2011 Baltic Congress on Future Internet and Communications

An Underwater Convergence Layer for Disruption Tolerant Networking

Diego Merani, Alessandro Berni and John Potter
NATO Undersea Research Centre
19126 La Spezia, Italy
Email: merani,berni,potter@nurc.nato.int

Ricardo Martins
LSTS – Underwater Systems and Technology Laboratory
Universidade do Porto - Faculdade de Engenharia
Rua Dr. Roberto Frias I204
4200-465 Porto, Portugal
Email: rasm@fe.up.pt

***Abstract*—Delay and Disruption tolerant networking (DTN) embraces the concepts of occasionally-connected networks that may suffer from frequent partitions and that may be comprised of more than one divergent set of protocols or protocol families. Since DTN was conceived to relax most of the assumptions associated with Internet protocols, the most important of which is the availability of an end-to-end path between source and destination for the duration of a communication session, it is possible to envision the application of DTN concepts to underwater acoustic networks, where the bandwidth-limited acoustic channel suffers frequent disruptions due to environmental factors. This paper presents the design and implementation of an Underwater Convergence Layer (UCL) for the DTN2 Reference Implementation, which provides DTN support for the WHOI Micro-Modem. UCL was tested in the field during the Acommsnet10 sea trial, in a heterogeneous context comprising underwater nodes communicating acoustically and surface nodes communicating with radio frequency. Initial results and lessons learned from field testing are also presented. The principal value of this development resides in the delivery of new capabilities towards the application of DTN concepts to the maritime domain, to enable functionalities like multi-hop acoustic communication, data mule using autonomous vehicles and gateways between different transmission media (e.g. between underwater acoustic communications and above water radio/satellite).**

***Index Terms*—Underwater acoustic networks, acoustic communications, discovery, delay and disruption-tolerant networking.**

I. Introduction

Mobile ad-hoc networking techniques can be effectively applied to the maritime domain to build scenarios in which surface nodes (ships, buoys) and unmanned air vehicles (UAV) communicate across radio frequency (RF) links: however a comprehensive solution for underwater networking does not yet exist.

This shortcoming follows from the fact that RF propagation below the sea surface is severely impaired, leaving acoustic communication as the foundational technology to interconnect nodes in the underwater domain. A major challenge of acoustic communication is that propagation of sound in the water occurs at a speed that is five orders or magnitude smaller than above-water RF (1,500 m/s versus 300,000 km/s) across a time-varying bandwidth-limited channel that is severely impacted by environmental conditions and subject to frequent disruptions.

As a consequence of the spatial-temporal variability of the acoustic channel it is also common that within underwater acoustic networks disconnection is much more common than connection. Additional sources of disconnection may derive from node mobility, as in the case of autonomous underwater vehicles (AUVs) and gliders, or from low-duty-cycle operation of fixed nodes that must use energy sparingly to carry out long-term missions. The above constraints make underwater acoustic networks incompatible with important assumptions associated to Internet protocols, such as the availability of an end-to-end path between source and destination for the duration of a communication session.

For this reason Delay and Disruption-Tolerant Networking (DTN) represents an interesting option towards the reliable interconnection of underwater networks using a general-purpose message-oriented overlay architecture [1] [2]. Based on earlier work on Interplanetary Networking, DTN represents an abstraction of asynchronous message delivery, where message aggregates are called bundles and the routers that handle them are called bundle forwarders or DTN gateways.

This store-and-forward mechanism can be exploited to deliver increased robustness and reliability against link disruption as well as to introduce useful features for maritime networks. Increased reliability is implemented on a hop-by-hop basis (as opposed to end-to-end, which would be unpractical within networks suffering frequent disruptions) using the “custody transfer” concept to delegate responsibility on a data segment (bundle or bundle fragment) to the following node in the path towards the remote endpoint.

In this paper we describe the application of the DTN2 Reference Implementation (RI) [2] [3] [4] to create networks composed of heterogeneous links, some of which provided by low-bandwidth of acoustic modems.

