Modeling the Underwater Acoustic Channel in ns2

Albert F. Harris III
Department of Computer Science
University of Illinois at Urbana–Champaign
aharris@cs.uiuc.edu

Michele Zorzi
Department of Information Engineering
University of Padova, Italy
zorzi@dei.unipd.it

ABSTRACT

Underwater acoustic networks have the potential to support a large variety of applications, such as mining equipment and environmental monitoring. Although underwater acoustics has been studied for decades, underwater networking and protocol design is just beginning as a research field. One critical tool used for the design and testing of new protocols is a network simulator. For simulators to be useful tools, accurate models of both the channel and the modem need to be implemented. In this paper we present the design and implementation of our interface and channel model for underwater acoustic networks in the ns2 network simulator. We show that the models accurately predict the channel conditions and interface costs by comparing them to previously published numerical predictions of channel state. Finally, we present a case study of a protocol designed and simulated using our model. Our simulation code is open source and available for general use.

1. INTRODUCTION

Underwater acoustic networks have the potential to support a wide variety of applications from facilitating communication between autonomous underwater vehicles to supporting the remote monitoring of underwater mining equipment or environmental conditions [1]. Even though underwater acoustics has been studied for many years, and communications technologies exist for underwater scenarios, interest on networking and protocol design in this environment is just beginning. As such, there is currently no standard underwater simulation module for any of the major network simulators. To accurately model underwater communication, channel, propagation, and interface models are required.

Since the underwater environment is so different from its terrestrial counterpart [1, 19], it is likely that existing wireless modules cannot be easily reused, and specific underwater extensions will be needed. The underwater environment differs from the terrestrial radio environment both in terms of its modem energy costs, and in terms of the channel propagation phenomena. The underwater channel is characterized by long propagation times and frequency-dependent attenuation that is highly affected by the distance between nodes [19] as well as by the link orientation.

The main contribution of this work is an underwater propagation, channel, and interface model for the widely-used ns2 network simulator [13]. A challenge in building such a model lies in the fact that many of the features of the channel, such as bandwidth, depend on the distance between the two nodes and the orientation of the link. However, in ns2, the bandwidth is usually assumed to be fixed and used by layers higher than the propagation and channel model. Therefore, in underwater scenarios, bandwidth information has to be calculated and returned to the upper layers of the simulator, which is not the case for other radio models. The ns2 simulator divides the layers below the MAC layer into four components: Propagation, Channel, Physical, and Modulation. We provide modules for each layer, allowing protocol developers to concentrate efforts on the higher layers of the network protocol stack.

In order to validate our model, we run a number of simulations and compare the results to those obtained from analytic and numerical methods of describing the underwater environment that are known to be accurate, ensuring that no implementation errors have been introduced. This validation additionally shows that the ns2 modules accurately capture the fundamental features of the underwater channel and can serve as a base to continue protocol research and development. Additionally, we present a case study of the ns2 underwater model being used to design and develop an energy-efficient routing protocol. This case study highlights the need for a complete model in ns2 to support underwater networking research in realistic scenarios.

The rest of this paper is as follows. Section 2 presents related work in the areas of simulation design and underwater acoustic networking. Section 3 motivates the need for a comprehensive underwater communication model for a standard simulator tool. Section 4 presents the details of our ns2 model and is broken up into four subsections. Section 4.1 presents the propagation model for the underwater channels. Section 4.2 presents the details of the channel model. Section 4.3 presents the physical layer model and Section 4.4 presents the modulation and error model. Section 5 compares simulation results with analytic results to verify the simulation model. Section 6 presents a case study, using the model to develop and simulate a routing protocol. Finally, Section 7 presents some conclusions and future directions.

NSTools '07, October 22, 2007, Nantes, France
Copyright 2007 ICST 978-963-9799-00-4.

2. RELATED WORK

The ns2 network simulator [13] is one of the most widely-used tools in network protocol research and development. It contains implementations of virtually all major networking protocols, allowing researchers to easily test the performance of new protocols against a large variety of other algorithms. Channel models that are currently implemented in ns2 include free space path loss, two ray ground, and a shadowing model. These are the most widely used. Additional channel models have been developed (*e.g.*, WiMAX [4]), but no model of the underwater acoustic channel is in place.

Previous work in simulator analysis [2] has demonstrated the importance of accurate channel and physical layer models to obtaining accurate simulation results. This strongly motivates the need for a dedicated underwater acoustic model for use in network simulators.

Analytic models of the underwater environment have recently been developed [11, 19, 22] and used in the design of underwater protocols [11, 5, 7, 12, 16, 17, 18]. However, because the models have been implemented in different simulators, or on the fly for specific experiments, it is difficult for other researchers to reproduce results or accurately compare new protocols to previously published work. Both of these are fundamental requirements of scientific research.

In this work we use models from Urick [22], Berkhovskikh and Lysanov [3], Coates [6], and Stojanovic [19] to accurately model the characteristics of the attenuation, ambient noise, propagation delay, bandwidth, and other physical layer characteristics for underwater acoustic communication.

