A Data Link Layer in Support of Swarming of Autonomous Underwater Vehicles

Daladier Jabba M.
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
Email: daladier@cse.usf.edu

Miguel A. Labrador
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
Email: labrador@cse.usf.edu

Abstract— **Swarms of underwater autonomous vehicles (UAVs) are being increasingly used in civilian and military applications. Formations of UAVs are being utilized to detect mine-like objects and oilfields, collect water quality information, detect intruders, and in navigation assistance applications, among others. This paper introduces a new data link layer for underwater communications suitable for non-synchronized ad-hoc networks using multiple frequency channels that supports linear and polygon formations of UAVs. At the MAC layer, the 2MAC protocol is introduced. 2MAC is a contention-based and collision avoidance scheme designed to work on top of an OFDMA-based acoustic channel at the physical layer and coordinate the simultaneous transmissions of multiple OFDMA sub-channels simultaneously with only one transceiver. This 2MAC-OFDAM multichannel transmission solves the hidden, exposed terminal, capture, and deafness problems commonly found in ad-hoc networks and also increases the network throughput and packet delivery rate considerably. In addition, since the nodes are only equipped with one transceiver, the design preserves the low cost advantage of current single-channel single-transceiver CSMA-based nodes. In addition, 2MAC includes an improvement of the Binary Exponential Backoff algorithm (BEB), more adequate for underwater channels. At the logical link layer, we present SW-MER, a new protocol based on Stop and Wait and Sliding Window approaches that includes an exponential retransmission strategy to achieve better channel utilization and high reliability. A performance evaluation is carried out using the well-known Bernoulli error model and also a more realistic underwater channel simulated using synthetic traces that correspond to a shallow water channel. Simulation results in terms of reliability and throughput efficiency demonstrate the superior performance of the new data link protocol when compared with the current CSMA/CA protocol at the mac sublayer or traditional Stop and Wait protocols at the logical link control sublayer.**

I. INTRODUCTION

Undersea exploration in military tactical and civilian applications using swarms of underwater autonomous vehicles (UAVs) is on the rise. Mine-like object detection, shore surveillance, intrusion detection, water quality monitoring, environmental data collection, and navigation assistance are among the many possible applications. One common aspect of these applications is the need of constant communication among the UAVs. The success of the mission highly depends on the ability of the UAVs to exchange both, application and swarm formation-related information at all times, not a trivial task using an underwater communication channel. The underwater channel is not precisely the best channel. For example, common radio frequency (RF) signals utilized in wireless communications cannot be used underwater, as electromagnetic waves attenuate very rapidly with distance, restricting the communication to only a few meters. In order to overcome this limitation, acoustic communications has been the method of choice for underwater communications.

However, acoustic communications has its own limitations as well. For example, the speed of sound is not constant underwater; it varies with depth, salinity, and other factors. Further, the speed of sound is five orders of magnitude lower than the speed of light, meaning that underwater acoustic communications experience very large propagation delays, even in short distances. In addition, the underwater communication channel is very noisy, and the bit error rate (BER) also changes depending on the depth and other factors. Normally the errors are correlated, like in wireless communications, but fading is deeper and longer underwater producing more and burstier errors. Further, as in wireless communications, transducers for acoustic communications are half duplex, precluding the application of well-known collision detection mechanisms in MAC layer protocols and negating the possibility of using more efficient flow and congestion control mechanisms at the logical link control layer. As a result, most data link layers use the simple but inefficient Stop and Wait protocol instead of a sliding window-based approach at the logical control sublayer.

In addition, applications with swarms of UAVs bring more problems to the already poor communication channel. Well-known problems in wireless mobile ad hoc networks, such as the hidden and exposed terminal problems and the capture and deafness problems that need to be solved at the mac sublayer, also occur here. All these factors combined make the performance of acoustic communications very poor and justify the need for new technologies and protocols better suited for underwater communications.

This paper proposes a new data link layer combined with Orthogonal Frequency Division Multiple Access (OFDMA) [1] technology in order to address most

---

[1] Professor on leave from Universidad del Norte, Barranquilla, Colombia. www.uninorte.edu.co.

of the problems mentioned before. OFDMA is a multiple access/multiplexing scheme that offers multiplexing operations over a single frequency band splitting the corresponding channel in several sub-channels. Therefore, with OFDMA nodes have the advantage of having several channels with a single transceiver. At the data link layer, a new MAC and logical link control sublayers are introduced. The proposed MAC protocol, called 2MAC, which is based on the well-known MACAW protocol [2], has been specifically designed to support linear and polygon formations of UAVs using OFDMA technology at the physical layer. 2MAC takes advantage of OFDMA's multiplexing capability to reduce collisions, eliminate the hidden, exposed, capture, and deafness problems, and improve network performance transmitting to different nodes simultaneously. At the logical link control sublayer, the proposed SW-MER protocol includes an exponential retransmission strategy that provides a highly reliable service over high error prone channels and a sliding window mechanism to increase the channel utilization under long propagation delay channels. Further, the protocol introduces a new backoff retransmission policy more adequate for long propagation delay channels, which is also meant to improve the channel utilization.

The new data link layer is evaluated via simulations using the well-known Bernoulli error model and also a more realistic underwater channel model simulated using synthetic traces that correspond to moving nodes in shallow water. A Hidden Markov Model and the well-known Baum-Welch algorithm are used to derive a two-state Markov model to introduce the errors in the simulations. Our results indicate the superiority of 2MAC and SW-MER in terms of throughput and reliability compared with well-known MAC and logical link control protocols.

The remainder of the paper is organized as follows. Section II includes a description of some traditional MAC and logical control protocols used in underwater communications. Section IV describes the 2MAC protocol and Section V presents the SW-MER protocol. Section VI describes the underwater channel trace and the two-state Markov error model. Section VII includes the performance evaluation of 2MAC and SW-MER, including other well-known MAC and logical link protocols for the underwater environment. Finally, Section VIII concludes the paper and presents directions for future research.

II. RELATED WORK

Most underwater MAC protocols have been designed following the contention-based approach, which includes random access and collision avoidance methods. In order to avoid collisions and reduce the hidden terminal problem, most random-based protocols use the RTS (Request To Send) and CTS (Clear To Send) signals to capture the media before starting the data transmission. Examples of protocol in this category can be found are MACA [3], MACAW [2], FAMA [4], Slotted FAMA [5], and others. However, these protocols are not well suited for underwater communications; adjustments need to be made to increase the performance of these protocols under long propagation and error prone channels.