In order to interface acoustic modems with the DTN2 RI we have developed an open-source convergence layer that is positioned between the DTN layer and the underlying acoustic communications protocols. This communications framework aims to allow access to acoustic-modems of several vendors in the form of a DTN2 Convergence Layer Adapter (CLA).

978-1-4244-8513-0/11/$26.00 ©2011 IEEE

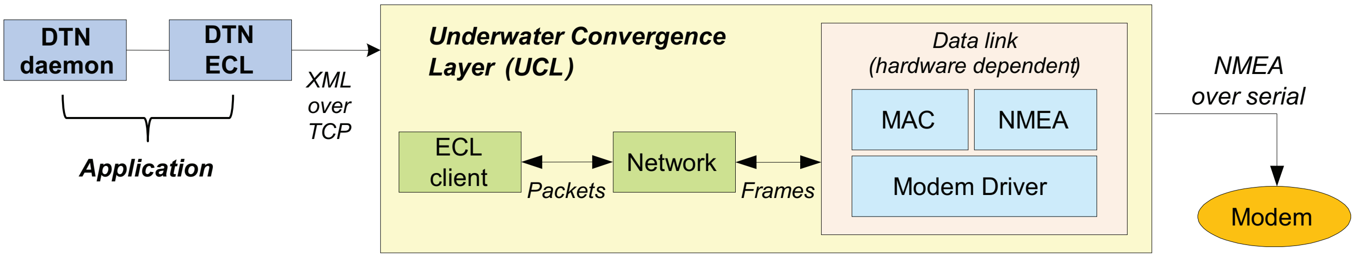

Fig. 1: Depiction of modules composing the Underwater Convergence Layer.

The role of the CLA is to manage protocol-specific details associated to acoustic communication, while presenting a consistent interface to the DTN2 framework.

In the initial implementation only the Woods Hole Oceanographic Institution (WHOI) Micro-Modem [5] is supported although work is ongoing to support additional types of acoustic modems, taking advantage of the modular structure adopted in the development of the code.

The remainder of this paper is organized as follows. Section II gives an overview of the DTN2 RI and describes the architecture of our software framework. In Section III we describe and present the preliminary results of the latest field tests. Finally, Section IV concludes and highlights directions for future work.

II. Software Architecture

A. Relationship with DTN2 Reference Implementation

DTN2 implements the DTN Bundle Protocol [6], a protocol for transmitting data over various link layers, such as Ethernet or Bluetooth, in the form of bundles that may be much larger than typical packets transmitted on Internet Protocol (IP) networks.

While it was designed as a platform to validate protocol designs and to assess performance and suitability of DTN protocols in an experimental context, DTN2 also provides a number of mechanisms to direct the forwarding of bundles to their intended destinations, including dynamic routing algorithms and a static routing scheme based on pre-configured routes [7].

The reference implementation also includes a number of interfaces between the Bundle Protocol and the transport layer, in the form of Convergence Layer Adapters (CLA). CLAs manage protocol-specific details of interfacing with a particular underlying protocol, network or device, presenting a consistent interface towards the bundle layer.

The complexity of one convergence layer may vary substantially from another, depending on the type of underlying protocol to which adaptation is provided. For example, a TCP/IP convergence layer for use on the Internet might only have to add message boundaries to TCP streams, whereas a convergence layer for some network where no reliable transport protocol exists might be considerably more complex (e.g., it might have to implement reliability, fragmentation, flow-control, etc.), if reliable delivery is to be offered to the bundle layer [2].

DTN2 includes a special convergence layer called the External Convergence Layer (ECL) that allows for the implementation of CLAs outside of the DTN2 code base. CLAs implemented using the ECL interface exchange XML-encoded [8] messages with the ECL via a TCP/IP socket. The set of available XML messages is similar to the one described in [9].

Interface messages are broadly divided into four categories. Event messages are issued by forwarders to indicate state changes that may be of interest to external routers. Request messages are sent by external routers to direct forwarders to perform an action (e.g. “forward the bundle with ID 56 out link tcp0”). Query and report messages are used by external routers to synchronize their state with forwarders after startup or during failure recovery.

