Status Packet Deprecation and Store-Forward Routing in AUSNet

Matt Haag
Emmanuel Agu

Computer Science Dept.
Worcester Polytechnic Institute
Worcester, MA 01609
emmanuel@cs.wpi.edu

Rick Komerska
Steven G. Chappell

Autonomous Undersea
Systems Institute
86 Old Concord Turnpike
Lee, NH, 03824
komerska,chappell@ausi.org

Radim Bartoš

Dept of Computer Science
University of New Hampshire
Durham, NH 03824
rbartos@cs.unh.edu

ABSTRACT

AUSNet is a functional network of autonomous undersea vehicles. We present two novel algorithms to enhance AUSNet. In live in-water testing, a packet queueing problem in which stale status packets accummulated while AUVs were out of transmission range, was discovered. A replace-and-hold algorithm was designed to replace stale status packets. A store-and-forward feature using data shuttles between disjoint network partitions, is also proposed. Both algorithms were evaluated in a time-driven simulator. The replace-and-hold method showed bandwidth usage improvements of 48% for a typical AUV scenario and improved performance even when nodes remained connected. Multiple store-and-forward models were compared and showed varying levels of improvement.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Algorithms, Experimentation, Performance

Keywords

undersea networks, store forward routing, partitioned networks, status packets

1. INTRODUCTION

The recent emergence of Autonomous Underwater Vehicles (AUVs) has fueled a desire for communication capabilities to facilitate node cooperation and data distribution. The Autonomous Undersea Systems Network (AUSNet) [1, 3] was initially created to allow communication between heterogeneous AUV systems and enable the operation of multiple cooperating AUVs in fleet behavior.

The undersea networking environment presents many challenges, including ultra-low bandwidth and extreme latency. Since radio waves do not propagate adequately through water, underwater wireless networking uses acoustic (sound) technology in place of Radio Frequency (RF) waves used in wireless networks. Acoustic modems transmit data using the propagation of sound waves through water. The acoustic network medium suffers from several unique problems. Current state of the art acoustic modems typically operate at a data rate of 800 bps. Transmission latency is limited by the speed of sound (approximately 1500 m/s in water) rather than the speed of light. In a typical AUV deployment, two nodes that are 2 km apart experience a latency of 1.3 seconds.

A growing number of underwater networks consist of multiple AUV systems, as well as many fixed sensor platforms. Thus, underwater acoustic networking also faces the traditional challenges of any mobile wireless network. Consequently, an adapted form of traditional Mobile Ad Hoc Networking systems is a prime candidate for application.

Figure 1: In-water Networking Tests

The AUSNet project has focused on the creation of a multi-faceted networking capability using a specialized implementation of the Dynamic Source Routing (DSR) Protocol [2] fused with an innovative Prediction Based Routing (PBR) Protocol [3] based on

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WUWNet'06, September 25, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-484-7/06/0009 ...$5.00.

the principle of Dead Reckoning. DSR is a reactive ad hoc routing protocol where all nodes can act as routers, and each packet's route is determined by its initial sender. Routes to unknown hosts are discovered via active route discovery, whereby either the desired destination or any intermediary host that knows a route to the desired destination can respond with route information. The use of Prediction Based Routing is unique to AUSNet. When the network topology changes, the route discovery mechanism uses a predicted route rather than a discovered one. Prediction is possible because, unlike typical ad hoc wireless systems, AUSNet nodes have a reasonably accurate picture of their location. Location information is transmitted to all nodes within the system enabling topological awareness. The use of this routing method has been shown to significantly improve bandwidth utilization [4].

AUSNet has been tested in water on several functional AUV systems using Benthos acoustic modems. The Autonomous Undersea Systems Institute (AUSI), in conjunction with Technology Systems, Inc. (TSI) and the Naval Undersea Warfare Center (NUWC), have tested AUSNet with the Solar-powered AUV (SAUV), pictured in Figure 1, and the Mid-sized Autonomous Research Vehicle (MARV). These tests took place in 2004 and 2005 at Lake George, NY, in Narragansett Bay, RI, and in the Hood Canal in Bangor, WA. During these in-water tests, a problem was discovered with the way AUSNet queues messages when the desired destination node is unreachable. This discovery provides the impetus for the first problem investigated, *packet queuing*, discussed in Section 2.1. Secondly, communication between disjoint network partitions by using messenger nodes travelling between the partitions, has long been a desired feature for AUSNet. These messengers nodes store packets from the sender in one partition, travel to the destination node's partition, where they forward stored packets to the destination. A second algorithm investigated, the *store-forward feature*, is discussed in Section 2.2.