3. THE NEED FOR A COMPLETE MODEL

Most of the current work in underwater protocol design has either implemented models in tools such as Matlab to run simple simulations (*e.g.*, Harris *et al.* [10]) or attempted to adjust parameters in current radio propagation and channel models to mirror the underwater environment (*e.g.*, by partially modeling the error rate and propagation delay without accounting for the distance–bandwidth relation [15]). However, it has been shown that such methods produce results that may not always accurately predict what happens in a realistic underwater environment [9].

On the other hand, the fact that in the recent literature on this topic several papers have presented simulation studies on networking issues clearly shows that there is a need for a standardized simulation model for this purpose. Such a model will facilitate accurate simulation and testing of underwater protocols while saving the need for each researcher to perform after-simulation processing or to re-implement the model in their own simulation environment.

Our contribution in this paper is the development of a set of functions that enable ns2 to accurately handle the underwater communication environment. While certain components of the underwater channel, such as the complex fading characteristics, require ray tracing methods that are computationally complex, and therefore currently left out of our ns2 model, we show that the model still provides accurate estimation of channel conditions and represents a good start towards a complete simulation model. This is a valuable step forward in unifying performance evaluation tools in underwater networking and protocol studies.

In the following section we present our ns2 underwater communications model in detail.

4. NS2 UNDERWATER MODEL

The ns2 simulator divides the channel and physical layer functions and characteristics into four components: Propagation, Channel, Physical, and Modulation. Figure 1 depicts this division, highlighting the characteristics within each component. The propagation component contains most of the characteristics of the signal propagation through the medium (including attenuation) and of the ambient noise. In addition to distance-dependent attenuation, in underwater channels the signal fading is also affected by the orientation of the link. This feature is also modeled in the propagation component. The characteristics exported to other components of the ns2 model include the calculation of the received signal strength and the interference range of a signal. The primary function of the channel model is to handle propagation delay calculations and to make use of the functions from the propagation model. The physical layer tracks energy consumption metrics and also calculates the transmission times. Unlike in radio models, where the bandwidth is assumed to be constant regardless of the transmitter-receiver distance and therefore no information for other layers is required, in an underwater network the link bandwidth does depend on the link length, and therefore bandwidth information from the propagation layer of ns2 must be exposed to other components. Finally, the physical model calls the modulation model to calculate bit error probabilities given a received signal strength, modulation scheme, and level of noise. It is interesting to note that no standard modulation schemes are currently used in the majority of ns2 simulations.

In the following subsections, we describe in detail the fundamental characteristics of the underwater channel and present our ns2 implementation of them. These characteristics include the propagation speed of sound in water, the bandwidth, noise, and attenuation for a transmission, the maximum interference distance of a signal, the received signal strength, and the bit error rate.

4.1 Underwater Propagation Model

In ns2, the Propagation models are responsible for calculating the signal-to-noise ratio at the receiver after attenuation and ambient noise are taken into account, as well as the interference range of a signal.

To use the underwater propagation model, it is only necessary to choose it in the TCL simulation script using the name “Propagation/Underwater”.

To calculate the signal-to-noise ratio (SNR) at the receiver and the interference range, both the attenuation of the acoustic signal in water and the ambient noise need to be accounted for. The total attenuation is calculated based on the spreading loss [22] and on Thorp’s approximation [3] for the absorption loss.

Therefore, the channel model includes a function to calculate Thorp’s approximation (See Figure 2) for absorption at a given frequency, which is as follows [3]:

$$10\log a(f) = \begin{cases} 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f} & \\ +2.75\cdot 10^{-4}f^2 + 0.003, & f \geq 0.4 \\ 0.002 + 0.11(f/(1+f)) & \\ +0.011f, & f < 0.4 \end{cases} \tag{1}$$

where $a(f)$ is given in dB/km and f is in kHz for underwater communications. Thorp’s approximation for absorption loss

