of information and tasks difficult. Therefore, an interactive network must support communications between any, or all of the sensors and robots, as well as human operators.

Marine robotic systems are typically based around one or more Linux class computers with WiFi communications whilst at the surface and acoustic communications subsurface. These systems are typically less energy constrained compared to a sensor node due to their relatively short operation time before recharging. Sensor networks on the other hand typically operate on small low-power microcontrollers to optimise the use of their limited onboard energy. Consequently, robots and sensor networks operate under different software development environments, e.g. Linux and a socket communication interface for robots, and TinyOS and Collection Tree Protocol (CTP) [13] for sensor networks.

Effective large-scale systems require a solution to bridge these two disparate software environments. Figure 1 illustrates the system considered in this paper where a large, distributed wireless sensor network interacts with autonomous vehicles operating within the bounds of the network. Access to the hardware by humans can be achieved through a remote internet connected host.

B. Functionality Requirements

In order to increase the operational and scientific capabilities of the integrated robotic and sensor network assets, it is required that the system has the following functionalities:

- Stream messages from any sensor node or robot to the host; e.g. a means for static sensor nodes to report environmental measurements, or a robot to report its location.
- Request some particular state from a sensor node or robot.

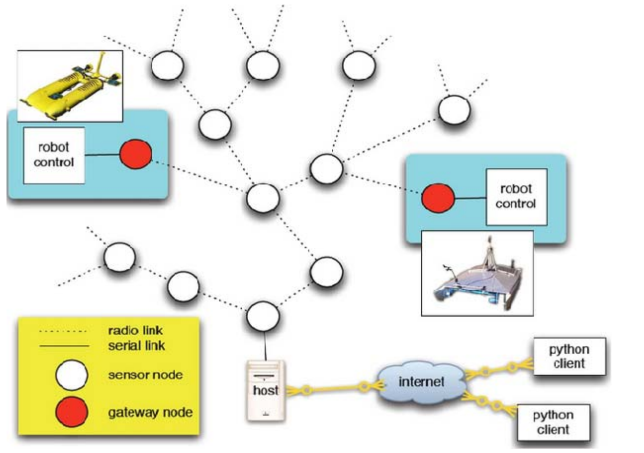

Fig. 1. Schematic representation of the sensor network, autonomous vehicles (robots) and remote internet connected host.

- Set some particular state on a sensor node or robot.

It is possible to achieve the first functionality using well known techniques such as CTP. In order to achieve the second and third functionalities, we consider that the network elements expose a set of services. These services can include the setting and requesting of particular device states. State, in the most general sense, includes measurements, program parameters, and even the program itself. Using this single abstraction it is possible to enable real-time re-tasking of sensor nodes and robots.

C. Integration Framework

The proposed integration framework is based around Remote Procedure Calls. The Remote Procedure Call (RPC) [14] has long been used for the provision of services in computer systems. The RPC system presented here is light-weight and uses a stateless protocol that allows a client running on a host computer(s) or sensor node(s) to seamlessly access services on one or more sensor nodes. Servers are written in C and clients are written in C or Python.

Figure 2 illustrates the method of code generation for this proposed RPC programming framework. In order to realise this RPC-based integration framework, the system consists of 5 elements; (1) A sensor node operating system (FOS), (2) an RPC code generator, (3) an interactive monitor, (4) a Tagged Data Format (TDF) message format, and (5) the autonomous vehicle software.

1) FOS: In the work presented here, the sensor network nodes and the communication gateway between the robot and sensor network is the Fleck^{TM} [15]. A large number of interface boards have been developed for this wireless network hardware including GPS, IMU, camera and audio boards, and 1-wire temperature sensor strings.

