I. Simulator

A. Introduction

We have designed and implemented a framework to adapt the popular NS2 simulator to the underwater networking environment. In this effort we have implemented new channel, propagation, phy, and MAC layers. Complete documentation of the code and its use is available at [**??**]. Our focus has been on accurately representing the underwater channel. In this paper, we only describe the functional components necessary for understanding the results given below.

The channel and phy layers are nearly identical to their terrestrial wireless counterparts. The main difference in the channel layer is the use of the propagation speed of sound in water. Additionally, the propagation layer is now attached to the channel layer, and the channel is reponsible for updating the signal strength at each of the receiving nodes.

We have developed a propagation layer based on the Bellhop Gaussian Ray tracing simulator. Bellhop is available at [**??**]. Bellhop calculates the channel impulse response given a set of environmental parameters. All of the UAN propagation models developed so far calculate channel noise using [**??**].

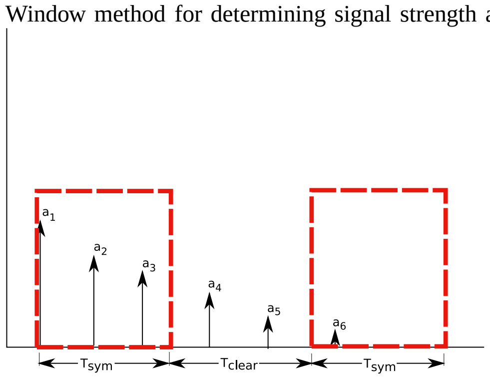
A detailed description of how Bellhop works is far beyond the scope of this document. We provide only a quick description of how Bellhop is used to get a channel impulse response, and then how we use that impulse response to calculate the pathloss. Bellhop takes, as an input, a set of envrionmental paramaters. These paramaters include the Sound speed profile, propagation frequency, and surface and bottom characteristics. The environment file also includes the transmitter depth, a set of receiver depths and a set of receiver ranges. Bellhop then creates an arrival file that includes ray arrivals, amplitudes, phase shifts, and delays to all of the receiver range and depth pairs. The arrivals for a receiver depth/range pair can be summed to find the pathloss from the transmitter to the receiver.

Two different propagation layers have been developed. One writes the necessary environment file and runs the Bellhop process. The other loads cached information from a set of data files. Because the Bellhop process is very time consuming (often more than 30 seconds on a 1.8 GHz Intel Core 2 Duo for a single run), significant performance improvements are seen by using pre calculated data. Sets of data files can be created using scripts found at **??**.

The transmission loss, TL, is calculated

$$TL = \sum_{\substack{i \\ \tau_{di} \in [\tau_{prop}, \tau_{prop}+t_{sym}]}} |a_i|$$

Fig. 1. Window method for determining signal strength and ISI



For τ_{prop}, the propagation delay from the source to the receiver, a_i, the amplitude of the ith multipath arrival as found by Bellhop and τ_{di}, the delay of the ith arrival. The received power is then $P_r = P_{tx} - PL$ where P_{tx} is the transmit power in dB.

Collisions and packet reception are handled in the Phy layer. When an incoming packet is handed to the Phy layer. The Phy layer keeps a record of all packets which may interfere with the current reception. The SINR is calculated as

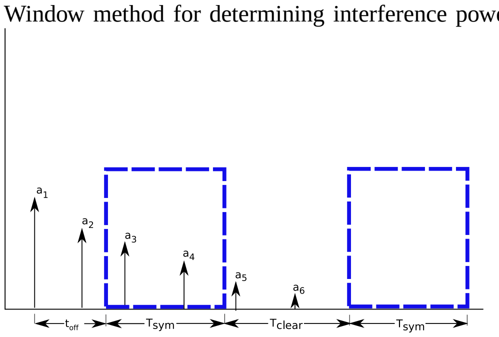
$$SINR = 20log(\frac{P_r}{(\sum_i P_i) + N})$$

Where P_r is the received power, P_i is the power of an interfering packet at the receiver and N is the noise power.

Two Phy layers have been developed, a generic PHY and a PHY which attempts to mimmick the implementation of slow FH-FSK found on the WHOI Micromodem. In the generic PHY layer, collision decisions are a function of SINR and overlap time. The maximum acceptable overlap between packets and the minimum acceptable SINR are user adjustable parameters. If a collision does occur, the PHY layer will still attempt to receive the entire packet. If a packet arrives before the PHY layer finishes attempting to receive, the arriving packet will be missed. This behavior is consistent with the behoavior of the WHOI Micromodem.