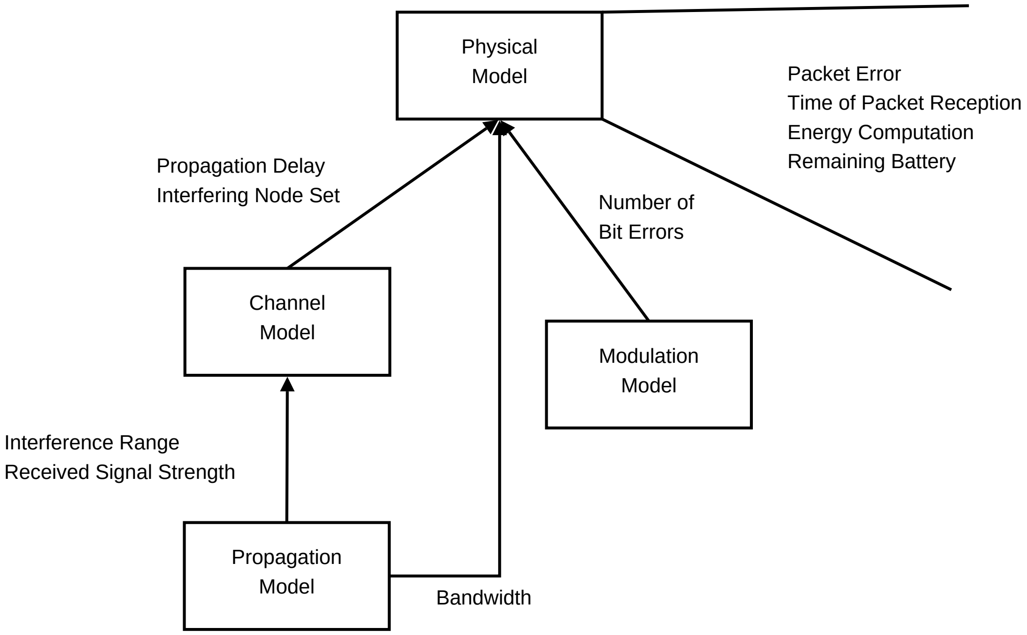

Figure 1: The ns2 channel and physical layer model

```
THORP($frequency$)
 1  $f \leftarrow$ POW($frequency, 2$);
 2  if $f > 0.4$
 3     then
 4          $atten \leftarrow 0.11 * f/(1+f)+$
 5                 $44 * (f/(4100 + frequency))+$
 6                 $2.75 *$ POW$(10, -4) * f+$
 7                 $0.003$;
 8     else
 9          $atten \leftarrow 0.002+$
10                 $0.11 * (f/(1+f))+$
11                 $0.011 * f$;
12  return $atten$;
```

Figure 2: Thorp's approximation

```
PR($transmitter, receiver$)
 1  $Pt \leftarrow transmitter \rightarrow$ GETTXPR();
 2  $distance \leftarrow$ CALCDIST($transmitter, receiver$);
 3  for $i \leftarrow 0$ to $NUM\_FREQ$
 4  do
 5       $AN[i] \leftarrow -(k * 10 *$ LOG10($distance$)$+$
 6            $distance *$ THORP($freq[i]$)$+$
 7            ORIENTATION($transmitter, receiver$)$+$
 8            LOG10(NOISE($freq[i]$)));
 9       if $AN[i] > AN[max\_index]$
10          then
11               $max\_index \leftarrow i$;
12  $Pr \leftarrow Pt + AN[max\_index]$;
13  return $Pr$;
```

Figure 3: SNR at the receiver

divides the frequencies into two groups, those under 400 Hz (see Figure 2, lines 2–7), and those over 400 Hz (see Figure 2, lines 8–11). We chose to use dB re μPa throughout our implementation, as this is a typical unit of signal strength in acoustic communications. Accordingly, all quantities are given using these units, and all tunable parameters (in terms of transmit power for example) should be given in dB re μPa as well.

Combining absorption effects and spreading loss, the total attenuation is as follows [22]:

$$10 \log A(\ell, f) = k \cdot 10 \log \ell + \ell \cdot 10 \log a(f), \tag{2}$$

where the first term is the spreading loss and the second term is the absorption loss. The spreading coefficient defines the geometry of the propagation (*i.e.*, $k = 1$ is cylindrical, $k = 2$ is spherical, and $k = 1.5$ is practical spreading [22]). This is used in the calculation of the SNR at the receiver (in a function that overloads the Pr function in ns2)in combination with the ambient noise calculation (see Figure 3 lines 5–6).

In addition to this attenuation, signal fading in the underwater environment is affected by the orientation of the link (*i.e.*, whether the link is horizontal or vertical). To account for this effect, we added a modifier function that takes the location of the sender and receiver and returns an additional attenuation factor that is combined to account for the total attenuation (Figure 3, line 7).

The calculation for the ambient noise in the underwater environment (see Figure 4) is divided into the major factors contributing to the total: *turbulence* (Figure 4, lines 1–2), *shipping* (Figure 4, lines 3–7), *wind* (Figure 4, lines 8–12), and *thermal* (Figure 4, lines 13–14). The following formulae give the power spectral density of the four noise components

```
NOISE($frequency, shipping, wind$)
$turbulence \leftarrow 17 - 30 * \text{LOG10}(frequency)$;
$turbulence \leftarrow \text{POW}(10, (turbulence * 0.1))$;
$ship \leftarrow 40+$
        $20 * (shipping - 0.5)+$
        $26 * \text{LOG10}(frequency)-$
        $60 * \text{LOG10}(frequency + 0.03)$;
$ship \leftarrow \text{POW}(10, (ship * 0.1))$;
$win \leftarrow 50+$
        $7.5 * \text{POW}(wind, 0.5)+$
        $20 * \text{LOG10}(frequency)-$
        $40 * \text{LOG10}(frequency + 0.4)$;
$win \leftarrow \text{POW}(10, (win * 0.1))$;
$thermal \leftarrow -15 + 20 * \text{LOG10}(frequency)$;
$thermal \leftarrow \text{POW}(10, (thermal * 0.1))$;
$noise \leftarrow$
    $turbulence + ship + win + thermal$;
return $noise$;
```