The Fleck Operating System (FOS), initially developed for the Fleck^{TM} sensor network board but which can equally operate on other sensor node hardware, provides a priority based, non-preemptive threading environment with separate fixed-size stacks for each thread and a separate interrupt stack. FOS

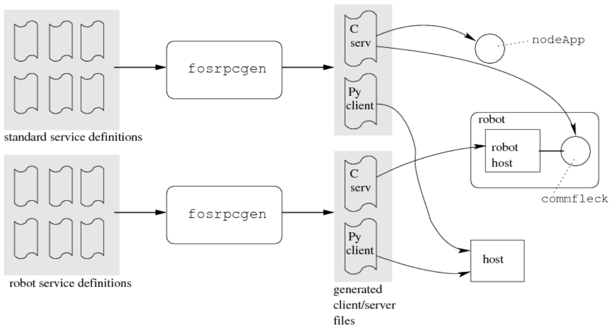

Fig. 2. RPC service definitions are used to generate code that runs in nodes, host and robot host, aiding system integration. Two sets of services are shown: FOS standard services and robot services.

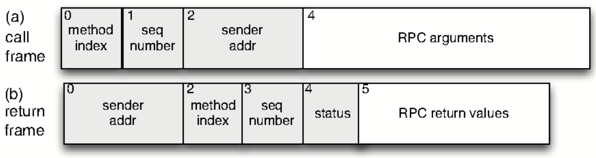

Fig. 3. Compact message structure for RPC call and returns via the sensor network.

provides uniform access to underlying resources via a POSIX-like API for configuring hardware and sending/receiving data using the hardware, including the SPI-bus, serial ports, non-volatile storage etc.

2) RPC Code Generator: The services to be exported are defined in terms of server-side code segments called actions and which closely resemble C-language functions. The RPC generator (`fosrpcgen`) is invoked from the command line and creates several output files for the server and client side of the node.

As our networking and MAC layers do not perform message fragmentation and reassembly, the RPC call and return are constrained to fit within the available payload of the radio message as shown in Figure 3. The available payload is typically 17 to 19 bytes and depends on the multi-hop routing protocol implemented. It is critical that the mapping from method index to service handler is consistent across all nodes and host-side tools. Therefore, `fosrpcgen` uses a simple text file to maintain a persistent mapping. The client and server output file are then compiled for the respective platform (either robot, sensor node or host).

3) Interactive monitors: A number of tools have been developed to invoke arbitrary functions on one or more mobile and fixed nodes. Current interactive tools include command line interfaces, interactive webpages, and internet enabled mobile phones.

4) Tagged Data Format: Tagged Data Format (TDF) is a generic self-describing schema for data transmitted within the network. When a node sends data it is packed into the payload of the message and tagged, prepended by a unique byte value to identify the type of data and implicitly its length in bytes. This enables creation of generic back end tools that can parse the message payloads without knowledge of the application

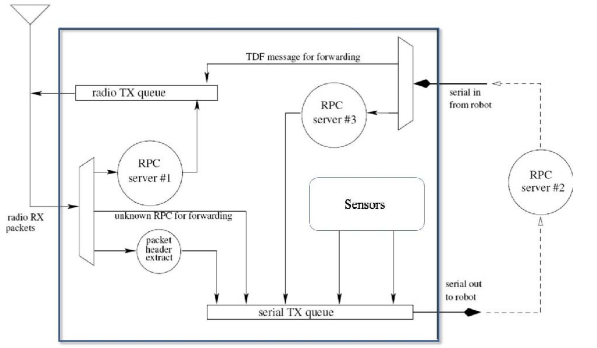

Fig. 4. Data flow in a communication node acting as the gateway between the sensor network and the robot's computer. Note the two RPC servers, one serving requests from the sensor network, and the other from the robot.

that generated them. We maintain a simple text file, a master list of all data types used by our sensor networks.

5) Autonomous Vehicle Software: Robots are typically built using Linux/x86 class computers and are equipped with a range of sophisticated sensors. We use the open-source robot software architecture DDX (Dynamic Data eXchange) [16]. DDX is component-orientated and allows run-time composition of components. Each component is a Unix process and communicates using what is effectively a distributed blackboard system.

On the autonomous surface vehicle considered in this paper, an RPC server running on the boat's Linux computer handles the following commands which are forwarded from the sensor network:

- `getpose`, return the robot's (x,y) position and heading angle
- `gotoxy`, request the robot to travel from its current location to the coordinate (x,y)
- `setvel`, set the robot's speed, heading, and duration
- `settime`, set the robot's onboard clock

The RPC server on the ASV is created using the same toolchain as for RPC servers running on sensor nodes.