In [6], the MACA protocol has been adapted for underwater networks by adjusting the time required to exchange control packets considering the distance between sender and receiver (defined as a minimum handshake length). Once a node receive an RTS, it immediately sends the CTS to the corresponding transmitter, and then waits for the data packet. If during this period the receiver hears an RTS from another node, it sends a short warning packet to the node it sent the CTS before, to prevent packet collisions. At the sender side, once the CTS is received the node also waits for some time before starting the data transmission. If another CTS or warning is received from the receiver, the sender aborts the transmission. Although the control packet exchanges are reduced, still large packet collisions can be presented in dense networks.

The protocol for underwater sensor networks presented in [7] is a multi-channel scheme based on the MACA protocol in [8] to improve the network efficiency using CDMA. RTS/CTS/DATA packets are also used but work with more than one channel; one channel for control packets (common code) and the others for data packet transmissions (different spreading codes). All nodes in the network are assigned to the same common channel (common code) and the common code is monitored for any packet arrival. Once the source sends the RTS and receives the CTS, the optimal spreading code (from the rest of the spreading codes) in which the data packet is going to be transmitted is chosen. After receiving the data packet, the destination decodes the received signal and retrieves the data, and at the end, the destination node will send an ACK packet to the transmitter. One of the disadvantages of this protocol is the centralized nature of the CDMA scheme.

The modified MAC protocol proposed in [9] also works with multiple channels. The communication process is divided in four stages: carrier sensing, transmission frame, receiving frame, and error control. In the carrier sensing process, flags are set to '0' for a that channel if the carrier is detected, meaning that the channel is busy. The transmission frame process starts by requesting one channel among the available ones. A procedure for searching the candidate channels is executed and as a result, the channel with the longest idle time is selected for the transmission. Once the frames arrives, either an ACK or NACK is sent.

At the logical link control sublayer, after the basic Stop and Wait protocol was introduced in [10], several Stop and Wait (SW) variants have been proposed, such as those included in [11], [12], and [13]. In the stop and wait variant proposed in [11], the sender transmits new packets one at a time. However, in case of a missing packet or a packet in error, the protocol retransmits a window of i packets in which the missing packet is repeated i times. This procedure is repeated until the transmitter either receives the acknowledgment or the maximum number of retries is reached. At that time, the transmitter is allowed to send a new single packet. This mechanism is expected to increase the reliability of the protocol at the expense of a poorer channel utilization.

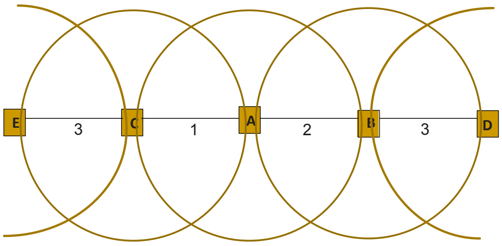


Fig. 1. Linear topology.

The stop and wait variant proposed in [12] works transmitting a window of m packets all the time. The transmitter sends m packets in every transmission opportunity and waits for the corresponding acknowledgments, which are sent by the receiver in one packet that includes the group information. After receiving the packet with the acknowledgments, the transmitter sends a new window of m packets filled with those packets from past windows received in error, and new packets if space in the new window is available. This protocol, which transmits m packets every round trip time, is expected to provide good reliability while utilizing the channel better than the one proposed in [11].

The stop and wait version described in [13] is very similar to the one just described, as it also works with a window of packets. However, upon receiving the acknowledgment packet, the transmitter sends a new window containing just the packets received in error. Once these packets are correctly received, the sender sends a new window with m new packets. This protocol has the advantage of requiring less buffer space at the receiver in order to guarantee the ordered delivery of packets to the higher layer.

Although these previous LLC protocols improve the throughput efficiency of the network, they still have performance issues when using high error prone and long propagation delay channels.

III. NETWORK TOPOLOGY AND CHANNEL ASSIGNMENT

As mentioned before, there are issues in underwater MAC protocols such as hidden and exposed terminal problems, and the capture and deafness problems, that combined with the characteristics if the underwater channel make the performance of acoustic communications very poor. Several of these problems can be solved by using OFDMA technology with appropriate data link layer protocols. OFDMA offers the capability of splitting the physical channel into several sub-channels, and transmitting data simultaneously over those channels with only one transceiver. Using an adequate topology and channel assignment strategy, the 2MAC protocol at the data link layer, and OFDMA, simultaneous data transmissions to different neighbors can occur without collisions.

By using several sub-channels per node simultaneously, it is possible to have different topologies, especially linear or polygon topologies, and transmit information to each of the neighbors without presenting collisions. Figures 1 and 2 show a linear and a hexagon topologies with the assignment of three sub-channels that allow simultaneous transmission without interference. For example, in the hexagon topology, node A has B and F as its neighbors, it communicates with B through channel 1 and with F through channel 2. A and E can communicate with F and D respectively at the same time, without experiencing the exposed terminal problem since F does not have the same channel of A and E. Similarly, A and E can communicate with F simultaneously without colliding. In this paper we consider a linear topology of UAVs using three channels collaborating in a sea floor scanning mission.

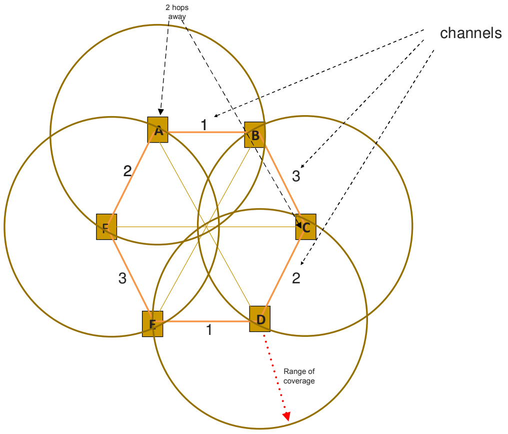


Fig. 2. Hexagon topology.