The FH-FSK PHY layer assumes that all nodes in the network are using FH-FSK and operating on the same hopping pattern. In this case, two packets may overlap in time yet never overlap in frequency. The FH-FSK modulation scheme calculates the interferring signal power which overlaps in frequency as well as time. Fig. 2 shows how the interfering signal power is calculated. t_{off} will be the difference in arrvial times between the packet being decoded and a colliding packet mod the symbol time plus the clear time. We make the assumption that the interference arrival spread does not significantly

Fig. 2. Window method for determining interference power.



overlap with two receiving windows. In all simulations we have done thus far, this has been a valid assumption.

II. Micromodem MAC Layers

We have explored the possibility of using the WHOI Micromodem in an underwater networking environment. Many proposed MAC protocols require time synchronization, position information, or rely on features that are not available in the Micromodem. We initially explore options that are easily implemented on the WHOI Micromodem without requiring changes to the modem's hardware or firmware. Later we identify key modem improvements that will allow for substantial increases in network performance.

We begin by analyzing the performance of pure aloha in a Micromodem network which uses FH-FSK as its protocol. We expect a higher throughput than the theoretical maximum, as the implementation of FH-FSK on the micromodem is resitant to interference. Further on we propose a simple backoff scheme similar to the one found in 802.11 and look at its performance in simulation. We then look at two simple improvements, the addition of carrier sensing and the addition of a capture threshold.

A. Pure Aloha

Following the standard Aloha analysis, consider a completely connected network of infinite users where aggregate traffic follows a Poisson distribution with mean offered load $G = g/p_l$. where g is the packets transmitted per second and p_l is the transmit time of each packet in seconds.

Nodes use a pure aloha access scheme where any node with a packet to send will immediately transmit the packet. For the case where any overlap of packets is considered a collision, the event of a successful transmission becomes the event that only one packet is delivered over a time period of $2p_l$, so the probability of a successful transmission is then

$$P\{Success\} = e^{2G}$$

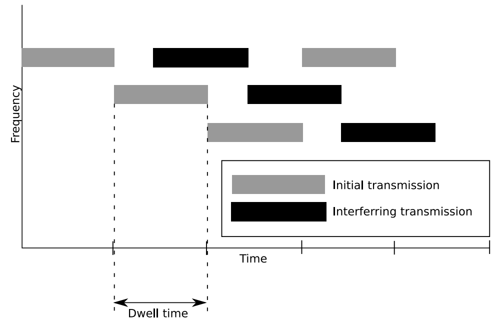


Fig. 3. Frequency Vs. Time

from the definition of a poisson distribution where the number of events per 'unit time' is 1. The probability of transmitting a packet is G, so the average throughput, S, is given by

$$S = Ge^{-2G}$$

Which yields the standard result of maximum throughput of 0.184 at G=0.5.

If nodes are using frequency hopping, all nodes are using the same hopping pattern, and nodes are not capable of receiving multiple simultaneous packets, the assumption of a collision occuring from any overlap is invalid. Two packets may overlap for a significant period of time without ever overlapping in frequency. The contention times for a transmitted packet in this case will still be stretched over a time period of $2p_l$, but now in spread out intervals as illustrated in Figure 3. Looking at the packet transmission over a single frequency, it is apparent that the contention window is now several intervals. If a potentially interferring transmission occurs such that it begins its hopping pattern while the initial transmission is transmitting on a different frequency, the two packets will not interfere with each other.

The length of time spent transmtting on each frequency before hopping is denoted t_{dwell}. The number of times cycled through each frequency per packet is n_{cyc}, and the average delay spread, of the arriving signal at a receiver is t_{del}. The contention time for each signal, t_{scon}, is give by

$$t_{scon} = 2t_{dwell} + t_{del}$$

All of the contention intervals summed up, t_{con} is

$$\begin{aligned} t_{con} &= 2n_{cyc}(t_{scon}) \\ &= 2n_{cyc}(2t_{dwell} + t_{del}) \end{aligned}$$

Because of the memoryless property of the Poisson distribution, we can assume the probability of transmitting in any one of the intervals is the same as transmitting in one long interval with length equivalent to