This paper proposes a solution to the packet queuing problem, and a store-and-forward packet delivery scheme for AUSNet. Both solutions are evaluated using a simulator and results are presented. The rest of the paper is organized as follows. Section 2 describes the AUSNet packet queuing and store-and-forward issues in more detail, Section 3 presents related work, Section 4 presents our solutions, Section 5 presents simulation results, Section 6 discusses our results, and Section 7 presents conclusions and future work.

2. PROBLEM DESCRIPTION

2.1 Packet Queuing Problem

Within AUSNet, the packet queuing problem is the consequence of a design decision that in retrospect has created several issues. Within the initial AUSNet design, if a route is unreachable when packet transmission is attempted, or if the desired route never existed, the packet is placed in a queue awaiting a proper route. While this appears reasonable, it in fact causes the following problem: When an AUV is operating, it will often send status packets at regular intervals. If the AUV is not within communication range of the destination controlling system, AUSNet enqueues the status packets, awaiting a working connection.

An example is used to illustrate this problem. In the left side of Figure 2, the AUV represented as node 2 is within range of nodes 1 and 3 and sends its status every minute. It then travels away from the other two nodes, until it becomes disconnected as in the right side of Figure 2. Node 2 remains at this point for 20 minutes, while enqueueing a status packet every minute. When it moves back within range and is reconnected to the active network, there will be 20 packets queued. At a representative size of 70 bytes per packet, these 20 packets have a total size of 1400 bytes in the queue. Even more crucially, given the previously discussed bandwidth limitations, this data transmission will consume more than 14 seconds of constant utilization of the communications channel. This increases at an order of $O(n)$, where n is the number of edges required for communication. While this status information is important, packets that are queued later contain updated status information and thus deprecate earlier packets. In order to save valuable bandwidth, it would be beneficial to dequeue deprecated packets and send only the most recent status packets.

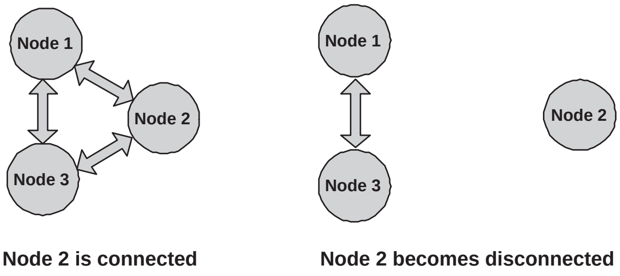

Figure 2: Node 2 periodically sends status packets, becomes disconnected

2.2 Store-and-Forward Feature

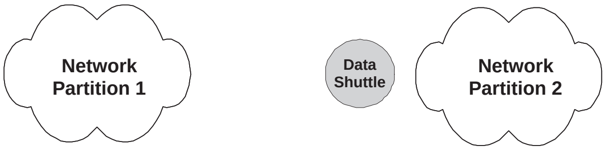

Figure 3: Store-and-forward with data shuttle

Within this project, a store-and-forward capability was designed, in which two unconnected network partitions intermittently communicate by using messenger nodes that travel between the two partitions. These intermediary nodes store the information while connected to one network, travel to the other partitioned network, and forward the stored information when they arrive within range of the second network. Figure 3 presents the basic scenario for Store-Forward Routing. The Voilà Protocol [9, 10], described as a unique solution to connect partitioned ad hoc networks, shares many similar goals to the Store-Forward extension to AUSNet. Voilà was designed to function along a beach using a standard 802.11 wireless LAN between multiple tourist hotspots. Voilà used the fact that people often congregate around certain spots to allow for controlled broadcasting of data to apparently mobile nodes. AUSNet's extensions while also designed to enable communication between partitioned networks has taken a different design choice. In summary, compared with AUSNet the main difference lies in the carrier selection. While any mobile node can potentially be a messenger in Voilà, within the AUSNet protocol a designated "Data Shuttle" AUV that exists mainly to connect partitioned networks, is used.

To enable new Store Forward options in future, a generic interface to various Store Forward models was designed. Additionally, two initial models where chosen for implementation. One based on a greedy choice, and the second around a method of Higher Level Route Specification. Design rationale and specifics are presented in Section 4.

3. RELATED WORK

A lot of work has been done in Store-Forward Routing in the area of distributed sensor networks, where a mobile node, often an Unmanned Ground Vehicle (UGV), acts as a Data Shuttle in a “Delay Tolerant Network.” [6] The Delay Tolerant Network Research Group [7] is a focal point for this effort. In the underwater domain, the Woods Hole Oceanographic Institute’s Arctic Group’s workshop on “Arctic Observing Based on Ice-Tethered Platforms” discussed scenarios where many sensors acted as store-forward hubs that divulge their data to passing AUVs [8]. The majority of these approaches, however, view the Data Shuttle as a sink to collect sensor data and then offload the data to an out-of-situation endpoint, rather than as a means to provide bidirectional communications between distributed devices.