D. Node Data Flow & Service Forwarding

In the experiments described in Section III we use a standard sensor node (Fleck) for the gateway (also known as a commFleck) between the sensor network and the autonomous vehicle's computer. The gateway application has two RPC servers each created by `fosrpcgen` as shown in Figure 4.

These servers deal individually with messages from the network or from the robot. If the gateway is unable to fulfil the service request itself, it passes the request to the robot host. If the robot is capable of fulfilling the request, it does so otherwise it sends a negative acknowledgement.

III. Experimental Evaluation

A. Hardware

1) Sensor Network: An experimental evaluation of the proposed RPC-based framework was undertaken using Australia's largest ad-hoc water quality sensor network. The Lake

Fig. 6. A floating lake sensor node as used in the evaluation trials.

Fig. 7. The Lake Wivenhoe Autonomous Surface Vehicle.

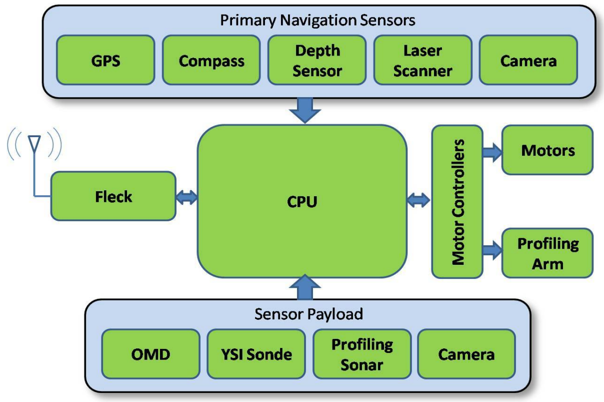

Fig. 8. The ASV system architecture showing the gateway (Fleck) that integrates the vehicle to the sensor network.

Wivenhoe sensor network [17] is located west of Brisbane, Australia (see Figure 5), and consists of 50 floating nodes with coverage over approximately 80 km^2 of the water storage. Each node, shown in Figure 6, has wireless ad-hoc communications provided by the Fleck^{TM} which has Nordic NRF905 radio transceiver operating in the 915 MHz ISM band. The antenna is a high-gain (6dB) whip mounted atop a mast 1.5 m above the water surface in order to improve range. The node has a thermister string (with 6 sensors) attached to the anchor rope in the water column. The sensor node reads the string and sending temperature measurements every minute through the network to the gateway node. Vertical temperature profiling of the water provides information about water mixing within the lake and can be used to help predict the development of algal blooms.

In this initial set of experiments, a subset of 5 of the sensor network nodes were moved closer to each other and their 6dB antennas replaced with unity gain antennas to reduce communication range to ensure that multi-hop routing was required between distant nodes and the shore gateway.

The floating node application comprised three concurrent threads: sample and return engineering data, sample and return temperature string data, and a watchdog. The engineering data returned includes: battery voltage and current, solar cell voltage and charging current, onboard temperature, downstream and upstream link quality, the number of hops to the data sink, and parent node. All data is returned in TDF format reports and the sample interval reset to 5 minutes. The node runs six other system threads associated with the networking function and RPC server.

2) Autonomous Surface Vehicle: The second component of the experimental evaluation was the mobile node, a 16 ft twin-hull Autonomous Surface Vehicle (ASV) as shown in Figure 7. The ASV is a mobile node with three key purposes in the context of this paper. Firstly, it is tasked to collect high spatial and temporal density water quality measurements throughout the lake. Secondly, it can be used to periodically calibrate the static sensors. Calibration or recalibration of sensors in the field is an important operational issue that is not much discussed in the literature and becomes more important as network scale increases. Thirdly, it can be used to repair network connectivity in the event of node failure or strong channel fading.

The ASV is propelled by a pair of electric trolling motors, with another electric motor that moves the twin rudders for steering. Two large solar panels provide power, a mast mounted scanning laser rangefinder provides obstacle detection, and a depth sounder is fitted. The onboard computer is an Pentium M 1.4GHz running Linux that communicates with the onboard control and sensing subsystems via a serial port as shown in Figure 8. Further details of the ASV can be found at [18]. Another serial port connects to the onboard Fleck gateway node for communications with the floating sensor network.