The Underwater Convergence Layer (UCL) software framework was implemented using the C++ programming language and runs on POSIX compliant operating systems. The convergence layer and associated DTN2 specific logic act as a client to the DTN2 ECL. The remaining parts of the UCL code are independent of DTN2 and can be used for other applications (e.g. testing Media Access Control (MAC) protocols, compression methods).

B. Software Implementation

In its present form the Underwater Convergence Layer (UCL) software delivers a CLA for the WHOI Micro-Modem to be used in conjuction with the DTN2 RI. The system structure is summarized in Fig.1.

1) Platform Access and Abstraction (PAA) Module: This module provides access to advanced operating system specific functionalities. More specifically, it implements the following features:

- logging relevant messages and performance statistics to files and to the system console;
- configuring and performing bi-directional communication with serial port devices and TCP/IP sockets;
- support for threading and concurrency.

2) Data Link (DL) Module: The Data Link (DL) module is in charge of delivering and receiving frames, maintaining a list of reachable nodes and advertising the local node in the acoustic network.

The data unit within the DL module is the frame, containing one byte of control information and up to *n-1* bytes of user data, where n is equal to the Maximum Transmission Unit (MTU) of the acoustic link.

Node discovery is implemented using an algorithm that dictates that every node that does not have any data frames to transmit within a certain amount of time must actively advertise itself, thus allowing other nodes to learn about its presence and reachability.

In our implementation with the WHOI Micro-Modem this active advertisement takes the form of a mini-packet directed to the broadcast address.

Every transmission is mediated by a MAC module, also positioned within the DL module, which has visibility over all relevant network events (e.g. transmission in progress) and decides when to execute pending transmissions. The current MAC module was designed to ensure simple collision avoidance and random backoff.

The following rules are used to mediate access to the medium:

- when a remote node is transmitting, the local node will abstain from transmitting until the medium is assumed to be available again; If the transaction requires an acknowledgment, the local node will wait for that event to occur;
- whenever a transmission is completed, the node responsible for it will not transmit again until a random period of time has passed;
- if a collision is detected, the node will retry after a random period of time or when the medium seems to be available again.
- a random timeout value is used to avoid waiting for events that will probably never occur, such as waiting for an acknowledgment that was not transmitted.

3) NMEA Module: The WHOI Micro-Modem is interfaced through a set of NMEA [10] sentences that are used for setup, control, communication and navigation [11].

The NMEA module contains C++ classes that can be used both to parse messages generated by the WHOI Micro-Modem and to translate C++ objects to NMEA sentences. The objective is to deliver a translation between native modem messages and more generic structures that can be used by other modules.

Each NMEA sentence is mapped into a C++ class by code generation. For this purpose, this component uses an XML document that describes the structure of all relevant NMEA sentences (identifier, field names, field types). This XML document is then transformed into C++ code by using an XSLT [12] document in a similar fashion to the one described in [13]. For each NMEA sentence there is a C++ class providing the following functionalities:

- data encapsulation through accessor and mutator member functions, with validating logic, for all fields;
- a member function used to generate a NMEA compliant representation of the object;
- another member function to initialize the member variables of the object by parsing NMEA sentences;

4) WHOI Micro-Modem Driver (WMMD): This module provides a high-level interface to control the WHOI Micro-Modem. The driver:

- uses the PAA module for serial port access to the modem;
- interacts with the modem using the NMEA module;
- services requests to transmit data.
- sends notifications to other modules when:
  1) local and remote modems are transmitting;
  2) data and/or acknowledgments arrive;
  3) collisions occur;

5) Network Module: The Network Module exposes an interface to send and receive data in the form of packets, defined as an arbitrarily long protocol data unit.

This module implements transparent compression and decompression using Bzip2, a Burrows-Wheeler block-sorting data compressor [14] [15], in order to minimize the size of transmitted data. To achieve good compression, Bzip2 needs input blocks of several thousand characters. When Bzip2 is not capable of reducing the size of a packet, the packet is sent uncompressed.