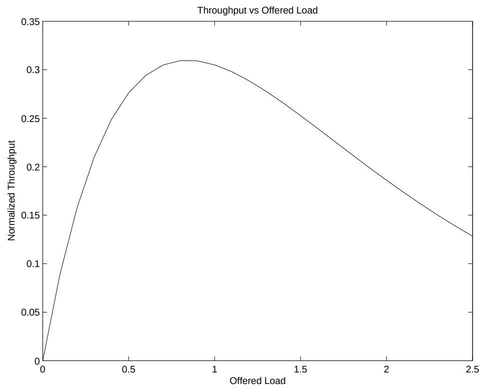


Fig. 4. Throughput vs Offered Load for Micromodem Aloha

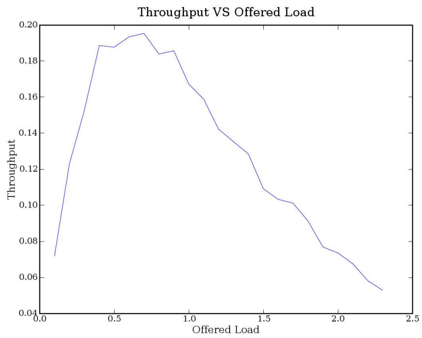


Fig. 5. Baseline Aloha. Bins=1, t_{sym}=0

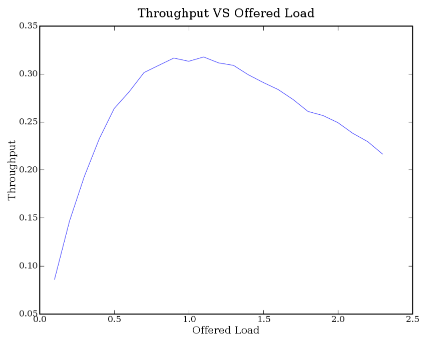


Fig. 6. NS UAN FH-FSK Simulation output

the sum of the lengths of all of the shorter intervals. The probability of a transmission being received successfully then becomes

$$P\{Success\} = e^{-2Gn_{cyc}(2t_{dwell}+t_{del})/p_l}$$

and the throughput

$$S = Ge^{-2Gn_{cyc}(2t_{dwell}+t_{del})/p_l}$$

The theoretical throughput using the parameters given in Table I are shown in Figure 4. Simulation output using the NS2 UAN module with the same parameters are shown in Fig. 6. The throughput results are slightly higher in simulation due to laxed collision requirements, i.e. It is possible for two symbol intervals to overlap in time and frequency but have SINR such that the originially acquired transmission can still be received. For contrast, the baseline throughput seen with a single center frequency (a simple FSK) network is shown in Fig. 5. Again in the non frequency hopping case, a slight improvement in throughput is seen over the previously mentioned theoretical maximum throughput due to the assumption that a packet is receivable until it falls below a predefined SINR threshold.

TABLE I
PARAMETERS USED IN WHOI MICROMODEM

Paramater	Value
# Bins	13
Dwell time (ms)	12.5
Delay spread (ms)	70

B. Random Backoff MAC

To increase the probability of successful packet reception a simple slotted random backoff time is proposed. The timer will increase the robustness of the network by reducing the probability of collision and does not increase overhead by requiring synchronization or network configuration information. This proposal is completely compatible with the current features included in the WHOI Micromodem.

The state transition diagram for the proposed MAC protocol is shown in Fig. 7. When a packet is received for transmition, a slot is chosen at random between 0 and CW-1. CW is called the contention window and is a network specific parameter that should be chosen based on the average number of contending nodes. The length of each slot is also a network specific parameter, which should be based on the maximum distance between nodes. When the packet for transmission is received at the MAC layer and a slot is chosen, a timer is set and begins counting down. Whenever the MAC enters the receive state, the timer is paused until the MAC transitions to another state. When the timer reaches zero, the MAC transitions to the TX state and the waiting packet is transmitted. When the transmission is finished, the MAC returns to the IDLE state.

The random backoff employed in this protocol resembles a simplified version of the 802.11 DCF. The main difference is in the lack of carrier sensins, which is a suggested improvement in the following section. Using the results from the analysis in [**??**], the expected saturation throughput in terms of contention window is