There is a relation between the number of channels in the network, two as a minimum, and the number of sides that a polygon topology can have. Using three channels in the network but two different per node, it is possible to have polygon topologies in which their number of sides is multiple of 3. The optimal number of channels required to set in a polygon topology can be obtained using Equation 1, in which S represents the number of sides that the polygon topology is desired to have and CH is the number of channels needed. The value of CH is such that n is the maximum quotient value. As an example for the hexagon topology shown in Figure 2, 3 or 6 channels can be defined, but using only 3, a maximum value n is obtained when S is divided by $CH = 3$. With 3 OFDMA sub-channels the maximum value is obtained for n, and the communication among the nodes can be effectively done. Linear topologies are also possible and they need only a 3-channel network, it does not matter how many nodes are in the linear network.

$$n = \max\left\{\frac{S}{CH}\right\} \tag{1}$$

IV. 2MAC: A CONTENTION-BASED MAC PROTOCOL FOR UNDERWATER ACOUSTIC NETWORKS

In this section, the 2MAC protocol is described in detail. General and detailed procedures are included as well as the modification of the binary exponential backoff algorithm.

A. General description

2MAC is a contention-based MAC protocol based on the MACAW protocol [2]. It has been designed to improve the performance efficiency in underwater ad hoc acoustic networks having neighbors located in different subchannels. 2MAC uses a four-way handshaking access method (RTS/CTS/DATA/ACK), a new control packet called BTS (blocked to Send), an Adjusted Response time (ARS) to wait for signals from both neighbors, and a listen/contention time to exchange data, see Figure 3. In every transmission process, M data packets will be transmitted through each channel. Short Inter-Frame Spacing (SIFS) and Distributed Inter-Frame Spacing (DIFS) are also used in the packet transmission process as in IEEE802.11.

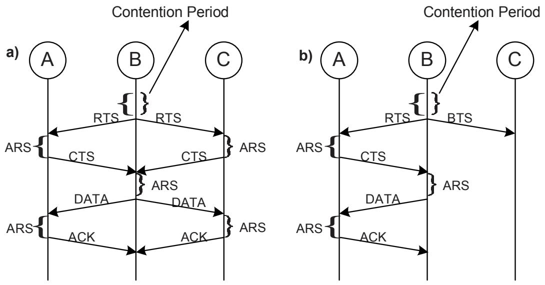


Fig. 3. 2MAC transmission process.

BTS is a control packet to inform that a channel will not be available for communications with the corresponding neighbor. Once a sender decides to start a transmission and just one channel is going to be used, the sender starts a handshaking process through this channel but at the same time transmits a BTS in the other channel to tell its other neighbor for how long the channel is going to be unavailable. This is shown in Figure 3 in which case Node A has packets for A but not for C.

B. Detailed Description

2MAC has the following ten states: Idle, Channel Assignment, Contention (Listen), Waiting for CTS, Receiving RTS, Waiting for ACK, Waiting for Data, Backoff, Adjusted Response, and Blocked to Send. Figures 4 and 5 show the 2MAC sender and receiver state machines, respectively. The states and state transitions needed in the sender and receiver process are explained next.

1) Idle state: When a node receives an RTS through one of its channels in this state, it goes to the Receiving RTS state to check if an RTS packet is coming from the other channel. It stays in the Idle state until it has packets to send when it goes to the Channel Assignments state. In the Idle state, a node can receive not only RTS but also BTS packets. If the node receives a BTS packet, it blocks the corresponding channel for the time defined in the BTS, and only the other channel is available for transmissions during that time. When a node receives a BTS packet from both channels, it goes to the Blocked to Send state.

2) Channel Assignment state: Since every node has two neighbors and each of them has a different channel assigned, two BTS fields are used to avoid collisions, one for every channel (similar usage like NAV in IEEE802.11), see Figure 14. Once packets arrive from the upper layer, their destinations are verified and the corresponding channels are selected and activated depending on their channel BTS values. If two data packets must be transmitted at the same time (one for every neighbor), both BTS are verified, otherwise only one data packet is transmitted and the corresponding channel BTS value is analyzed. A nonzero BTS value means that the channel is unavailable to send packets because that neighbor is already using it.

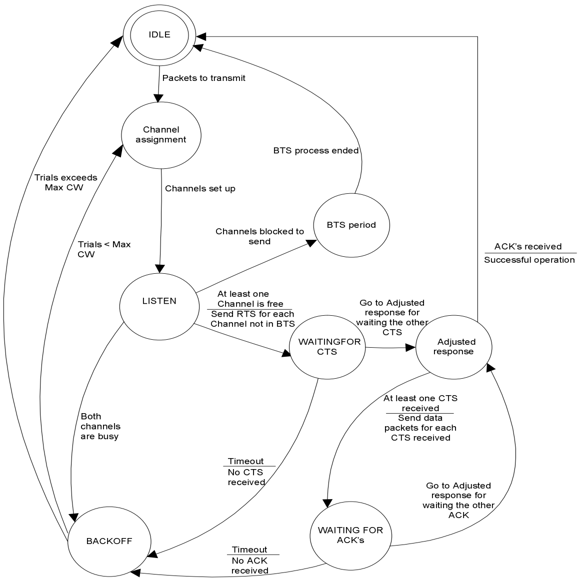


Fig. 4. 2MAC process at the sender.

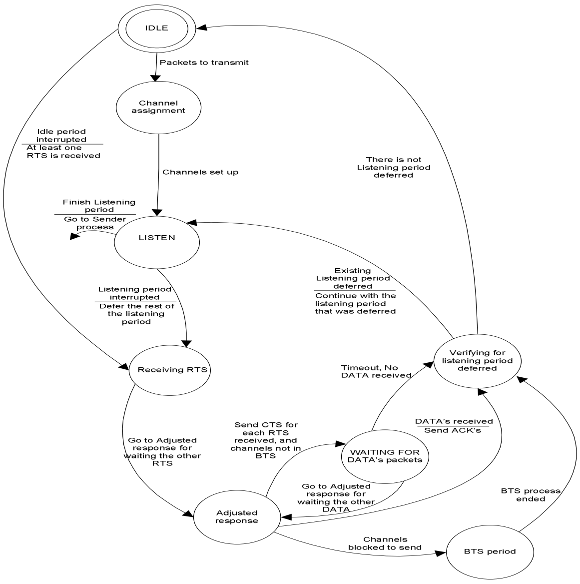


Fig. 5. 2MAC process at the receiver.