After the compression stage a packet is fragmented into frames and sent to the DL module for delivery. Reassembly of packets is performed once the last frame of a packet is received.

6) ECL Client: The ECL client interacts with the local DTN2 ECL. It is responsible for managing DTN-capable communications using for that purpose the API provided by the Network Module. Communication management consists of maintaining visible nodes as a set of reachable destinations, reporting incoming bundles and also sending bundles on demand.

The ECL module:

- maintains a list of links that are presently open. All links are considered to be opportunistic, unscheduled and unreliable;
- parses, validates and generates DTN ECL compliant XML messages.
- uses the ECL XML Schema for communication with the local DTN daemon. ECL XML messages are exchanged through a TCP/IP socket supplied by the PAA module;
- informs DTN about acoustic links available within range.

III. FIELD TESTING

Following an initial phase of testing in a laboratory environment, field testing for the UCL implementation was conducted during the Acommsnet10 sea trial that took place in September 2010 by the Island of Pianosa, in Italy's Tuscan Archipelago.

A total of eleven trials were conducted, with increasing complexity both in terms of topology and assets involved. The following assets were made available:

- three fixed acoustic nodes positioned to form a triangle with side length in the order of 1 km (positioned at 15, 21, 30 m depth);
- one fixed acoustic node mounted on a buoy (positioned at 13 m depth), which could be repositioned to vary the topology of fixed nodes;
- three mobile nodes equipped with acoustic modem (at variable depth) and IEEE 802.11n wireless interface, one

mounted on NURC's Research Vessel Leonardo [16], the other two mounted on rigid-hulled inflatable boats (RHIBs)

During the trial we intended to test some fundamental features of DTN, which will prove to be instrumental in the design and operation of future maritime networks. Examples include:

- Basic and unidirectional DTN messaging: exchanging DTN bundles between two nodes when a workable channel exists, saving the bundle in persistent storage when it does not, and sending DTN messages to a receive-only underwater node.
- DTN custody: A message is sent from one node to another (beyond single-hop acoustic communications range) via a surface or submerged intermediate gateway. The gateway acts as custodian and relays to the destination node when it comes in range.
- Developing and testing gateways to bridge clusters of underwater networked assets and linking them to existing Command and Control networks and to the internet.

A summary of the most significant scenarios experimented during Acommsnet10 is provided in the following subsection.

A. Experimental scenarios

1) Acoustic multihop: The underlying idea that originated this scenario is that traditional networking protocols (such as TCP/IP) perform poorly, or do not work at all, if there is no end-to-end network path from source to destination (intermittent connectivity), in presence of high bit error rates or long/variable delays. Underwater networks suffer all those limitations because of high variability, both in space and in time, of the acoustic channel. DTN offers a set of mechanisms that may increase the performance of the disadvantaged networks.

In this scenario fixed node A sends a bundle to fixed node D that is positioned outside its acoustic communications range. Node D is however reachable passing the bundle through nodes B and C, as summarized in Fig. 2.

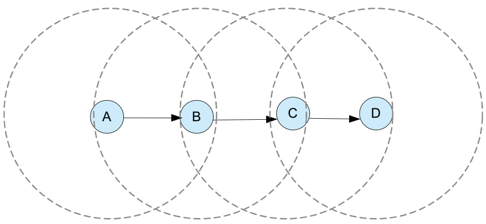

Fig. 2: Acoustic multi-hop scenario.

2) Data Mule (with one and two data mules): The "store and forward" approach of DTN and the lack of end-to-end connectivity that characterize DTN networks, allow the exploitation of non-traditional routing approaches. In the case of mobile nodes, the interaction between routing algorithms and the application layer, allows to manipulate network topologies according to communication requirements, exploiting scheduled or opportunistic contacts between nodes.

The Data Mule is a vehicle within the DTN underwater network, e.g. an AUV, which physically carries data between two or more non-adjacent network nodes, to enable network communication beyond acoustic link capacity and range.