Figure 4: Noise approximation

in dB re μPa per Hz as a function of frequency in kHz [6]:

$$\begin{aligned} 10\log N_t(f) &= 17 - 30\log f \\ 10\log N_s(f) &= 40 + 20(s - 0.5) + 26\log f \\ &\quad -60\log(f + 0.03) \\ 10\log N_w(f) &= 50 + 7.5w^{1/2} + 20\log f \\ &\quad -40\log(f + 0.4) \\ 10\log N_{th}(f) &= -15 + 20\log f, \end{aligned} \tag{3}$$

where N_t is the noise due to turbulence, N_s is the noise due to shipping, N_w is the noise due to wind, and N_{th} represents the thermal noise. The overall noise power spectral density for a given frequency f is then:

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f). \tag{4}$$

Each component impacts the noise power spectral density differently at different frequencies. For example, in the frequency ranges encountered for transmission distances over 10's of meters, the turbulence and shipping components have very little effect, whereas the wind and thermal noise may be more significant. By default, in our modules both the shipping variable, s, and the wind variable, w, are set to 0. These variables are bound to TCL variables called *ship_* and *wind_* respectively and can be set in the usual way with lines such as:

*Propagation/Underwater set ship_* `value`

*Propagation/Underwater set wind_* `value`

where *ship_* can take values from 0 to 1 and *wind_*, which represents wind speed, can take positive values in m/s.

```
GETDIST($Pr, Pt, frequency$)
1  $distance \leftarrow 0$;
2  $target\_A \leftarrow Pr - Pt$;
3  while $A > target\_A$
4  do
5      $distance++$;
6      $A \leftarrow -(k * 10 * \text{LOG10}(distance)+$
7              $distance * \text{THORP}(frequency))$;
8  return $distance$;
```

Figure 5: Affected distance

The ns2 simulator has a node class that keeps information specific to each node in the simulation, including location coordinates (*x,y,z*) and transmit power settings. The node class also has a number of member functions used to access information about the nodes. The *Pr* function takes pointers to the two communicating nodes and is used by the Channel model in the calculation of packet loss probability. To find the attenuation for a given transmission between two nodes, the center frequency for the transmission must be found. This corresponds to the frequency that exhibits the best propagation conditions, for a specific distance between the communicating nodes. More specifically, consider the SNR as a function of frequency, which is given by

$$SNR = \frac{P_T}{A(\ell, f)N(f)}, \tag{5}$$

where $N(f)$ is given by Equation (4) and $A(\ell, f)$ is given by Equation (2). For a given value of the transmit power, P_T, the SNR is inversely proportional to the so-called attenuation–noise (or AN) factor [19] (in our code, the AN factor is represented by the variable AN, see Figure 3, which is defined as $-10\log(A(\ell, f)N(f))$).

In order to find the center frequency, the distance between nodes is calculated (see Figure 3, line 2). Lines 3–8 in Figure 3 calculate the AN factor for each of the possible frequencies for the transmission. As each AN value is calculated, the frequency with the lowest AN factor (largest value of the AN variable) is tracked (see Figure 3, lines 9–11). Finally, the AN factor that corresponds to that frequency is combined with the transmitted power to calculate the SNR at the receiver (Figure 3, lines 12–13) and is taken to be equal across the the frequency spectrum. More specific calculations could be used to include the frequency selective nature of noise and attenuation, the interference due to other transmissions, and the effects of fading.

The final function the propagation model is expected to perform for ns2 is defining the radius in which a transmission needs to be considered for interference with other nodes' transmissions. The function *getDist* takes a threshold received power level, the transmit power level and the frequency at which the signal was sent, and returns the largest distance that a node should be from the transmitter and still be considered interfered with by its transmission (see Figure 5). Essentially, this function finds the target attenuation that is needed to result in a received signal strength so low that it does not need to be considered for interference calculations (Figure 5 line 2). It then iteratively calculates the attenuation at distances starting at one meter until it finds the target factor. This function is only accurate to the closest meter but could easily be changed by editing line 5 if needed.

The propagation model is used by the channel model to make collision and transmission error decisions; therefore, it does not need to calculate propagation delay or bandwidth. These functions exist in the channel model, which is described in detail in the following subsection.

4.2 Underwater Channel Model

The channel model in ns2 maintains the node lists used

```
GET_PDELAY($transmitter, receiver$)
 1  $distance \leftarrow$ CALCDIST($transmitter, receiver$);
 2  $tZ \leftarrow transmitter.Z$;
 3  $rZ \leftarrow receiver.Z$;
 4  if $tZ < rZ$
 5     then
 6           $lZ \leftarrow tZ$;
 7           $hZ \leftarrow rZ$;
 8     else
 9           $lZ \leftarrow rZ$;
10           $hZ \leftarrow tZ$;
11  while $distance > 0$
12  do
13      SETDISTVAR($tempDist, z, lZ, hZ, t$);
14      $pdelay \leftarrow pdelay + (tempDist/$GETSPEED$(t, z))$;
15      $distance \leftarrow distance - tempDist$;
16  return $pdelay$;
```