Three possibilities can happen. The first one is when two M data packets must be transmitted simultaneously in different channels and at least one of the corresponding BTS values is zero. If both BTS are zero, all channels are activated and the

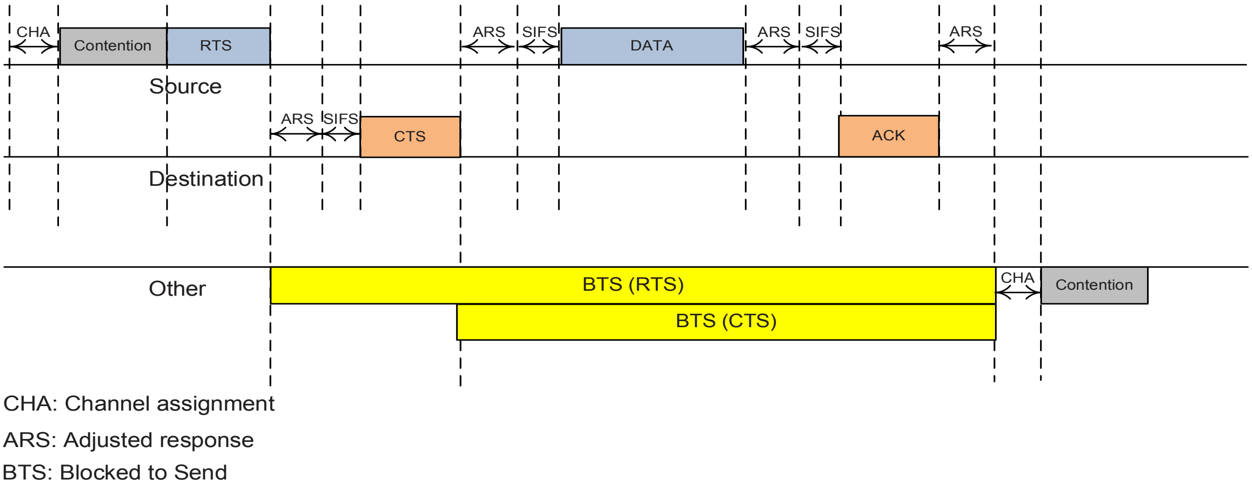


Fig. 6. 2MAC, BTS process assignment.

transmitter goes to the Contention state in both channels. If one BTS is zero, the data packet that corresponds to the blocked channel stays in the queue and waits for another transmission process, and the node goes to the Contention state only for the channel with the zero BTS value. In the case that both BTS values are nonzero, the node goes to the Blocked to Send state. Once its Blocked to Send state finishes, the node goes back again to the Channel Assignments state.

3) Contention (Listen) state: In this state the transmitter will be listening the activated channels (either both or just one) for a time equal to a round trip time of a control packet (CTS or RTS), depending on what happened in the Channel Assignments state. If the channel (or channels) are free (no communication occurs during the Contention state), the node sends RTS simultaneously to all the activated channels to start a communication process (a BTS will be send through the activated channel that is not going to do a data transmission), and goes to the Waiting for CTS state. If the channel is busy, the node defers its remaining contention time, and receives the neighbor transmission(s). Once transmission finishes, it goes back to this state to finish the listening period.

4) Waiting for CTS state: In this state a node waits until either it receives both CTS or a timeout occurs. After receiving the first CTS packet, the transmitter goes to the Adjusted Response state to wait for the second CTS arrival. If the sender does not receive the second one, it considers that a collision occurred in that neighbor and the corresponding channel is disable for data packet transmissions. When a timeout occurs, the sender enters the Backoff state for certain amount of time. Once the Adjusted Response state finishes, the transmitter sends the data packets through the corresponding activated channels and goes to the Waiting for ACK state.

5) Receiving RTS state: If a node receives an RTS while being in the Contention state, it defers its remaining time until the communication process finishes. In the Receiving RTS state, when the first RTS is received the node goes to the Adjusted Response state to verify that another RTS is coming from the other neighbor. If two RTS are received, a CTS is transmitted through its activated channels simultaneously, otherwise just one CTS is sent to answer the RTS, and a BTS is transmitted to its other neighbor carrying the time it will be busy. Once the CTS are transmitted, the node goes to the Waiting for Data state.

6) Waiting for ACK state: In this state the node waits until either it receives both ACK, a timeout occurs, or just one ACK arrived. After receiving the first ACK packet, if only one channel is activated, the transmitter finishes its transmission process, channels are set up to the available mode, and the node goes to the Idle state. If after receiving the first ACK packet both channels were activated, the transmitter goes to the Adjusted Response state to wait for the second ACK. If the sender does not receive the second ACK, it considers that a collision occurred in that neighbor, its corresponding data packet will be retransmitted in a next transmission process, and the channels are set as available and the node goes to the Idle state. When a timeout occurs, the sender goes to a Backoff state for certain amount of time.

7) Waiting for Data state: Once the first Data packet is received, the receiver goes to the Adjusted Response state to verify that another Data packet is coming from its other neighbor. Otherwise, a timeout occurs and the node goes either to its deferred contention period or to an Idle state. Once the Adjusted Response state finishes, ACK for the corresponding Data packets are sent through each activated channel, and the receiver waits until the time defined in its BTS finishes to set up its channels as available and start BTS in zero.

8) Backoff state: In the case of collisions, nodes execute a Binary Exponential Backoff (BEB) retransmission algorithm for collision recovery that is a modification to the used in [3]. The sender goes to the Backoff state for certain amount of time, and the transmitter starts again a Channel Assignments period for a new contention time. Once the Backoff finishes, if the number of retransmissions have reached the maximum allowed number the packet is dropped and the node goes to the Idle state. A node receiving an RTS while in the Backoff period defers its remaining time until the communication process finishes.

9) Adjusted Response state: Nodes in this state wait for a certain period of time called ARS to see if another packet is coming or not. Since a node can receive packets from its neighbors through different channels, these packets do not necessarily arrive at the same. In this case a ARS time is included. The goal of this state is to avoid that this ARS also occurs when a node responds to its neighbors by using an Adjusted Response period. With this state, a node knows if there was a request or not from both channels and later responds simultaneously through both channels. This ARS time is defined as a constant value in the 2MAC protocol, and included as a portion of the timeout (the timeout will be the ARS plus the cost of a control packet round trip time).

10) Blocked to Send state: A node is in this state when BTS are received from its both channels or just from the activated channel. It uses the biggest of the BTS times to block the channel for that period of time. Once the time finishes, the BTS values are reset and the node goes to a Channel Assignments state if there are packets to transmit, otherwise it goes to the Idle state.