The Voilà Protocol [9, 10] is a serious attempt to provide this functionality. Through the selection of carrier nodes, Voilà attempts to send message to nodes located at the edge of network partitions and relies on those edge nodes to transfer the message to the destination.

Finally, Li and Rus [11] use mobile modes whose trajectory can be altered, in conjunction with *a priori* distributed information about the speed and trajectory of all involved nodes. While this presents interesting options, the assumption that a networking protocol can alter the motion of a vehicle is unrealistic in typical AUV deployment scenarios.

4. METHODOLOGY AND SOLUTIONS

Within the scope of this research, both proposed solutions were implemented and evaluated using simulation. A detailed description of our proposed solutions and motivations behind our approach for each problem is presented in Sections 4.1 and 4.2 for Packet Queuing and Store-Forward respectively. All the work within this project was tested in a time-driven simulation via the Mobile Ad Hoc Undersea Network Simulator that was developed while designing AUSNet. The simulator is described in Section 4.3.

4.1 Proposed Packet Queuing Algorithm

To mitigate the accumulation of stale status packets in the queues of AUSNet nodes, we propose a “Replace” and “Hold” model of packet expiration. Additionally, the ability to deal with other expiration models is desirable and consequently there needs to be support for them within the protocol architecture. This solution was arrived at after considering two fundamental packet types: a) those which are replaceable over time, and b) those that are not replaceable. Packets that can be replaced later by more up-to-date versions of themselves are replaced as needed while those set to hold remain.

Several minor concerns arise regarding this approach based mainly on the amount of processing time that is required to handle packet replacements. From a typical outlook, if all packets are inserted in order into this queue, the replacement would require at a minimum $O(lg(n))$ per insertion. With a large collection of packets this would significantly increase the handling time of an incoming packet, to determine if it should replace an earlier packet or simply inserted at the end of the queue.

A numerical example illustrates this concern. AUV systems currently have packet generation rates of one status packet every minute or less. Conceivably, the arrival rate of individual packets is limited by about 100ms of transmission time for an 80-byte packet and the propagation delay based on the speed of sound in water. If two nodes are placed 500 m apart, the transmission of an 80-byte packet would require 1133 ms of combined transmission and propagation time between the two nodes. On this time scale required to generate a packet, and the fact that the channel is limited to the transmission of one packet at a time, the required insertion time is negligible in comparison, even considering the low clock speeds of the embedded processors used in AUV Systems. Therefore, the overhead associated with these insertions and comparisons is insignificant when compared to the overhead caused by the medium. To revisit Section 2.1, as Node 2 transitions from its connected state, gets disconnected and then reconnected, it is inserting a packet every 30 seconds. Using our proposed ”Replace” and ”Hold” strategy, node 2’s queue does not grow since the previous packets are removed.

4.2 Proposed Store-Forward Approach

Two Store and Forward models were implemented: a simple greedy solution, and a higher-level decision based solution. The greedy solution requires the inclusion of state information. As was previously mentioned, AUSNet has an optional routing method known as Prediction Based Routing, whereby the location and state information is relayed to all communicating nodes and a three dimensional real time model of the network state is maintained by the routing protocol. This network state model is used to make optimal routing solutions at a given time. The Greedy Store-Forward model uses this information to determine at a given time, which node in the same partition as the source node, is closest to a packet’s destination node. Periodically, a node will check the status of packets in its store and forward queue, along with its model of the state of the network. For a given stored packet, if another node located nearby is deemed to be closer to that packet’s destination than the current host node, then the packet will be routed to the closer node. This option is mainly limited by the distribution of changing node velocity and location information in a disjoint network.

The second implemented Store and forward model is that of Higher Level Control. This option is designed for use in a network of AUV’s operating under a high-level autonomous controller such as the DIstributed Controller Environment (DICE) [12] that designates a specific node as the Data Shuttle between two disjoint groups. DICE uses a group behavior model that decides what nodes would accomplish a given task best based on a set of variables. Shuttling data is another task to be designated amongst nodes, for which the most suitable node under a set of rules can be chosen. A higher level application uses its ability to plan and command nodes such that a specific node can be tasked (in addition to its other goals) with the role of being the data shuttle between network portions. All nodes are then informed of three timestamps, depending on their location within the network. For nodes that are part of the destination of the data shuttle 1) the time given is the time at which the specified data shuttle will arrive at their location 2) the time the data shuttle will no longer be in communication, and 3) the time the shuttle will next be back in in communication with the source partition. Oppositely, nodes whom the data shuttle is departing are informed first 1) when the shuttle will disconnect with them 2) when the shuttle will arrive at the opposite partition, and 3) when the shuttle will next arrive at their location. This information is used to determine the time required for the packet to wait for the arrival of the data shuttle, if necessary, and secondly the time the packet will need to wait on the data shuttle.