Although the ASV is capable of profiling the water column with its 5m long sensor boom, in these experiments the boat carried an extra sensor node to measure water temperature. The node is essentially a clone of the static nodes except that it has just a single temperature sensor at the same depth as the topmost sensor of the floating nodes, approximately 50 cm. This sensor node transmits readings every 10 s via CTP

Fig. 5. The Lake Wivenhoe water quality sensor network with node locations shown in green (top left quarter). For scale note the city of Brisbane (population 1.9M) to the right.

over the WSN.

B. Results

Using the RPC framework described in the previous section, the mobile robot was tasked to collect fine-grained water temperatures between the nodes. The commands to control the boat's path were issued from the shore using a command line tool connected to an RPC gateway node. Commands issued were position requests using the `gotoxy` which typically corresponded to node locations within the sensor network.

Figure 9 shows a section of the path taken by the boat, as well as the location of the static floating nodes. This path has two sections: 1–17, from node 102 to 101, and 18–49 from node 102 to 104 to 102 then 101. The locations from which a temperature measurement were received at the base, via the network, are shown as small circles. We can see some gaps where measurements were not reported. However, these experiments demonstrated the ability of the RPC framework to reliably monitor the position and status of ASV through multi-hop communication paths as well as the ability to remotely task the ASV across the network.

Figure 10 shows the temperature measured by the boat along the path just described. Also shown, as horizontal lines, are the average temperatures measured at the locations of the static nodes. Several things are evident from the graph. Firstly, the temperature measurements are highly quantized, and the total span is just 0.6 Degrees Celcius. Secondly, there is evidence of variation in water temperature as the robot moves between nodes. The robot starts (point index 1) in the vicinity of node 102 and moves through node 103 (index 6) to node 101. We can see the temperature measurements show an increasing trend, moving from the temperature near node 102 to that near 103 and approaching that at node 101. The temperature sensors used in this experiment have a significant time constant (approximately 30s) so the temperature measurements are lagging—this needs to be changed for future work. The second path segment, starts in the vicinity of node 102 (index 18) heads toward node 104 and shows a decreasing temperature trend as expected, then moves back toward node 102 and then on to node 101. This demonstrates that the robot node can, indeed, provide unprecedented *fine-grained spatial* physical phenomenon measurements in real-time. Furthermore, when the mobile node is close to a static node, we can also use the sensor readings of the mobile node to *recalibrate* the sensor readings of the static node.

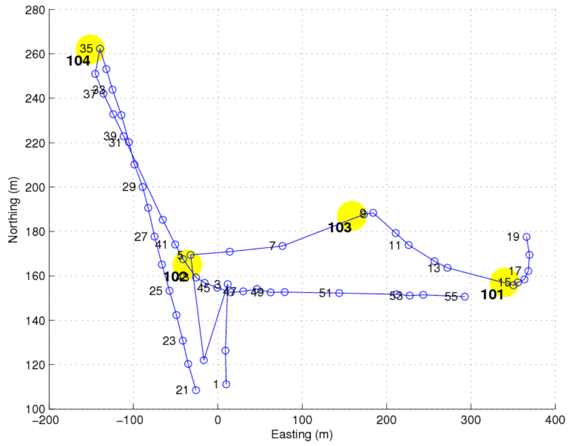

Fig. 9. Path taken by the robot boat. Yellow disks represent floating nodes, 'o' represent points at which a temperature measurement was taken and received at the data sink, and the number is the sequence number of the measurement.