V. SW-MER: A STOP AND WAIT AND SLIDING WINDOW-BASED ARQ PROTOCOL WITH EXPONENTIAL RETRANSMISSIONS

A. General description

The SW-MER protocol is a combination of stop and wait and sliding window protocols. The transmitter sends a group of M packets (window size of the transmission) and then waits for the (group) acknowledgment. As in the regular stop and wait protocol, the sender is not allowed to send more packets until the acknowledgment is received. In order to implement the SW-MER protocol, each packet contains additional information, such as a consecutive number representing its position in the window, a number that tells how many packets are missing to finish the reception of the current window, and the number of times the packet is being repeated in that window.

At the other end, the receiver verifies that the incoming packets are error-free and in sequence. If so, it sends them to the upper layer and acknowledges them all in one ACK packet to the sender. This ACK packet contains a vector V of size M in which every position in the vector reflects the state of every packet received, as follows:

$$V[i] = \begin{cases} 1 & \textit{if packet i arrived with errors} \\ & \textit{or did not arrive} \\ 0 & \textit{if packet i arrived without errors} \end{cases} \quad (2)$$

Upon receiving the ACK with vector $V[i]$, the sender retransmits all packets i that were not received correctly. As in any ARQ protocol, the SW-MER sender also waits for the corresponding ACK packet before a retransmission timer expires. If a timeout occurs, the sender retransmits the same M packets that were sent in the last window; otherwise, only those packets with $V[i] = 1$ are retransmitted, which are sent in the initial positions of the following window. If there is enough space in the next window after including the retransmitted packets, the sender fills the rest of the window with new packets. In the case of retransmissions of already retransmitted packets, the sender repeats those packets in an exponential manner, i.e., packet i will be retransmitted 2^n times in the following window until a maximum of M times, where n is the number of windows where packet i has been received with errors. Figures 7 and 8 graphically show how the window mechanism of the SW-MER protocol works in the case of a sender that wants to send 8 packets using a window of size $M = 6$.

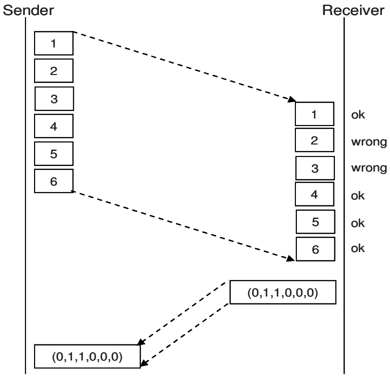


Fig. 7. First transmission, M=6 and packets 2 and 3 arrive with errors.

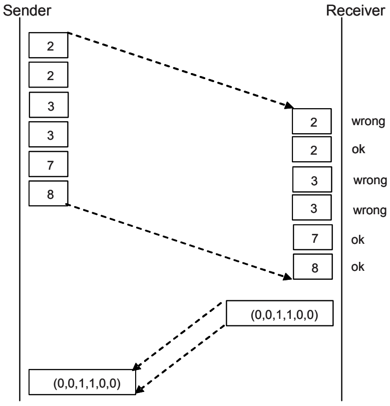


Fig. 8. Second transmission, M=6 and packet 3 arrive with errors again.

After the first transmission, the receiver sends an ACK packet with vector V saying that packets 2 and 3 arrived with errors (see Figure 7). Upon the reception of the ACK packet, the sender transmits packets 2 and 3 two times each, and fills the rest of the window with new packets. During this second transmission, packet 3 arrives with errors again (none of the two copies arrived without errors). The receiver sends the appropriate ACK packet with the new vector V (see Figure 8). Then, the sender retransmits packet 3 2^2 times in the next window. The process continues until the packet is received correctly or the maximum number of retransmissions is reached, which is set in our protocol to the window size $M = 6$. With this procedure, more copies of the same packet are transmitted in case of repeated errors with the idea of increasing the reliability of the protocol.

An important aspect of the protocol is the size of the buffer at the receiver. If the buffer is not dimensioned appropriately, packets might be dropped because of lack of space. In order to guarantee that packets are sent to the upper layer in order, packets that arrive correctly are stored in memory and kept there until any of the missing packets are retransmitted and received correctly. Given the size of the window, M, the size of the buffer BS needed to guarantee the operation of the protocol is given by Equation 3 as:

$$BS = M \times (\lg_2 M - 1) + 2 \tag{3}$$

VI. CHANNEL ERROR MODELS

In order to evaluate the performance of the data link control protocols under consideration, two channel error models were utilized. The first model corresponds to the simple Bernoulli model in which the Packet Error Probability (PER) can be easily calculated as:

$$PER = 1 - (1 - BER)^N \tag{4}$$

where N is the number of bits in the packet and BER is the Bit Error Rate.

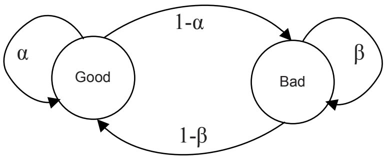


Fig. 9. Two-state Markov model representation.

The second model was developed by modeling the errors of an underwater channel using a similar approach as the ones described in [14] and [15], where a two-state Markov chain is used to model the errors in wireless communications channels. The first step in the model generation is the characterization of the channel. We did not perform any real measurements; instead, we utilized the channel characterization results presented in [16], where real measurements were taken using acoustic communications in shallow water (depth 15-20m) at a range of 50m.

From [16], we used the channel impulses (the amplitudes and the values) to recreate the channel behavior, including the effect of mobility considering the doppler effect while moving at 1 m/sec, a depth of 4m, and a distance of 50m. Using Matlab simulations, we generated a trace of 10 million impulses using Orthogonal Frequency-Division Multiplexing (OFDM) and Binary Phase Shift Keying (BPSK) modulation schemes. The generated impulse results, +1, -1 and 0 were taken as a representation of bit with errors (+1 and -1 values) and bits without errors (0 value). Then, with this trace, we obtained the transition probability matrix A and the error probability matrix B of the two-state Markov chain shown in Figure 9.