Within each system, higher level control dictates the use of specific intermediary nodes as gateways in the path, similar to Loose Source Routing in wired networks [13]. This leaves the sending node with only the responsibility of transmitting the packet as far as the gateway. Once the packet arrives, the remaining routing to the destination is the responsibility of the gateway. These approaches differ in that while Loose Source Routing merely specifies the gateway, High Level Control also specifies a specific amount of time for the packet to wait before rerouting and retransmission to allow the data shuttle to arrive at the intended network partition. This is implemented as a list of wait times per hop in a network route such that the packet will be held in a queue for a specified period of time before being transmitted again to the next hop in the network path.

The design of a Higher Level decision maker within the simulator, to simulate the possibilities for an existing vehicle controller, was a challenging task. The calculation of a node's time of reconnection and disconnection to a network segment involves the following vector based mathematics.

Given two AUVs A and B with velocities $\mathbf{v}_a$ and $\mathbf{v}_b$ and initial positions of $\mathbf{A}_0$ and $\mathbf{B}_0$ the vector functions of position for $\mathbf{A}$ and $\mathbf{B}$ are as follows:

$$\mathbf{A} = \mathbf{V}_a t + \mathbf{A}_0$$

$$\mathbf{B} = \mathbf{V}_b t + \mathbf{B}_0$$

And the distance between A and B is defined as the length of their vector difference:

$$D(\mathbf{A}, \mathbf{B}) = |(\mathbf{A} - \mathbf{B})|$$

Using this information, we will derive the time of connection (t_c) and disconnection (t_d) for a pair of nodes. So

$D(\mathbf{A}, \mathbf{B}) = |(\mathbf{V}_a t + \mathbf{A}_0) - (\mathbf{V}_b t + \mathbf{B}_0)|$

$D(\mathbf{A}, \mathbf{B}) = \sqrt{((\mathbf{V}_a - \mathbf{V}_b)t + (\mathbf{A}_0 - \mathbf{B}_0)) \cdot ((\mathbf{V}_a - \mathbf{V}_b)t + (\mathbf{A}_0 - \mathbf{B}_0))}$

For visual ease, $\mathbf{C} = (\mathbf{V}_a - \mathbf{V}_b)$ and $\mathbf{E} = (\mathbf{A}_0 - \mathbf{B}_0)$. Hence:

$$D(\mathbf{A}, \mathbf{B}) = \sqrt{(\mathbf{C}t + \mathbf{E}) \cdot (\mathbf{C}t + \mathbf{E})}$$

$$D(\mathbf{A}, \mathbf{B}) = \sqrt{t^2 \mathbf{C} \cdot \mathbf{C} + 2t\mathbf{C} \cdot \mathbf{E} + \mathbf{E} \cdot \mathbf{E}}$$

$$D(\mathbf{A}, \mathbf{B})^2 = t^2 \mathbf{C} \cdot \mathbf{C} + 2\mathbf{C} \cdot \mathbf{E}t + \mathbf{E} \cdot \mathbf{E}$$

$$0 = t^2 \mathbf{C} \cdot \mathbf{C} + 2\mathbf{C} \cdot \mathbf{E}t + \mathbf{E} \cdot \mathbf{E} - D(\mathbf{A}, \mathbf{B})^2$$

Since we want to determine the time at which the distance between the nodes is equal to the transmission range, $D(\mathbf{A}, \mathbf{B}) = R_t$

$$0 = (\mathbf{C}) \cdot (\mathbf{C})t^2 + 2(\mathbf{E}) \cdot (\mathbf{C})t + (\mathbf{E}) \cdot (\mathbf{E}) - R_t^2$$

Applying the quadratic formula here presents t_c and t_d if $\mathbf{V}_a$ and $\mathbf{V}_b$ remain constant.

$$t_c = \frac{-2\mathbf{E} \cdot \mathbf{C} - \sqrt{(2\mathbf{E} \cdot \mathbf{C})^2 - 4\mathbf{C} \cdot \mathbf{C}(\mathbf{E} \cdot \mathbf{E} - R_t^2)}}{2\mathbf{C} \cdot \mathbf{C}}$$

$$t_d = \frac{-2\mathbf{E} \cdot \mathbf{C} + \sqrt{(2\mathbf{E} \cdot \mathbf{C})^2 - 4\mathbf{C} \cdot \mathbf{C}(\mathbf{E} \cdot \mathbf{E} - R_t^2)}}{2\mathbf{C} \cdot \mathbf{C}}$$

This methodology can be used in a network that does use the Prediction Based Routing component of AUSNet. In this case, a path for the packet could be discovered upon arrival at the reconnection point through standard DSR active route seeking methods. This method could be used to define a small time window in which shuttle connectivity is assumed for a node and all queued packets are sent.