Figure 6: Propagation delay

to calculate neighbor sets, collisions, etc. It is additionally responsible for calculating propagation delays. Essentially, the physical layer calls a *sendUp* function with a packet and a pointer to itself, and the channel model calculates neighbors that may be affected by the transmission as well as propagation delays and returns this information. Details on the exact functionality of the ns2 simulator can be found on the official website [13]. Aside from calling the appropriate propagation model functions, such as *getDist*, the ns2 channel model has to implement the propagation delay model as well, which is somewhat complex due to the dependency of the speed of sound on the depth of the water. In addition to the depth in the water, the propagation speed also depends on the temperature of the water, which in turn depends on the depth through a non-linear relationship. There are roughly five zones in which the temperature change in the oceans can be linearly approximated (see [21] for details). Then, within these zones, the underwater acoustic propagation speed in m/s (c) can be accurately modeled as follows [22]:

$$\begin{aligned} c \; &= 1449.05 + 45.7t - 5.21t^2 + 0.23t^3 \\ &+(1.333 - 0.126t + 0.009t^2)(S - 35) \\ &+16.3z + 0.18z^2, \end{aligned} \qquad (6)$$

where t is one tenth of the temperature of the water in degrees Celsius, z is the depth in meters, and S is the salinity of the water.

Figure 6 shows the propagation delay function used by the channel model. Essentially, the function takes segments of distance traveled depending on their depth and calculates distance traveled divided by the speed. When all of the segments of the path have been added together, the total propagation delay is returned. The *getSpeed* function on line 17 implements the approximation in Equation 6. The function *SetDistVar* takes the current values of the highest and lowest depth (z-variables) and returns the distance traveled in the next segment of linear temperature change, the average temperature in that zone and the updated values for the z-variables. This function is straightforward and omitted here.

To use the underwater channel model, it is only necessary to choose it in the TCL simulation script using the name "Channel/UnderwaterChannel". There is only one bound variable to set in the channel model, the salinity value for the water used in the propagation delay calculation. This value defaults to the average salinity in the world's oceans (35 parts-per-million) [14] and can be set to some other value (*e.g.*, 32 ppm) as follows:

Channel/UnderwaterChannel set salinity_ 32

The physical model uses information from both the channel model and the propagation model to calculate transmission times, total delays, and the success or failure of packet reception. The physical model is described in detail in the following subsection.

4.3 Underwater Physical Layer Model

In ns2, the physical layer model calculates the final statistics used in the simulation with respect to packet reception, including packet error, transmission time, and propagation delay. For most of these calculations, calls are made to functions in the channel and propagation models. Additionally, information about energy costs associated with the physical interface are stored and used to calculate residual battery charge and transmission energy costs. We leave all the specific parameters of interface energy consumption as bound variables to be set by the user, since they depend on the specific hardware being modeled. Additionally, the receive signal strength threshold and the maximum transmit power levels are interface specific and are set through bound variables. The default sets of parameters for the maximum transmit power, receive threshold, and the interface energy consumption parameters are set to model the WHOI micromodem [8]. Also included is the parameters set to model the Teledyne-Benthos modem [20].

To use the underwater physical model, it is only necessary to choose it in the TCL simulation script using the name "Phy/UnderwaterPhy". To set the maximum transmit power and the receive threshold, set the variables *Pt_* and *Pr_* respectively, both in dB re μPa.

The primary function of interest used in the physical layer is the calculation of the available bandwidth given the distance between the transmitter and receiver. Figure 7 shows the bandwidth function, which resides in the propagation model, but is described here since this is the only place where it is used. First, using the distance between the transmitter and receiver (line 1), the frequency experiencing the minimum attenuation–noise factor is found (lines 2–9). This frequency is used as the center frequency for communication. Then, we use the 3 dB definition of bandwidth to find the edges of the usable frequency band (lines 13–19). Finally, the bandwidth is calculated and returned (lines 20–21).

The physical layer also calls the modulation model to calculate effective bitrate and bit error rate, given the SNR and bandwidth used. The following subsection describes the modulation model.

4.4 Underwater Modulation Model

The Modulation model in ns2 is responsible for bitrate and bit error calculations based on signal strength and modulation scheme. The error probability is a function of SNR. The bitrate and number of bit errors is returned by the modulation model.

```
BANDWIDTH($transmitter, receiver$)
$distance \leftarrow$ CALCDIST($transmitter, receiver$);
for $i \leftarrow 0$ to $NUM\_FREQ$
do
    $AN[i] \leftarrow -(k * 10 *$ LOG10($distance$)$+$
            $distance *$ THORP($freq[i]$)$+$
            LOG10(NOISE($freq[i]$)));
    if $AN[i] > AN[max\_index]$
       then
            $max\_index \leftarrow i$;
$max\_AN \leftarrow AN[max\_index]$;
$freq\_center \leftarrow freq[max\_index]$;
$j1 \leftarrow 0; j2 \leftarrow 0$;
for $i \leftarrow max\_index$ to $0$
do
    if $max\_AN - AN[i] \leq 3.0$
       then
            $j1++$;
       else
            BREAK
for $i \leftarrow max\_index$ to $NUM\_FREQ$
do
    if $max\_AN - AN[i] \leq 3.0$
       then
            $j2++$;
       else
            BREAK
$bandwidth \leftarrow freq[j2] - freq[j1]$; return $bandwidth$;
```