In this scenario nodes B and D have to send a bundle to node A which is outside the communication range of both and no intermediate fixed nodes exist that can act as data relay.

A mobile node M1 is tasked to collect the outgoing bundle from fixed node B, while mobile node M2 is tasked to perform the same function for fixed node D. After storing the bundle in persistent storage the data mule moves closer to fixed node A, and when it enters the acoustic communications range it proceeds with the delivery of pending bundles.

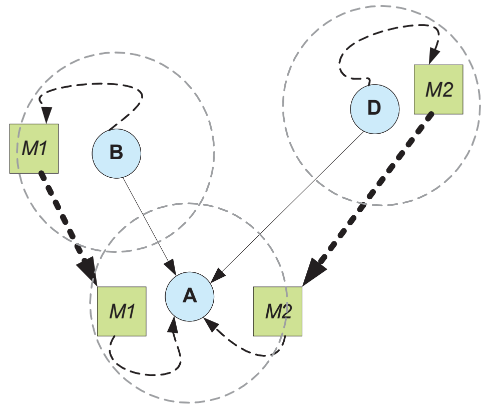

Fig. 3: Data mules enable communications between far away fixed nodes.

3) Bridging clusters of acoustic modems with 802.11n: The DTN bundle protocol can be seen as an intermediate abstraction layer between the applications and the network transport. Its ability of relying on many different transport protocols (through dedicated convergence layers), both in IP and non-IP networks, makes it a good choice to interface traditional networks (such as TCP/IP based command and control networks or the Internet) with non-traditional networks (such as underwater acoustic networks). In this scenario fixed nodes A and B, representative of cluster heads for two distant underwater clusters, have to exchange traffic on a regular basis. Mobile nodes M1 and M2 shuttle between the two clusters and exchange pending bundles (across a radio link) as soon as they enter in range.

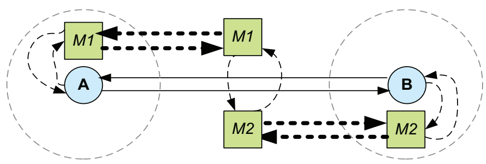

Fig. 4: Exchange of DTN traffic using data mules across different media.

Successful completion of this trial provides proof-of-concept of interfacing an Underwater Acoustic Network with an heterogeneous IP-based networks across a wide range of media, regardless of the difference in bandwidth, reliability and latency between the underwater domain and above-water wireless networks.

4) MAC performance testing: This "ring" scenario was used to evaluate the behavior of the simple MAC protocol embedded in the UCL when nodes try to transmit in a concurrent fashion.

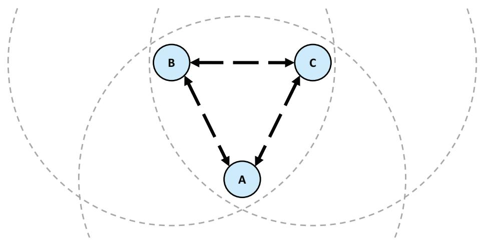

Fig. 5: Topology for MAC protocol testing.

B. Preliminary results from field testing

As mentioned earlier, the current version of the UCL delivers support for the WHOI Micro-Modem. The WHOI Micro-Modems that were used for field testing were configured to use Frequency Hopping FSK "Rate 0" modulation, which has the characteristics shown in Table I.

TABLE I: WHOI Micro-Modem FH-FSK characteristics.

Bandwidth	4 kHz
Burst rate	80 bps
Frames per packet	1
Frame size	32 bytes
Packet length (in time)	3.9
Signal length (probe+null+packet)	12.15 s

The challenges of underwater acoustic communications that were mentioned in the introduction of this paper become evident when looking at the performance that can be delivered by this particular type of acoustic modem.

Even if the burst rate is as low as 80 bits per second, the best-case throughput that is actually made available to the user is in the order of 20 bits per second, as a consequence of the modems' cycle-init process and duty cycle constraints, with the additional complication associated to underwater acoustic communications, such as significant spatial-temporal channel variability and high latency (e.g. about 700 milliseconds are required to cover the distance of 1 km).