4.3 Mobile Ad Hoc Undersea Network Simulator

To facilitate the design of AUSNet, and to enable rapid testing of alteration to the code base, the Mobile Ad Hoc Undersea Network Simulator was developed by Matthew Haag at TSI. The simulator is a time-driven two-dimensional environment designed to model the propagation of messages to mobile nodes through water. Nodes move through a set of waypoints that can be loaded from an initial configuration file or added on the fly. The simulator provides the means for saving configurations of node locations and waypoints as well as the ability to store network performance statistics. Conceptually, the simulator treats each node as a point in a two dimensional world, and treats each message as an expanding circle. The simulator is designed to function at approximately 30 frames per second, and at various simulation times the calculation of the location of each node and message is accomplished using the time offset from the previous frame. This enables the simulator to be sped up or slowed down significantly based on user input to focus on the interaction of an individual packet, or to run for an extended time to generate statistical data. The simulator can also transmit random messages between specified nodes for a given period of time at a desired rate, and to manipulate the physical constants of the environment to match desired simulation conditions.

5. SIMULATION RESULTS

5.1 Results of Packet Queuing

Two scenarios were used to exercise the solutions to the Packet Queuing problem. The first, or "Normal" scenario, was undertaken to examine the effects of using "Replace" in a standard network scenario; one that had been previously used to examine the effectiveness of AUSNet. Within this scenario the network remains connected constantly. The second scenario is similar to the real world situation in which the packet queuing concern was first raised. The "Narragansett" scenario is a simulation of the Narragansett Bay test run for the SAUV and the MARV where the MARV was disconnected from the network and spent significant time queuing up packets. Each of these scenarios were simulated many times adjusting for various AUSNet settings, altering the state of the various DSR-based configuration parameters for broken route recognition. The default AUSNet settings are that the protocol will retransmit a packet at most 5 times before assuming that the route hop being attempted is no longer functional. Further, by default, AUSNet will wait for 30 seconds between retransmission attempts. These values define how DSR diagnoses broken routes and reacts to discover new routes.

The implementation of the Replace flag for simulated status packet transmissions from AUVs showed marked improvement not only in the case of node isolation within a disjoint network, but also for normal operation in a connected network.

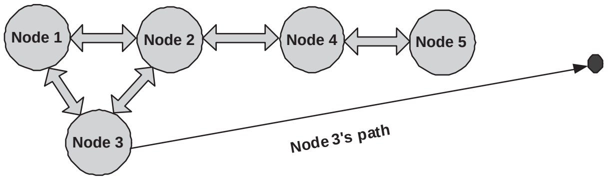

Figure 4: Simulated AUSNet testing scenario

In AUSNet, all packets that are queued in any location, waiting for a route, waiting for transmission, or waiting for confirmation, are processed using the replacement model system. This leads to significantly improved bandwidth utilization even within non-disjoint network systems. Specifically, this prevents the build up of non-critical traffic after a network change has happened and prior to its detection by the Dynamic Source Routing mechanisms. Using a previously presented AUSNet testing scenario, a network of 5 nodes starts in Figure 4 with node 3 in contact with nodes 1 and 2. Node 3 then travels slowly towards the endpoint of the arrow attempting to transmit a simulated status packet every 30 seconds. Along its travel, node 3 will break connections along each step forcing packets to be enqueued before the route is rediscovered.