Figure 7: Bandwidth available

5. MODEL VERIFICATION

To validate the implemented ns2 underwater model, we ran a number of simulations and compared the resulting values of specific parameters with those calculated using analytic models. Specifically, we wanted to validate the major characteristics of the model, which are: noise, AN factor, propagation delay, bandwidth, and transmission power to ensure there was no error in the final ns2 models. In each case, the simulated results exactly match the analytic model.

5.1 Noise and Attenuation

The noise and attenuation calculations are critical for accurate bandwidth calculation and for accurate received signal strength calculation. We placed two nodes between 100 m and 10 km apart and sent packets between them, modifying the code to output the noise approximation. The noise and AN factor accurately modeled the underwater environment. Figure 8 depicts the results for the noise in dB re μPa as a function of distance between nodes. While the power spectral density of the noise is not directly dependent on distance, the frequencies used for communication are. Therefore, the total noise affecting communication between nodes at a particular distance will vary.

5.2 Propagation Delay

Testing the accuracy of the propagation delay calculation requires a number of experiments since the result depends on the depth of the communication in the water. In all cases we compared the simulation results to calculated methods drawn from [21, 22]. We did a large variety of tests, varying both the depths of the nodes, and their distances apart. Additionally, to test the simplest cases, we used scenarios where the sender and the receiver were both at the same depth. In all cases, the simulation results modeled the propagation delays accurately. As an example, Figure 9 depicts the propagation delay to cross a distance of 1 km as the depth of the two nodes is varied (but equal to each other).

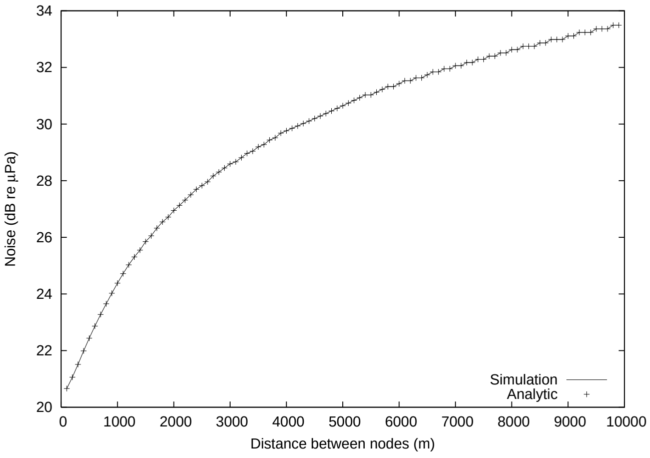

Figure 8: The effect of frequency (and hence distance) on noise

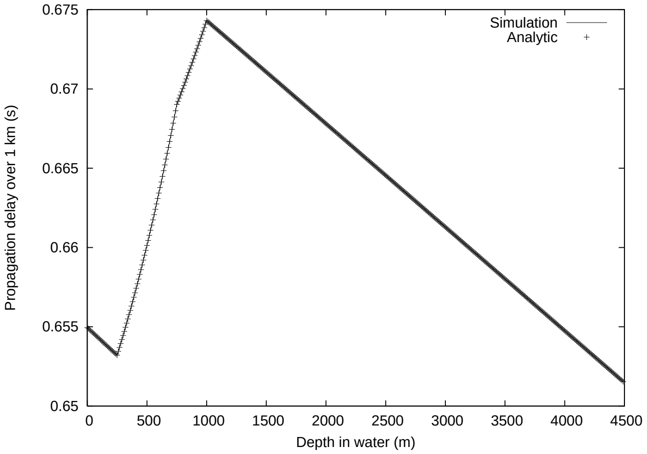

Figure 9: The effect of depth on propagation delay

5.3 Bandwidth and Transmission Power

Finally, the model needs to accurately predict the transmission power required to successfully meet the threshold receive power given the distance between nodes and the bandwidth available for communication. This section presents the experiments to test these two functions. In both experiments the distance between nodes was varied between 100 m and 10 km. Figure 10 presents the results from the bandwidth calculation tests. We omit the points from the analytic results to improve readability, but they matched exactly. The line depicts the center frequency, corresponding to the minimum AN-factor. The vertical bars show the bandwidth available around that center frequency. It is clear from the figure that, as distances increase, not only does the