During the field testing we were able to verify that the UCL operated according to the specifications; however, a performance issue was observed with denser deployments. With an underwater network of six nodes, the contention to the shared underwater channel became evident through a very high number of collisions. The cause of this issue was tracked down to an implementation flaw that allowed for node advertisements to bypass the MAC module.

In the original implementation of the UCL the policy for transmitting node advertisements was driven by the time elapsed since the previous transmission event, i.e. an advertisement was transmitted n seconds after the previous transmission, as long as the modem was not currently engaged in receiving a frame. This implies that a node that had just finished receiving a frame could send an advertisement (because the advertisement timer had expired) in place of the acknowledgment associated with the successful reception of the frame.

To correct this problem, during the execution of the trial we implemented the handling of node advertisements as regular data transmissions, with MAC layer mediation, transmitting the advertisement only if:

- n seconds have elapsed since a transmission
- the DL Module is not receiving
- the DL Module is not transmitting
- the DL Module does not have any pending transmissions (data frames or acknowledgments)

The effectiveness of this modification was assessed in a realistic environment implementing the "ring" setup described in III-A4 instructing all three nodes to transmit one bundle (composed of three frames) to the respective neighbors, in a concurrent fashion.

Fig. 6 shows the results for the original implementation of the node advertisement mechanism, using the vertical axis to depict transmissions of frames between the three nodes, while the horizontal axis shows elapsed time (in seconds since the queuing of the first bundle).

It is easy to note that for the first 10 minutes it was extremely difficult to get any useful traffic across the network, because of unresolved contention in accessing the acoustic channel.

A definite lack of fairness can also be observed, whereas node 1 prevailed over the others in getting access to the channel to transmit the six queued frames (three directed to node 2 and three directed to node 3). Following the completion of transmissions from node 1, we can observe that nodes 2 and 3 were eventually able to access the channel to complete the assigned transmissions.

Fig. 7 shows the performance improvements introduced by changing the handling of node advertisements. A performance improvement in the order of 100% can be observed, coupled with a definite improvement of the fairness of the protocol.

The value of the field testing presented in this section resides mainly in the functionality that we have demonstrated, which can be used to deliver new DTN capabilities in the maritime domain, and not necessarily in quantitative performance metrics. The performance improvement deriving from the correction of an implementation flaw that allowed advertisements to bypass MAC module, discussed in III.B, can be seen as an indication of how the underlying transport services may affect Bundle Protocol functionalities and in the end, the overall performances of the network.

IV. Conclusions and Future Work

The processing of the data collected during the Acommsnet10 trial is still ongoing at the time of writing, in order to

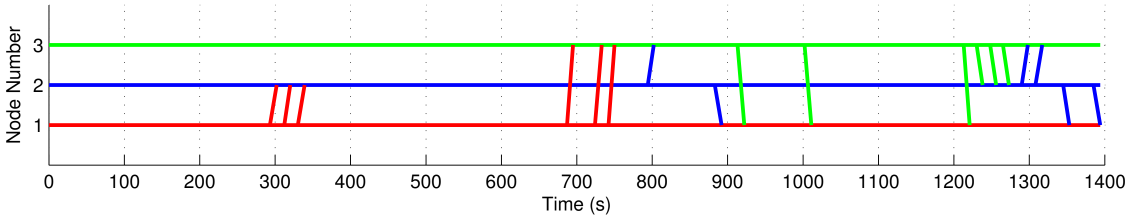

Fig. 6: Performance assessment for concurrent access using "ring" topology (initial implementation).