Given initial values of A and B, a Hidden Markov Model (HMM) approach was taken to generate the channel error model [17]. In this process the well-known Baum-Welch algorithm [18] was used to find the unknown parameters of the HMM. The initial values of the A and B matrices (A_0 and B_0) as well as the final values (steady state matrices) found by the model are as follows:

$$A_0 = \begin{bmatrix} 0.98 & 0.02 \\ 0.05 & 0.95 \end{bmatrix} A_{ss} = \begin{bmatrix} 0.8116 & 0.1884 \\ 0.0095 & 0.9905 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0.9 & 0.9 \\ 0.1 & 0.1 \end{bmatrix} B_{ss} = \begin{bmatrix} 0.9909 & 0.68 \\ 0.0091 & 0.32 \end{bmatrix}$$

VII. PERFORMANCE EVALUATION

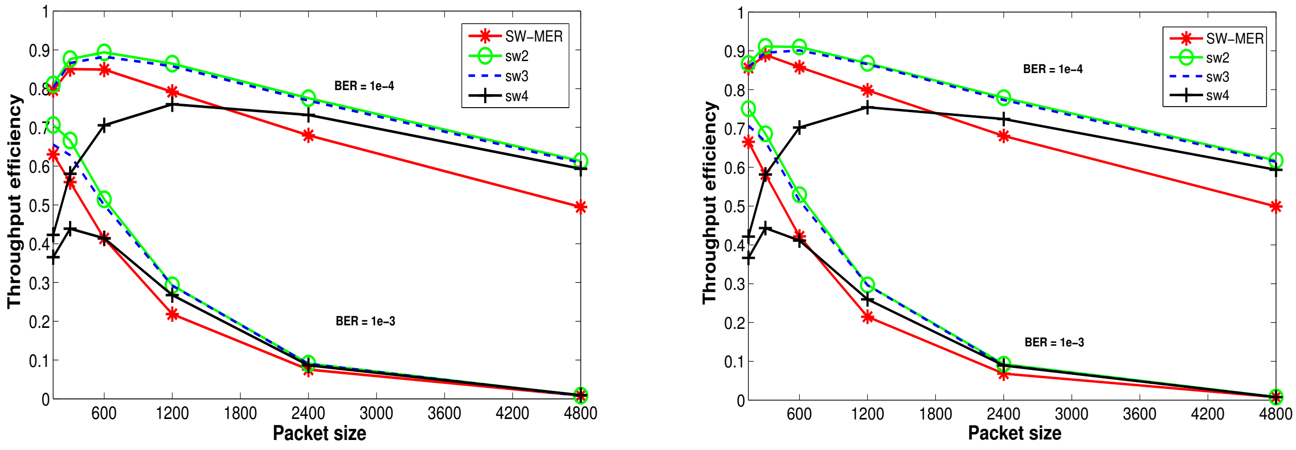
The performance evaluation is presented in three sets of results. The first and second sets include the evaluation of the logical link control protocols and the MAC protocols alone. The third set presents the evaluation of the combined SW-MER and 2MAC protocols together. These results are presented next. In all experiments the acoustic channel speed of the links was set to 2400 bps and the propagation speed utilized was 1500 m/s.

A. *Logical link layer protocols*

In this section we compare the performance of the proposed SW-MER protocol with that of the Stop and Wait protocols described in [10], [12] and [13] (shown in the plots as $sw2$, $sw3$, and $sw4$, respectively). The evaluation was performed using the two error models described in the last section, and the performance metrics utilized were throughput efficiency and packet delivery rate. This last metric indicates the level of reliability provided by each protocol.

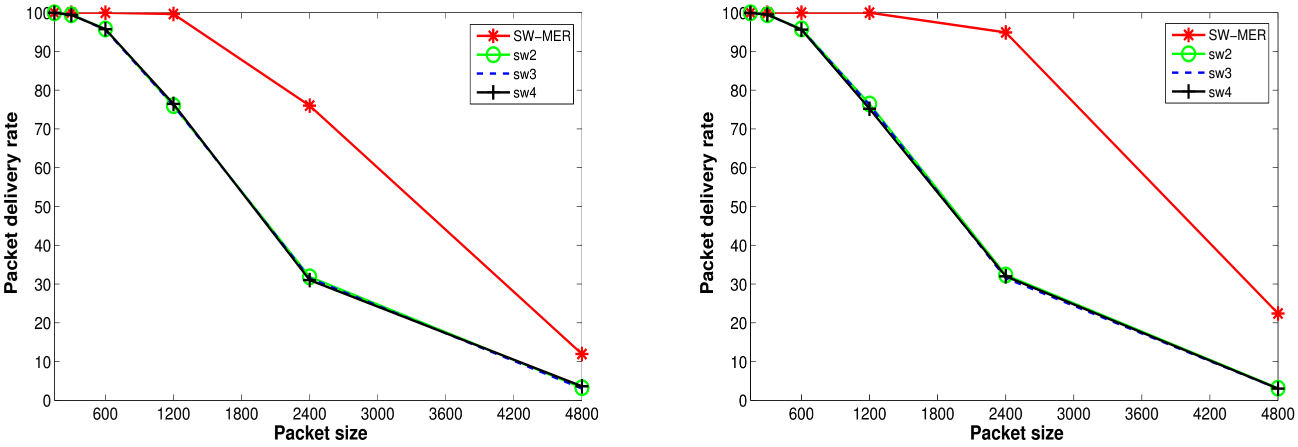
Figures 10(a) and 10(b) show the throughput efficiency of the logical link control protocols using a window size of 8 and 16 packets. As it can be seen, regardless of the BER, all protocols present similar performance. The benefit of the SW-MER protocol is in the packet delivery ratio as shown in Figures 10(c) and 10(d) based on a channel with a BER of 10^{-3}. As expected, the performance of the protocols decreases with the packet size. However, the superiority in reliability of the SW-MER protocol is demonstrated. This is due to the exponential increase packet retransmission strategy utilized by SW-MER. In the case of a BER of 10^{-4} (results not shown here), all protocols experienced similar performance, indicating that SW-MER is better only in those extreme scenarios with very poor quality channels.

In the case of the underwater channel error model, we performed similar experiments to compare the results with the Bernoulli error model, using the trace, Markov model, and parameters described in Section VI. As it can be seen from Figure 11(a), the throughput efficiency is very low compared with the Bernoulli model, indicating that the underwater error model introduces a larger amount of errors. This is magnified considering that the experiments utilized smaller packet



(a) Throughput efficiency, window size = 8.