IV. Discussion

The effectiveness of our proposed RPC-based service framework for a wireless water quality sensor network and an Autonomous Surface Vehicle has been demonstrated in Section III. The evaluation trials proved that the robot control software running on a Linux-class computer with a component-based middle architecture is easily abstracted as a service within our sensor network framework. In these trials, robotics engineers were able to add new services easily and invoke

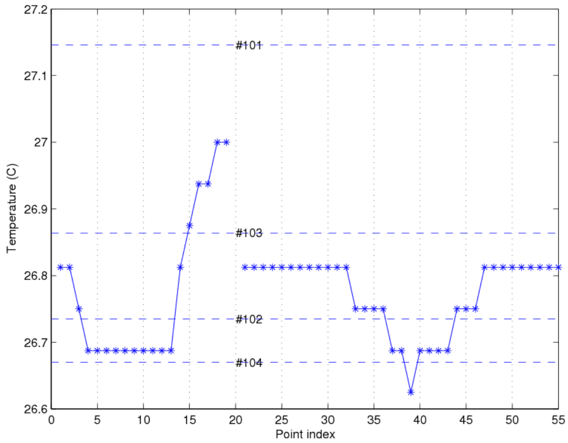

Fig. 10. Temperature measurements taken by the boat along profile shown above. Horizontal lines represent the mean temperature observed at the nodes. Note that temperatures are quantized to 1/16 deg.

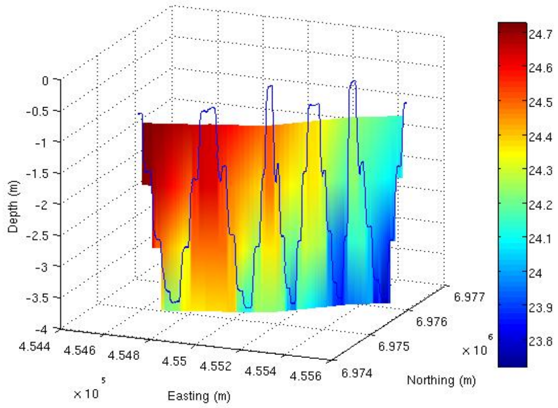

Fig. 12. Interpolated temperature distribution between two sensor nodes 1500 m apart using measurements obtained by the ASV with the profiling sensor arm. The blue trace shows the sensor trajectory through the water column between 1 and 4 meters below the surface.

Fig. 11. The ASV can undertake more complex tasks requested remotely via the sensor network such as "dock into the security cage".

them from the Python interpreter without needing to know anything about serial forwarders, MAC or routing protocols. Similarly the sensor network engineers were able to query and change the state of the robot in exactly the same way they currently do this for a remote sensor node.

Based on the above RPC framework, a further set of experiments have been conducted whereby, the same commands are capable of tasking the ASV to undertake more complex services such as "ASV to collect an AUV at location XY" and "ASV to return home to docking cage" (see Figure 11). Furthermore, tasks based on the `gotoxy` function have been implemented whereby the sensor boom onboard the ASV is used to profile the water column between two nodes such as shown in Figure 12 to determine the spatio-temporal variation in various water parameters.

Current work is now focusing on multi-robot interaction using the proposed framework as well as collaboration between ASVs and AUVs. The framework described in this paper forms a compact foundation for multi-robot and sensor network interaction, however, when complex services are required and multiple assets can address these, higher level deliberation between the devices is required to choose the most appropriate robot to perform the service.

In the implementation described in this paper, the sensor network has communicated by through-air radio. This has limitations for controlling other marine vehicles such as AUVs unless they are at the surface. A key factor in this RPC programming framework is that the overhead is rather low and using different communication modalities such as underwater acoustics, and a microcontroller that can interpret the messages such as described in [19], the same system could be applied both above and below the waterline.

V. Conclusion

This paper has presented the design, implementation and an experimental evaluation of a unique RPC-based programming framework allowing creation of effective systems of static and mobile sensors that are suitable for tasks such as large-scale environmental monitoring. In the proposed framework, robots and static sensors consider each other as service providers and consumers, which allows us to create effective systems of static and mobile sensors that provide unprecedented fine-grained spatial-temporal resolution for large-scale marine environmental monitoring. Additionally, in the presence of uncertainty the framework allows seamless re-tasking of sensor nodes and robots by sending commands or uploading new code which will be essential to the utility of the system.

Acknowledgment

The authors would like to thank Wen Hu and Brett Wood from the CSIRO Autonomous Systems Laboratory for their contributions during the construction, testing and evaluation of the system.

References

[1] G. Sibley, M. Rahimi, and G. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks," in *Proc. Int. Conf. Robotics and Automation*, Washington, May 2002, pp. 1143–1148.

