UNDERWATER ACOUSTIC COMMUNICATION (UWA) DATA PACKET SIZE OPTIMIZATION BASED ON THROUGHPUT

HAI SOCHEAT VIRAKRAINGSEI

INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNOLOGI PETRONAS

JULY 2010

Underwater Acoustic Communication (UWA)

Data Packet Size Optimization based on Throughput

by

HAI SOCHEAT VIRAKRAINGSEI

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Technology (Hons)
(Information & Communication Technology)

JULY 2010

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Underwater Acoustic Communication (UWA)
Data Packet Size Optimization based on Throughput

by

HAI SOCHEAT VIRAKRAINGSEI

A project dissertation submitted to the
Information & Communication Technology Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirements for the
Bachelor of Technology (Hons)
(Information & Communication Technology)

Approved by,

(Mr. Low Tang Jung)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
October 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

HAI SOCHEAT VIRAKRAINGSEI

គោរពចំពោះអ្នកម្តាយ និងលោកឪពុក

To Mom and Dad

ACKNOWLEDGMENT

First and foremost, I would like to acknowledge God for His continual blessings and guidance in my life. "*And whatever you do, whether in word or deed, do it all in the name of the Lord Jesus, giving thanks to God the Father through him.*" - Colossians 3:17.

I would like to thank my supervisor, Mr. Low Tang Jung for guiding me through the thesis process. It has been an honor to work under him. I would also like to thank my dear research team members, How Mei Le and Mohd Shafwan Bin Abdullah. They have given their full commitment in making this research a success. My special thanks to my dear friend, Pan Sophoana (*ប៉ាន សុភណ្ណា*), who has been helped me and given encouragement as well as his own resources.

Last but not least, my deepest thank to Universiti Teknologi PETRONAS and all of my lecturers who have given me the knowledge.

ABSTRACT

The aim of this project is to obtain a new method which will help to maximize the performance of the Underwater Water Acoustic Network. The metrics that will be used namely are Throughput, Bit Error Rate, and Energy. These three metrics are very crucial for the network life time. Throughput will be discussed heaver. Various mathematical methods such as interpolation will be address in order to justify between these three metrics. The main contribution of this project is the algorithm and the lookup table construction developed for selecting optimal packet size in underwater communication efficiency.

Table of Contents

List of Tables

List of Figure

Abbreviations

UWA	Underwater Acoustic
NS	Network Simulator
MATLAB	Matrix Laboratory
MIRACLE	Multi InteRfAce Cross Layer Extension for ns
IEEE	Institute of Electrical and Electronics Engineers
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
BER	Bit Error Rate
PER	Packet Error Rate
ALOHA	ALOHA protocol
CSMA	Carrier Sensing Multiple Access
DACAP	Distance Aware Collision Avoidance Protocol
Tcl	Tool Command Language
ARQ	Automatic Repeat-reQuest Protocol
EPUB	Energy per useful bit
FEC	Forward Error Correction scheme
ACK	Acknowledgement packet used in CSMA protocol

CHAPTER 1

INTRODUCTION

1.1 Background of Study

The underwater acoustic (UWA) have been used since the early 20^{th} century. The first application was to detect icebergs by using sonar waves. Because of the later advancement, the military started using underwater acoustic network in detecting submarine and mines. It also has been used widely for seafloor imaging, object localization and tracking, data communication for ocean exploration and management of coastal areas and environment application [1]. Environmental application includes monitory of physical indicator (salinity, pressure, and temperature), chemical/biological indicator (bacteria levels, and contaminants levels, and dangerous agent in reservoirs and aqueducts), and sea floor activity (earthquake activity, warning signal of tsunami).

Because of its potential to support a large variety of application, the underwater acoustic communication has been a challenge for many researchers for decades. It is just a beginning in a research field. Compare to its counterpart networks, the underwater acoustic network provide a better motivation in deployment. They eliminate the need of cables and they do not interfere with shipping activity

Although underwater acoustic network provide the ease in the deployment, there exist many challenges. Cost and limited battery resources are the main challenges in deploying such a network. Thus, many researches aim to improve or enhance the underwater acoustic