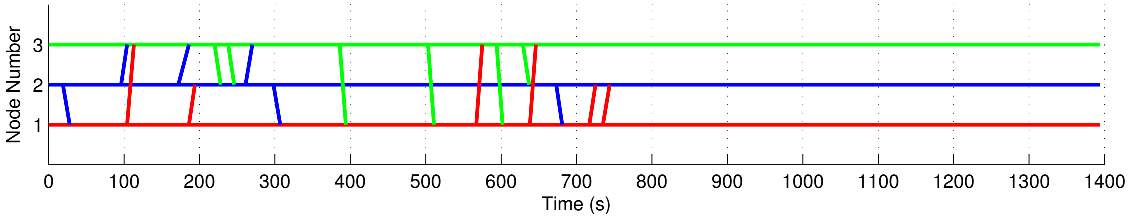

Fig. 7: Performance assessment for concurrent access using "ring" topology (improved version).

reach a greater understanding of the relationship between environmental factors (natural and antropic) and network protocol performance.

In a broader perspective, we have shown that DTN2 and UCL can be used to transparently and reliably interconnect "traditional" IP-based and acoustic networks. The software framework we have developed is suited for use in maritime hybrid networks, e.g. in gateways between "traditional" IP-based networks and underwater acoustic networks, in networks with mobile nodes serving as data relays/mules, and for mission critical transactions where data must be delivered reliably across a set of highly heterogeneous links in order to reach the intended destination.

The layered modular approach chosen is a fundamental enabler of future developments and performance improvements. Indeed the software has been developed with a particular attention to layer abstraction: the specificity of the UCL together with the detailed definition of interfaces, services and functionalities, will simplify future enhancements such as support for multiple MAC protocols or different acoustic modems, as a step towards a fully software-defined and modular underwater networking platform.

ACKNOWLEDGMENT

The authors would like to thank Robert Been, Piero Guerrini, João Alves, Roberto Lombardi of NURC and Lee Freitag of WHOI for the useful discussions and for the fruitful collaboration in the planning and execution of the Acommsnet10 trial. Universidade do Porto would like to thank CONET, the Cooperating Objects Network of Excellence.

REFERENCES

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 27–34.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "RFC4838: Delay-tolerant networking architecture," 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt

[3] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra, "Implementing delay tolerant networking," in *Intel Research Technical Report IRB-TR-04-020*, 2004.

[4] "Bundle Protocol Reference Implementation (DTN2)," 2010. [Online]. Available: http://www.dtnrg.org/wiki

[5] L. Freitag, M. Grund, I. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: an acoustic communications and navigation system for multiple platforms," in *In Proc. IEEE OCEANS'05 Conf*, 2005.

[6] K. Scott and S. Burleigh, "RFC5050: Bundle protocol specification," 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt

[7] "DTN2 Reference Manual," 2010. [Online]. Available: http://www.dtnrg.org/wiki

[8] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, and J. Cowan, "Extensible Markup Language (XML) 1.1 (Second Edition)," World Wide Web Consortium, Recommendation REC-xml11-20060816, August 2006. [Online]. Available: http://www.w3.org/TR/2006/REC-xml11-20060816

[9] R. Krishnan, C. Small, R. Ramanathan, J. Mikkelson, and P. Basu, "Dtn plugin architecture," 2007. [Online]. Available: http://www.dtnrg.org/wiki

[10] Marine Electronics Association, "NMEA 0183 Standard For Interfacing Marine Electronic Devices, Version 3.01," 2002.

[11] Acoustic Communications Group, Woods Hole Oceanographic Institution, "Micro-modem software interface guide, version 3.03," 2010.

[12] "Extensible Stylesheet Language (XSL) Version 1.1," World Wide Web Consortium, Recommendation REC-xsl11-20061205, December 2006. [Online]. Available: http://www.w3.org/TR/2006/REC-xsl11-20061205

[13] R. Martins, P. S. Dias, E. R. B. Marques, J. Pinto, J. B. Sousa, and F. L. Pereira, "IMC: A communication protocol for networked vehicles and sensors," in *OCEANS 2009-EUROPE, 2009. OCEANS '09.*, 11-14 2009, pp. 1 –6.

[14] J. Seward, "bzip2 and libbzip2," 2010. [Online]. Available: http://www.bzip.org/

[15] M. Burrows and D. Wheeler, "A block sorting lossless data compression algorithm," 1994.

[16] "CRV Leonardo," 2010. [Online]. Available: http://www.nurc.nato.int/research/leonardo.htm