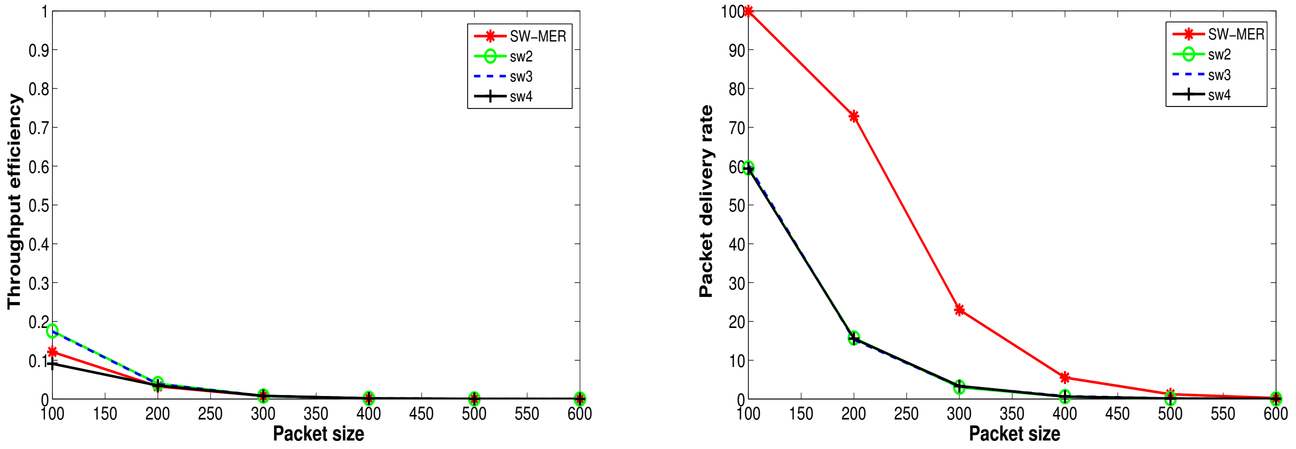
(b) Throughput efficiency, window size = 16.



(c) Packet delivery rate, window size = 8, $BER = 1\text{x}10^{-3}$.

(d) Packet delivery rate, window size = 16, $BER = 1\text{x}10^{-3}$.

Fig. 10. Throughput efficiency and packet delivery rate using the simple Bernoulli error model.



(a) Throughput efficiency, window size = 16.

(b) Packet delivery rate, window size = 16.

Fig. 11. Throughput efficiency and packet delivery rate using the shallow water error model.

sizes. The packet delivery ratio of the protocols is shown in Figure 11(b). From these last plots, it can be concluded that in order to have an acceptable throughput and packet delivery ratio in an underwater channel, packets cannot be longer than 250 bytes. Similar results were found using smaller window sizes.

B. MAC layer protocols

The experiments at the MAC layer compare the IEEE 802.11 protocol with the proposed 2MAC protocol both using a channel speed of 2400 bps and packet sizes of 1200 and 2400 bits. The protocols were evaluated considering a linear network of four nodes 50m apart simulating a swarm of UAVs mapping the ocean floor without errors and without the logical link control sublayer.

The traffic in the network was generated as follows. Each node generated flows to all other nodes sending packets according to a Poisson process. The rate of the flows was set so that the load in the network was set to the specific desired level. In addition, a tagged flow sending packets from node one (left extreme of the network) to node four (right most node) was established and monitored. The performance results shown in all graphs are related to the performance of this tagged flow, which, as a reference, is indicated in the plot by a dotted line with a maximum possible throughput of 60 bps or 30 bps in the case of packet sizes of 2400 and 1200 bits, respectively. A simple network layer was included on top of the MAC layer to route incoming packets to the following adjacent node.

Figure 12 shows the throughput results of the MAC protocols. As it can be seen from the figure, the new MAC protocol provides considerably better performance than the current CSMA/CA protocol, which can be easily explained by the use of multiple channels and the reduction of collisions.

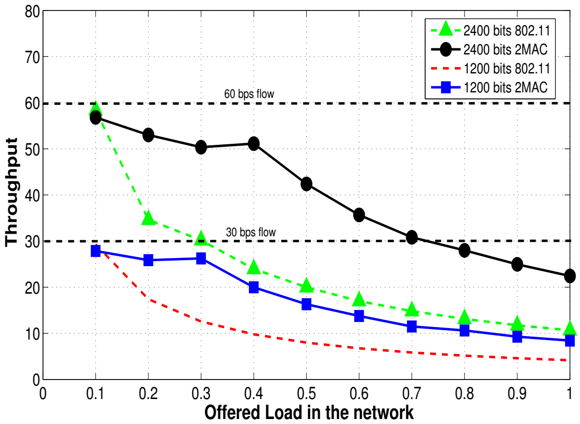


Fig. 12. Throughput of the MAC protocols.

C. The combined SW-MER and 2MAC data link layer

The performance of the tagged flow was also assessed in the same linear network using the same parameters in the last subsection but now using the proposed SW-MER and 2MAC protocols together and including the Bernoulli and the underwater error models (shown in the plot as MM - Markovian Model). Figure 13 shows the performance of the tagged flow when sending packets of 1200 bits. From the figure, two main conclusions can be drawn. First, the proposed MAC protocol, as expected, improves the performance over the CSMA/CA protocol. Second, it is clear that the performance in all cases is fairly poor but even worse using the underwater channel error model, indicating that more research is needed at all these layers to improve the performance of acoustic-based underwater communication systems.

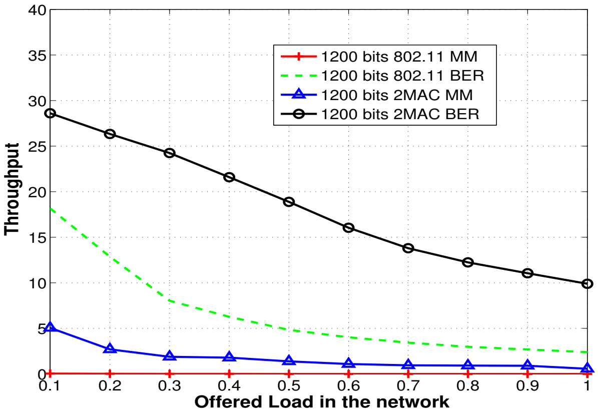


Fig. 13. Throughput of the combined SW-MER and 2MAC protocols.

D. 2MAC's Backoff algorithm

The well-known Binary Exponential Backoff (BEB) algorithm was designed to deal with congestion in 802.11-like networks. During congestion epochs, nodes are forced to wait longer and longer after successive collisions by doubling the size of the contention window each time. Although this has been shown to be a good mechanism in wireless local area networks, its direct application in long propagation delay underwater channels might not be a good idea. In other words, doubling the contention window at every collision opportunity may be too much of an increase since a large contention window will make the node to wait for a very long time.