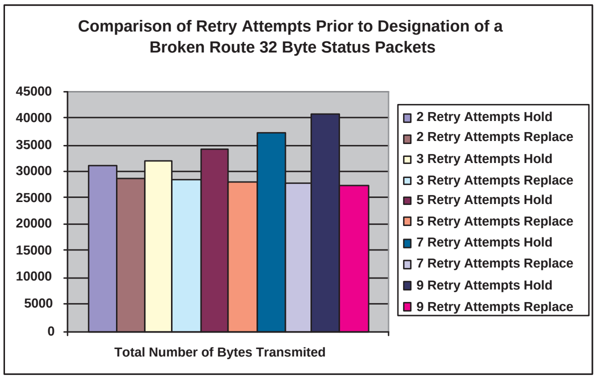

Figure 5: Results for replace-hold simulation

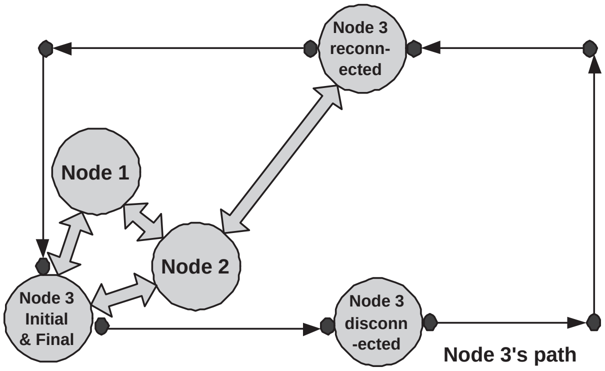

Figure 6: Simulated version of in-water scenario requiring replace-hold solution

As Figure 5 shows, the use of Replace flags yields a significant 18% improvement in network bandwidth utilization with standard retry settings and 32-byte data packets. The more sensitive a set of network settings is to a change in connection topology, the less improvement is shown from the use of Replace vs. Hold.

As was discussed in Section 2.1, the discovery of the Packet Queuing problem came from an in-water operation where a node became disconnected from the AUSNet network, and on reconnection it used too much of the available bandwidth transmitting status packets. This situation was recreated within the Mobile Ad Hoc Undersea Network Simulator using conditions to mimic the original run, as depicted in Figure 6. Within the scenario, node 3 moves in a large 6000 m by 2000 m simulated box while nodes 1 and 2 remain stationary inside the box. For approximately half of the time, node 3 is unable to communicate with nodes 2 or 1 despite the fact that it is attempting to send a status packet to node 1 every thirty seconds. Clearly, the advantages of the Replace method versus the Hold method is expected to become evident here. The long period of node isolation shows exactly how useful this method can be. Replace queue handling showed a marked near 50% reduction in utilized bandwidth over the use of the Hold method across the board with respect to all applicable variables. Specifically, with 64-byte status packets as shown in Figure 7, standard AUSNet Configuration yields a 48% improvement.

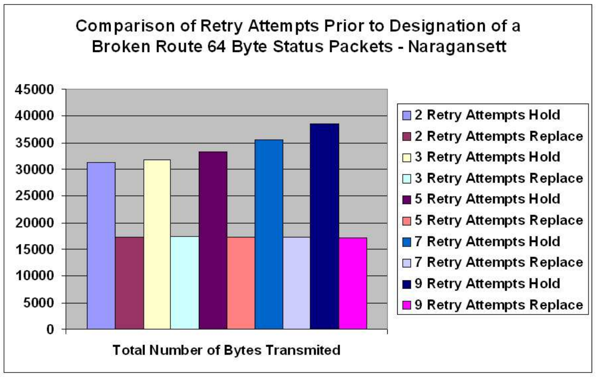

Figure 7: Simulation results for replace-hold scenario shown in Figure 6

5.2 Results of Store-Forward

Store-Forward testing was done with two similar scenarios, where a group of nodes initially start connected, and then become disjoint, with a single node acting as a data shuttle between the partitions. The only difference between the two is that in the first "static" scenario, once the initial disjunction has occurred, only the data shuttle remains mobile. In the "dynamic" scenario, once the initial disjunction has occurred, several nodes in each partition follow a simple mobility pattern, similar to a repetitive search. Much of the functionality of Store-Forward exists as a proof of concept that in AUSNet, nodes can be selected and used as Store-Forward routers. The graphs presented here are representative of the differing approaches, however, they are limited in that the best baseline case to be provided is the Greedy method. The graphs are generated to examine the effect of time and message transmission on the methods. Within the scenario, one node in each partition attempts to transmit to a node in the opposite partition once every 150 seconds. Each method is used with varying success.

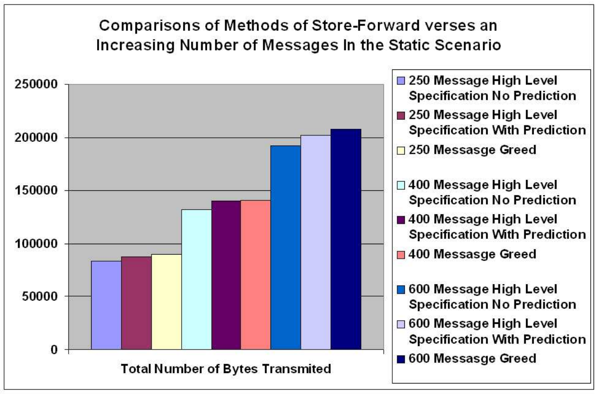

Figure 8: Store-and-forward methods vs. number of messages (static scenario)

In the static scenario, as Figure 8 shows, the non-predictive, High Level Control model fares the best in terms of overall bandwidth usage. All methods, however, enjoy nearly perfect message delivery due to the stable routes.