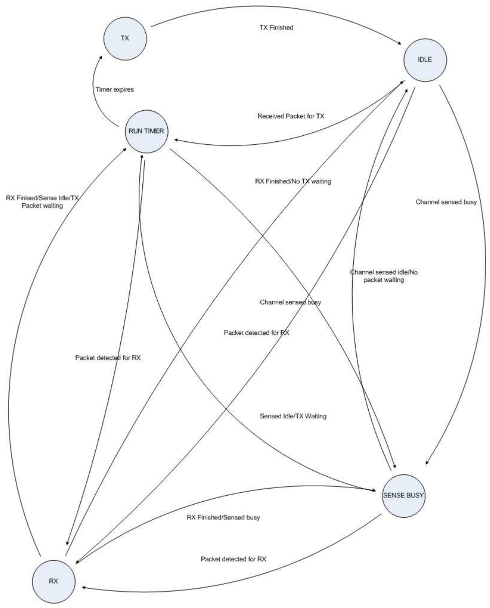


Fig. 7. State transition graph for Aloha with Random Backoff

given by

$$S = \frac{T_p}{T_s - T_c + \frac{T_{slot}(1-P_{tr})/P_{tr}+T_c}{P_s}}$$

T_p is the average length of a packet. T_s is the time the channel is busy for a successful transmission. T_c is the time the channel is busy for a collision. P_{tr} is the probability of at least one transmission in a slot. P_s is the probability of a successful transmission. Let $\tau = 2/(1 + CW)$ and this results in

$$S = \frac{T_p}{T_s - T_c + \frac{T_c-(1-\tau)^n(T_c-1)}{n\tau(1-\tau)^{n-1}}}$$

With n the number of contending nodes. For our case, we let T_s=T_c=$T_p + \delta$ for δ, the max propagation delay. We also set T_{slot} to the max propagation delay.

We ran simulations with 9 saturated nodes in a star topology placed 200 m from a central receiver and varied the contention window size from 5 to 99. $T_p = 3.2s$ and $T_{slot} = 0.5s$. Fig. 8 shows the simulation results overlayed the predicted results. For high values of CW the theory matches the results exactly. Using the same simulation scenario, but using traffic generated from a poisson distribution, the throughput vs offered load is shown in **??**.

III. Improved MAC

The above results represent the limit of a network using a fixed hardware platform like the Micromodem. In this section we explore the effects of two significant improvements on the random backoff protocol shown

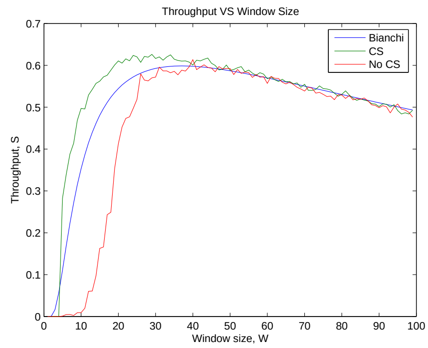


Fig. 8. Throughput vs CW size

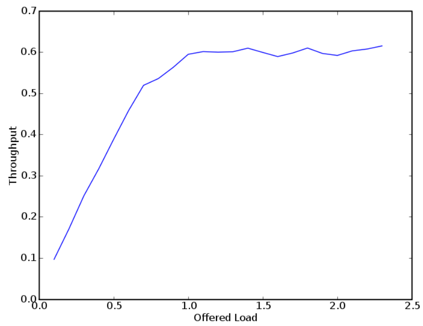


Fig. 9. Simulation results for random backoff MAC with no CS

above. The first improvement is the utilization of carrier sense. Secondly we investigate the effect of implementing a capture threshold on the protocol.

A. Carrier Sense

Here I was planning on implementing a threshold based carrier sense mechanism. I haven't come up with a good way of modeling the preamble carrier sense yet, but I've been thinking about it.

The random backoff MAC is tailored for carrier sensing by adding a "sensed busy state". If the channel is sensed to be busy, the MAC transitions to the "Sensed Busy" state and stays there until the channel is again sensed to be idle. The countdown timer will be stopped now whenever the MAC is in either the RX or the Sensed Busy states. The same simulation scenario as done in section **??** is performed with the described additions. The results are shown in **??**.

B. Capture

The above MAC protocols make no assumptions about a single or multi hop environment. Implementing a mechanism for capturing a strong signal in the presence of a weak increases the robustness and can drastically increase the theoretical throughput. To illustrate this point we consider a network topology that has two gateway nodes and that there are clusters of nodes centered around each of the two gateways. Nodes are placed such that some subset of nodes can reach either gateway, however, the signal at the furthest gateway will be severely attenuated. If the gateway were to attempt to decode the incoming packet in its entirety, even after a stronger signal arrives from a closer node, both packets are likely to be lost. This effectively reduces the capacity of the network by half.

The capture is implemented such that if the ratio between the incoming signal powers is greater than a predefined thereshold, λ, i.e. capture an arriving packet with receiver power, P_{R2} if

$$\lambda > \frac{P_{R1}}{P_{R2}}$$

The effect of implementing capture in the Pure Aloha MAC with random backoff is shown in Fig. **??**

IV. Conclusion