In this section we introduce a very simple modification of the BEB algorithm and compare it with the one presented in [2], which increments the contention window following a Multiplicative Increase Linearly Decrease (MILD) strategy. Our algorithm, which is depicted in Equation 5, increments the contention window by a factor of 1.25 and limits the number of increments to 3; therefore the CW_{Max} is equal to 9 given that the CW_{Min} is equal to 4. Once the retransmitted packet goes through, the contention window is not reset to CW_{Min}, as in 802.11, but it is reduced by that value for every successful packet. Given the numbers for CW_{Min} and CW_{Max} utilized, the contention window goes to CW_{Min} in just two steps.

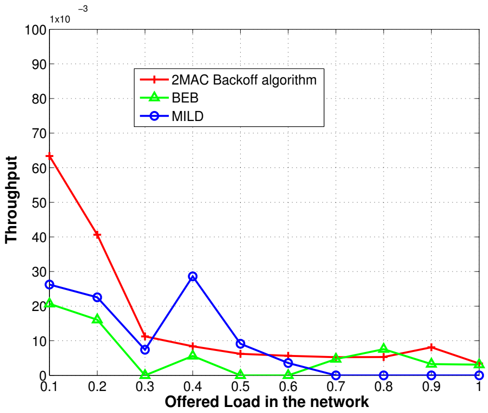


Fig. 14. Throughput with different backoff algorithms.

$$\begin{cases} CW = min(1.25CW, CW_{Max}) & \textit{when collisions occur} \\ CW = max(CW - CW_{Min}, CW_{Min}) & \textit{when successfully packets} \end{cases} \qquad (5)$$

Figure 5 shows the throughput of the three schemes as a function of the offered load using packets of 300 bytes, the underwater channel error model, and the combined SW-MER/2MAC data link layer. As it can be seen, the proposed algorithm improves the throughput of the tagged flow, especially at low network loads. Looking at the figure, since the performance of the three algorithms is very similar at higher loads, one may think that the protocol is not working well when it is supposed to provide more benefits; however, this is not true. What happens is that at high loads the performance of the tagged flow is dominated by the bad underwater channel, mostly by retransmissions of packets in error, not collisions.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper a new data link layer designed for acoustic-based underwater communication systems using OFDMA technology at the physical layer is proposed to support applications of swarms of autonomous underwater vehicles. A logical link control protocol, called SW-MER, is included. SW-MER combines stop and wait and sliding window mechanisms to increase the channel utilization and implements an exponential retransmission strategy to increase the reliability of the protocol under poor channel conditions, like the ones commonly found underwater. SW-MER is compared with other traditional ARQ protocols in terms of throughput efficiency and percentage of accepted packets. The results of the simulations show that SW-MER provides similar throughput efficiency but better reliability compared with the existing protocols, especially under poor channel conditions.

A new MAC layer protocol, called 2MAC, designed to coordinate and take advantage of the multiple sub-channels made available by the use of OFDMA is also included. The 2MAC protocol shows its superiority over the well-known 802.11 protocol. 2MAC uses three channels to transmit or receive data simultaneously from different neighbors without collisions and using only one transceiver. The combined SW-MER and 2MAC protocols are also evaluated using a linear network of four nodes using a Bernoulli based error model and a more realistic underwater channel error model. Although the superiority of the proposed protocols is demonstrated, the performance if fairly poor especially when using the underwater channel error model, indicating that more research is needed in the area.

ACKNOWLEDGMENT

This work has been partially supported by the National Science Foundation's Research Experiences for Undergraduates program under grant No. 0754537. The authors would like to thank Mr. Javier Espinosa from University of Puerto Rico for his help in the simulation work.

REFERENCES

[1] H. Yin and S. Alamouti, "OFDMA: A Broadband Wireless Access Technology," in *Sarnoff Symposium, 2006 IEEE*, March 2006, pp. 1–4.

[2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 212–225.

[3] P. Karn, "MACA - a new channel access method for packet radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, September 1990, pp. 134–140.

[4] C. Fullmer and J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet Radio Networks," *Computer Communication Review*, vol. 25, No. 4, 1995.

[5] M. Molins and M. Stojanovic, "Slotted FAMA: A MAC Protocol for Underwater Acoustic Networks," in *Proceedings of IEEE Oceans*, 2006.

[6] B. Peleato and M. Stojanovic, "A MAC protocol for ad-hoc underwater acoustic sensor networks," in *Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 113–115.

[7] M. Tao, S. Haoshan, and W. Yu, "A MAC Protocol for Underwater Sensor Networks," pp. 144–148, 2007.

[8] R.K.Creber, J.ARice, P.A.Baxley, and C.Fletcher, "Performance of undersea acoustic networking using RTS/CTS handshaking and ARQ retransmission," in *Proceedings of EEE Oceans 2001*, 2001, pp. 2083–2086.

[9] J.-K. Yeo, Y.-K. Lim, and H.-H. Lee, "Modified mac(media access control) protocol design for the acoustic-based underwater digital data communication," *IEEE*, p. 364368, 2001.

[10] M. Schwartz, *Telecommunication Networks*. Adison Wesley, 1998.

[11] A. R. K. Sastry, "Improving Automatic Repeat-Request (ARQ) performance on Satellite Channels Under High Error Rate Conditions," *IEEE Transactions on Communications*, vol. 23, No. 4, pp. 436 – 439, 1975.

[12] J. M. Morris, "Optimal Blocklengths for ARQ Error Control Schemes," *IEEE Transactions on Communications*, vol. 27, No. 2, pp. 488–493, 1979.

[13] P. F. Turney, "An improved Stop-and-Wait ARQ Logic for Data Transmission in Mobile Radio Systems," *IEEE Transactions on Communications*, vol. 29, No. 1, pp. 68–71, 1981.

[14] H. S. Wang and N. Moayeri, "Finite-State Markov Channels-A Useful Model for Radio Communications Channels," *IEEE Transactions on Communications*, vol. 43, pp. 163–171, February 1995.

[15] W. Turin and R. V. Nobelen, "Hidden Markov Modeling of Flat Fading Channels," *IEEE Journal on Selected Areas in Communications*, vol. 16, No. 9, pp. 1809–1817, December 1998.

[16] M. Chitre, J. Potter, and O. S. Heng, "Underwater acoustic channel characterization for medium-range shallow water communications," *OCEANS'04 MTTS/IEEE Oceans, 2004*, pp. 40–45, November 2004.

[17] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, pp. 4–16, 1986.

[18] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *The Annals of Mathematical Statistics*, vol. 41, pp. 164–171, February 1970.