[2] S. Bergbreiter and K. Pister, "CotsBots: an off-the-shelf platform for distributed robotics," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, October 2003, pp. 1632–1637.

[3] M. Hsieh, A. Cowley, V. Kumar, and C. Taylor, "Towards the deployment of a mobile robot network with end-to-end performance guarantees," in *Proc. Int. Conf. Robotics and Automation*, May 2006, pp. 2085–2090.

[4] F. Zeiger, N. Kraemer, and K. Schilling, "Commanding mobile robots via wireless ad-hoc networks – a comparison of four ad-hoc routing protocol implementations," in *Proc. Int. Conf. Robotics and Automation*, May 2008, pp. 590–595.

[5] M. B. Dias, R. Zlot, M. Zinck, J. P. Gonzalez, and A. Stentz, "A versatile implementation of the TraderBots approach for multirobot coordination," in *Proc. of the 8th International Conference on Intelligent Autonomous Systems (IAS-8)*, 2004.

[6] T. Voigt, F. Osterlind, N. Finne, N. Tsiftes, Z. He, J. Eriksson, A. Dunkels, U. Bamstedt, J. Schiller, and K. Hjort, "Sensor networking in aquatic environments – experiences and new challenges," in *Proc. Second IEEE International Workshop on Practical Issues in Building Sensor Network Applications*, 15-18 October 2007.

[7] C. Palazzi, G. Woods, I. Atkinson, and S. Kininmonth, "High speed over ocean radio link to great barrier reef," in *Proc of IEEE TENCON'05*, Melbourne, Australia, 21-24 November 2005.

[8] K. Johnson, J. Needoba, S. Riser, and W. Showers, "Chemical sensor networks for the aquatic environment," *Chemical Reviews*, vol. 107, pp. 623–640, 2007.

[9] G. Timms, J. McCulloch, P. McCarthy, B. Howell, P. de Souza Jr., M. Dunbabin, and K. Hartmann, "The Tasmanian marine analysis network (TasMAN)," in *Proc. OCEANS'09*, 2009.

[10] D. Codiga, J. Rice, and P. Baxley, "Networked acoustic modems for real-time data delivery from distributed subsurface instruments in the coastal ocean: Initial system development and performance," *Journal of Atmospheric and Oceanic Technology*, vol. 21, no. 2, pp. 331–346, 2004.

[11] A. Pereira, H. Heidarsson, C. Oberg, D. Caron, B. Jones, and G. Sukhatme, "A communication framework for cost effective operation of AUVs in coastal regions," in *Proc. of the 7th International Conference on Field and Service Robotics*, July 2009.

[12] G. Sukhatme, A. Dhariwal, B. Zhang, C. Oberg, B.Stauffer, and D. Caran, "Design and development of a wireless robotic networked aquatic microbial observing system," *Environmental Engineering Science*, vol. 24, no. 2, pp. 205–215.

[13] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2003, pp. 14–27.

[14] P. G. Soares, "On remote procedure call," in *CASCON '92: Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 1992, pp. 215–267.

[15] P. Sikka, P. Corke, L. Overs, P. Valencia, and T. Wark, "Fleck a platform for real-world outdoor sensor networks," in *Proc. 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP 2007)*, Melbourne, Australia, 3-6 December 2007, pp. 709–714.

[16] P. Corke, P. Sikka, J. Roberts, and E. Duff, "DDX: A distributed software architecture for robotic systems," in *Proc. Australian Conf. Robotics and Automation*, Canberra, Dec. 2004.

[17] M. Dunbabin, J. Udy, A. Grinham, and M. Bruenig, "Continuous monitoring of reservoir water quality: The Wivenhoe project," *Water*, vol. 36, no. 6, pp. 74–77, September 2009.

[18] M. Dunbabin, A. Grinham, and J. Udy, "An autonomous surface vehicle for water quality monitoring," in *Proc. Australasian Conference on Robotics and Automation*, December 2009.

[19] K. Bengston and M. Dunbabin, "Design and performance of a networked ad-hoc acoustic communications system using inexpensive commercial CDMA modems," in *IEEE OCEANS 2007 - Europe*, June 2007.