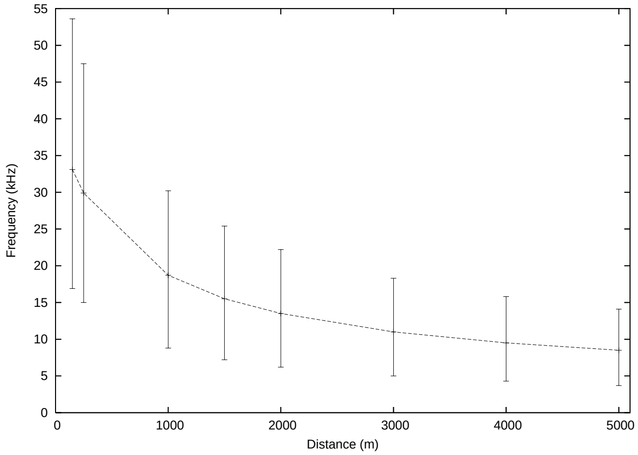

Figure 10: The effect of distance on center frequency and available bandwidth

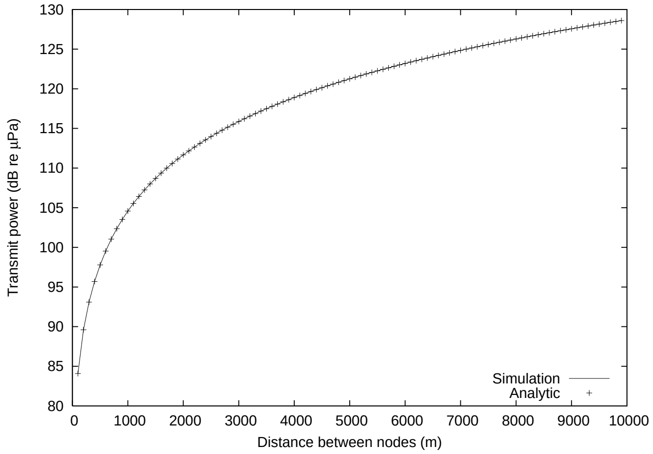

Figure 11: The effect of distance on required transmit power

center frequency shift, but also the bandwidth decreases. Figure 11 shows the transmit power needed to successfully get a signal strength of 20 dB re μPa at the receiver.

The following section presents a case study, demonstrating the use of the ns2 model for protocol analysis and design.

6. CASE STUDY

Our development of the ns2 model presented in this work was directly motivated by our need to test the design of a routing protocol. Due to the high energy costs of transmissions in underwater acoustic networks, and the larger available bandwidth on shorter links, using multiple short hops to traverse a path as opposed to fewer long hops has the potential to conserve energy. In [11], we present the design of a routing algorithm that attempts to choose the optimal hop lengths as packets are routed through a multihop underwater network. The analysis of this protocol strongly depended on the bandwidth–distance relationship, as well as an understanding of the interference due to other nodes in the network and of the propagation delays. No partial implementation of the underwater model would have sufficed to produce an accurate simulation study.

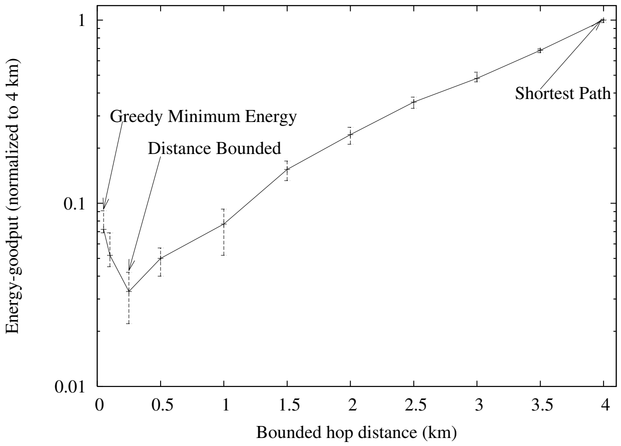

Figure 12: ns2 simulation: energy consumption vs. hop distance

As an example of the results obtained, Figure 12 depicts the average energy-goodput per flow on a log-scale. The figure represents the average of 25 simulations, each run for 300 seconds. The results, presented in detail in the original paper [11] show that our Bounded Distance protocol is more energy-efficient than a range of other protocols, including popular choices such as Shortest Path or Greedy Minimum Energy.

In addition to the use of the ns2 simulator, we used an implementation of the underwater channel previously developed in Matlab that, while able to accurately represent propagation and physical layer issues, failed to incorporate protocol issues such as collisions and multiple-access interference. The most significant point when comparing the results from the two simulators was that the energy savings due to the use of shorter hop distances was significantly underestimated by the Matlab simulator compared to the ns2 results, due to the absence of collisions and interference in the former. Using power control to reduce the transmission range also decreases the interference range. Without the entire model in the ns2 network simulator, our conclusions about the protocol performance would have been off by as much as 30%.

We can conclude that simulation code based on Matlab can accurately account for propagation issues and therefore can be suitable for physical layer studies, but is not adequate for networking evaluations, where protocol behaviors and interactions among multiple data flows need to be included. On the other hand, popular network simulators, such as ns2, provide insufficient support for advanced channel and physical layer descriptions and therefore may lead to inaccurate evaluations. The model we presented in this paper is a first step to fill this gap, making it possible to run complex network simulations without necessarily resorting to unacceptable simplifications of the physical reality.