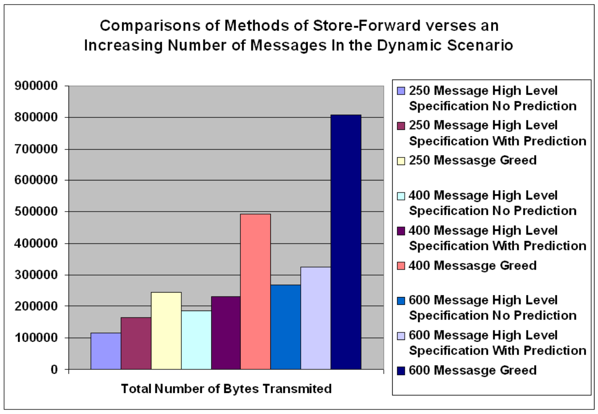

Figure 9: Bytes transmitted by store-and-forward methods (dynamic scenario)

Additionally, due to the lack of movement, the state information required for prediction-based solutions is useless overhead, leading to inefficiency for both models using it. Again, it should be noted that only the least efficient Greedy model will function without oversight from a higher level program. Both of the High Level Control models require upper level input.

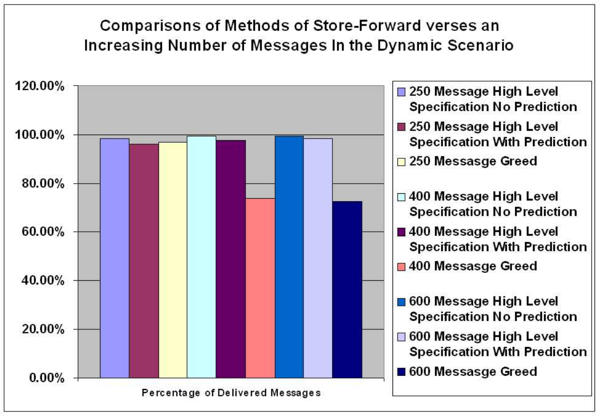

Figure 10: Percentage of delivered messages by store-and-forward (dynamic scenario)

The dynamic scenario demonstrated a significant departure from the static. Clearly, the addition of dynamic elements give rise to significantly higher overhead required to properly operate the network, both with respect to the distribution of state information for prediction and with respect to the DSR-based route maintenance that would be required. However, what was not expected was the growing cost associated with the Greedy model as more packets were sent over time, as shown in Figure 9. In addition, there was a significant decrease in the performance of properly delivered packets. This is shown in Figure 10 with the drastic reduction as time passes of the delivery percentage, falling to less than 70% with 600 messages. This behavior turns out to be specific to the prediction model's appliction to Store-Forward, not the Greedy method itself.

6. DISCUSSION

Packet Queuing Problem: The introduction of the "Replace" algorithm into AUSNet packet queuing led to significant reduction of bandwidth usage in both scenarios. This is interesting because the algorithm was designed not to lower bandwidth usage for a normally connected network, and instead was simply intended to eliminate the barrage of transmission of useless information that would occur when a node is disconnected from a network. This improvement was found in a normal connected network because the pace at which the replaceable data was generated, and would be typically generated by real AUVs, exceeds the time required to determine a route to be broken. Through variation of the key factors to determine broken routes within DSR, it was shown that as the time to determine a broken route increases, the value of "Replace" increases as well.

Store-Forward Routing: With both Store-Forward models, as the network continues to operate, the disjoint partitions have an increasingly inaccurate model of the location of the opposing partitions. This is due to the Dead Reckoning based model in AUSNet, which assumes that a node travels with a fixed speed and direction between updates. When two networks are portioned, the nodes do not receive updates from other partitions. Ideally, the data shuttle could be configured to store a reasonable amount of location information updates so that when the shuttle arrives at the opposite partition, this information could be disseminated. While this seems to be a simple conceptual approach, it lacks several subtleties. First, with the Greedy methodology, there is no simple "Data Shuttle". Instead, all nodes act as data shuttles as the need arises. Hence no specific node can easily be placed with the burden of informing the partitions. Secondly, if a node is moving parallel to a data shuttle at the time of the shuttle's disconnection from the network's partition, the shuttle will continue to assume the node is traveling along that path as it does not receive a subsequent update if the node alters course. In general this exposes a weakness not in the Greedy method of data shuttling, but rather within the distribution of information for Prediction Based Routing in general.

Resolving this concern requires an analysis of the design goals of the Prediction Based Routing system in the context of the underwater acoustic medium. Many AUV systems, utilize way-point navigation. Despite the prevalence of such navigation methods, AUSNet's PBR capability exists solely in terms of initial position and velocity.