7. CONCLUSIONS AND FUTURE DIRECTIONS

The drive to design and test protocols for the underwater acoustic environment is growing steadily due to the desire to monitor and explore the world's oceans. However, no standard model is available for researchers to use in the simulation phase of their work. In this paper, we have presented the design and implementation of our underwater acoustic model for the ns2 network simulator. Our model has four components: propagation, channel, physical, and modulation. We have described the critical functionalities of each of these layers with detailed descriptions of the algorithms used in each layer. In addition, we presented some simulation results and compared them to results obtained through analytic methods. This comparison validated the models, showing them to accurately approximate the underwater communication conditions. Finally, we presented a case study, demonstrating the use of the ns2 model to design and simulate an energy-efficient routing protocol. This case study further motivates the need for a complete underwater model for simulation studies.

Work towards accurate modeling of the complex fading and multipath effects in the underwater acoustic environment would round out the simulation model. Current best practices use ray tracing; however, such techniques are often too computationally complex for inclusion in network simulators such as ns2. Therefore, approximations are needed to support PHY and MAC layer protocol development that takes into account these characteristics of the acoustic channel. More complex modulation schemes could be added to allow more realistic simulation of error tolerance in the channel as well as the true available bit-rates given the bandwidth. Additionally, models of additional acoustic modems could be included to facilitate energy consumption analysis for energy-aware protocol design.

Finally, a full protocol stack, including MAC, routing, and transport layers, is required to round out the ns2 underwater simulation suite. However, research in designing protocols for each of these layers is just beginning. We believe that the existence of simulation tools for underwater networks that properly capture the essential behavior of acoustic propagation in the water will provide a valuable instrument for protocol design and evaluations, and will greatly help promoting research in this area.

8. ACKNOWLEDGMENTS

The authors would like to thank Robin Snader from the University of Illinois at Urbana–Champaign for his tremendous help in making the code correct.

9. REFERENCES

[1] I. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Elsevier's Ad Hoc Networks*, 3(3), 2005.

[2] M. Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001.

[3] L. Berkhovskikh and Y. Lysanov. *Fundamentals of Ocean Acoustics*. Springer, 1982.

[4] L. Betancur, R. Hincapie, R. Bustamante, and J. E. Sierra C. WiMAX Channel - PHY Model in Network Simulator 2. In *WNS2*, 2006.

[5] P. Casari, S. Marella, and M. Zorzi. A comparison of multiple access techniques in clustered underwater acoustic networks. In *Proc. IEEE/OES OCEANS*, Aberdeen, Scotland, June 2007.

[6] R. Coates. *Underwater Acoustic Systems*. Wiley, 1989.

[7] M. Feder and J. A. Catipovic. Algorithms for joint channel estimation and data recovery-application to equalization in underwater communications. *IEEE Journal of Oceanic Engineering*, 16(1):42–55, Jan. 1991.

[8] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms. http://www.whoi.edu, 2005.

[9] A. Harris III, D. Meneghetti, and M. Zorzi. Maximizing Channel Utilization for Underwater Acoustic Links. In *Proc. IEEE/OES Oceans*, Aberdeen, Scotland, June 2007.

[10] A. Harris III, M. Stojanovic, and M. Zorzi. When underwater acoustic nodes should sleep with one eye open: idle–time power management in underwater sensor networks. In *Proc. ACM WUWNet*, pages 105–108, Los Angeles, CA, Sept. 2006.

[11] A. Harris III and M. Zorzi. Energy–efficient routing protocol design considerations for underwater networks. In *Proc. IEEE SECON*, June 2007.

[12] M. Molins and M. Stojanovic. Slotted FAMA: a MAC Protocol for underwater acoustic networks. In *Proc. IEEE Oceans*, Singapore, Sept. 2006.

[13] ns2 Network Simulator. http://www.isi.edu/nsnam/ns/.

[14] Office of Naval Research. Ocean Water: Salinity. http://www.onr.navy.mil/Focus/ocean/water/salinity1.htm.

[15] D. Pompili, T. Melodia, and I. F. Akyildiz. Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks. In *Proc. ACM Mobicom*, 2006.

[16] W. Qian and J. A. Ritcey. Spatial diversity equalization applied to underwater communications. *IEEE Journal of Oceanic Engineering*, 19(2):227–241, Apr. 1994.

[17] M. Stojanovic. Recent advances in high-speed underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 21(2):125–136, Apr. 1996.

[18] M. Stojanovic. Retrofocusing techniques for high rate acoustic communications. *Journal of the Acoustical Society of America*, 117(3):1173–1185, Mar. 2005.

[19] M. Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. In *Proc. ACM WUWNet*, pages 41–47, Los Angeles, CA, Sept. 2006.

[20] Teledyne. Teledyne-benthos modem. http://www.rdinstruments.com/nemo.

[21] The National Center for Atmospheric Research. Temperature of Ocean Water. http://www.windows.ucar.edu/tour/link=/earth/Water/temp.html&edu=high.

[22] R. Urick. *Principles of Underwater Sound*. McGraw-Hill, 1983.