Much of the problem specified here with respect to location models applies to the use of prediction based routing as a component of the High Level Control model as well. When the specified shuttle arrives at the location where it is expected to disperse the received messages, its Dead Reckoning model is quite out of date. This leads to poor route choices due to inaccurate predictions. The use of poor routes in many cases is more costly with respect to used bandwidth than the use of DSR-based discovery to find proper routes. However, this is partially an artifact of the specific simulation scenarios. There was little inter-partition communication in either of the examples, as they were designed to test the extra-partition functionality. The use of both tactics to improve the Greedy model also has application here, notably to improve prediction data.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have examined the packet queuing and store and forward routing issues as they relate to AUV networking. It has been shown that significant improvement can be gained through the use of expiration of stale data. Furthermore, two different methods of transmitting data between partitioned networks have been proposed and evaluated with varying success. The use of packet replacement in all possible applications will result in marked reduction of unneeded network traffic when ever the network topology alters and the reactive DSR based component of AUSNet needs to re-establish a proper route between hosts. This applies not only to the Narragansett Bay scenario, where a node is temporarily disconnected from the network, but also in many other typical ad hoc network scenarios, provided that an alteration of network linkage occurs.

Several opportunities exist to build upon these stale packet expiration schemes. This includes the possibility of incorporating a time-based expiration into the scheme as an additional expiration model. This would allow a packet to be held for a period of time after which it should be assumed to be stale and removed. There are several subtle issues involved in this solution because expiration models need to work on all nodes, not simply the originator. For example, to conserve bandwidth, there is no timestamp in AUSNet.

The two Store-Forward models used to span partitioned networks, Greedy and High Level Control, each performed with varying success. The first conclusion is that there are problems with respect to the Greedy model, and its data shuttle choices. Second, and much in the same vein, the usefulness with respect to the chosen scenarios of Prediction Based Routing (PBR) is called into question.

An investigation of how state information can be best spread in the dynamic and unconnected state of AUV networks would be interesting. The most promising suggestion is the use of an optional waypoint-based system of state information rather than the simple position and velocity based system now incorporated within AUSNet's prediction based routing scheme. On top of that, additional future work would involve the use of data shuttles to also shuttle some semblance of position-based data, to enable all network segments to have moderately accurate predictions of node locations of nodes in other segments. Finally, these algorithms shall continue to be tested within AUVs situated in the field to validate simulation results, and build their credibility within the AUV user community.

8. REFERENCES

[1] C. Benton et al. "Autonomous Systems Network (AUSNet) - Protocols to Support Ad Hoc AUV Communications," in Proc. IEEE/OES AUV2004: A Workshop on Multiple Autonomous Vehicle Operations, June 2004.

[2] J. Broch, D. Johnson, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet Draft, July 2004.

[3] C. Benton et al. "Advancements Within the AUSNet Protocol," in Proc. Int'l Symp. Unmanned Untethered Submersible Technology (UUST05). Aug. 2005.

[4] S. Mupparapu et al., "Performance Evaluation of Ad Hoc Protocols For Underwater Networks," in Proc. Int'l Symp. Unman. Unteth. Sub. Tech. (UUST05), 2005.

[5] R. Shah et al. "Voilà: Delivering Messages Across Partitioned Ad Hoc Networks," in Proc. IEEE WLN 2004.

[6] M. Ho and K. Fall. "Delay Tolerant Networking for Sensor Networks," Poster, IEEE SECON 2004.

[7] Delay Tolerant Network Research Group, `http:\\www.dtnrg.org`.

[8] A. Proshutinsky et al. "Ice-Based Observatories: A strategy for improved understanding of the Arctic atmosphere-ice-ocean environment within the context of an Integrated Arctic Observing System," Woods Hole Oceanographic Institute, June 2004.

[9] R. Shah and N. C. Hutchinson "Delivering Messages in Disconnected Mobile Ad Hoc Networks," in Proc. ADHOC-NOW 2003.

[10] R. Shah, "Delivering Messages in Disconnected Mobile Ad Hoc Networks," Master's Thesis, Univ. of British Columbia, 2004.

[11] Q. Li and D. Rus, "Sending Messages to Mobile Users in Disconnected Ad Hoc Networks," in Proc. ACM/IEEE Mobicom 2000.

[12] C. Duarte et al., "A Common Control Language to Support Multiple Cooperating AUVs," in Proc. Int'l Symp. Unman. Unteth. Sub. Tech. (UUST05), 2005.

[13] "Internet Protocol," IETF RFC 791, Internet Protocol Specification - Loose Source and Record Route.