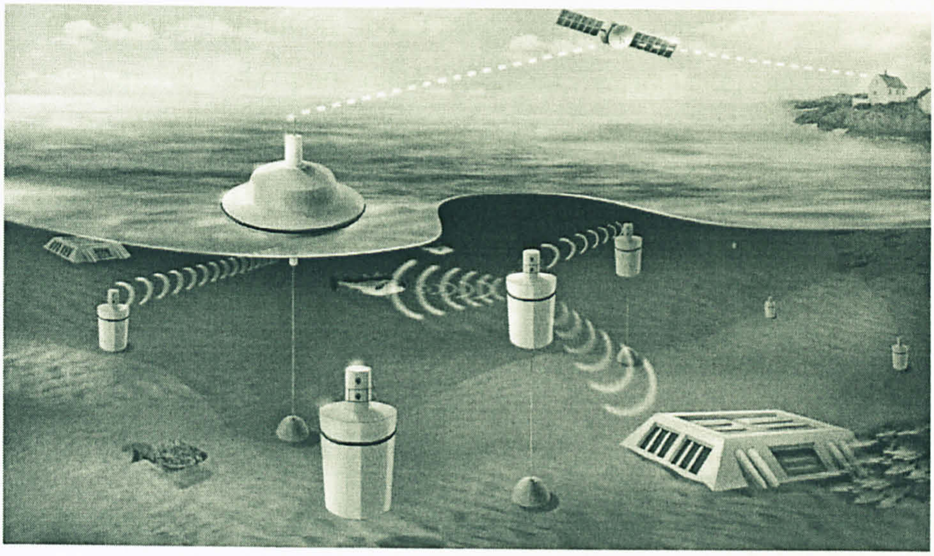

Figure 1 Underwater Network Topology

communication efficiency by focusing communication protocol, routing protocol, network topology which is result in maximizing the battery lifetime while minimizing power consumption ratio (PCTR). There seems a less focus on data packet size optimization. Data packet has a great influence on the performance of such network.

With the aim of improving the performance of the UWA, data packet size optimization is what this project is focus on. In UWA community, there is less research about it. By the end of this project, an algorithm plus many methods will be proposed for the further enhancement of UWA network performance based on *Throughput*.

1.2 Problem Statement

The underwater acoustic network (UWA) has been on research for decades because of the important roles in applications for oceanographic data collection, environment application (pollution monitoring, chemical changes, and so on), offshore exploration, distaste prevention (tsunami, earthquake, and sea floor activity), navigation, and tactical surveillance applications and in military purpose (unmanned underwater vehicle such as UUV and AUV, and submarine vehicle). By seeing its significant roles, the underwater acoustic network needs to be developed and researched in order to maximize of its potential.

There are many challenges in deploy such network. The most challenging are:

- *Cost*: It is costly to deploy such network because the hardware and the technology that is being used. The reason is because the limitation of electromagnetic communication system and the high rate absorption of electromagnetic signals in sea water/water [1].
- *Limited battery resource*: The network will be deployed in underwater, so each node's battery life time is very crucial. The operation in changing the battery is very hard and costly. The solar power cannot be deployed to recharge the equipment [1].
- *Performance*: We need an accurate data. To achieve that we have to design a special topology, protocol and so on in order to obtain high level of accuracy and in real time. The performance of the network also has a great influence on battery level.
- *Channel Impairment*: The underwater channel is severely impaired, especially due to multi-path and fading [4].

This research is mainly focus on the improvement on the performance of the UWA network. Most of recent researches are focusing on the communications protocols or routing protocols. Data packet optimization helps to increase the performance of the network which also helps to save the energy used by each node in the network. There are many parameters that help to improve the performance of UWA channel such as transmission range, bit rate, error probability type of protocol, energy per useful bit and modulation used, etc.

1.3 Objective

The main objective of this research is to seek effective solution for data packet size optimization in the UWA communications in the context of shallow tropical water (depth of 50m to 200m) with a transmission range from 200m to 2km. To achieve that, a new data packet size optimization mechanism or algorithm for UWA will be designed and implemented by taking into consideration the various unique characteristics of the UWA.

The specific objectives are:

1. To design and implement a new data packet optimization mechanism or algorithm for UWA communication by taking various unique characteristic of the UWA channel in the context of shallow tropical waters.
2. To investigate the corrections between data packet size and the various unique parameters of UWA channel thus indentifying high impact factors for improving UWA channel performance.
3. To identify the optimal packet size based on *Throughput*.
4. To compare the outcomes against other similar works accomplished by the UWA community.

1.4 Significant of the Project

After the research, the following output will be expected:

- Algorithm : The data packet optimization algorithm for UWA Communication
- Theories: Data Packet Size Optimization theory in the context of UW data transmission for the unique UWA channel characteristics/metrics.
- A look-up table/graph or a database which can be installed into UWA modem for optimal data packet size selection.

1.5 Scope of the Project

This project is mainly focus on the data packet size optimization in the UWA communication in the context of shallow tropical waters (depth of 50m to 200m) with the transmission range from 200m to 2km. Fist we will test it on NS2 (network simulation) and then we will have a test in laboratory.

CHAPTER 2

LITERATURE REVIEW

2.1 What is *Throughput?*

The throughput efficiency is the ratio of delivered bit rate and the total number of transmitted bit. This is the very basic of throughput.

Throughput = ND/NT

Where: ND is the number of delivered bits;

NT is the total number of bits sent.

Past research [10] has conclude that throughput efficiency is greater with less bit transmitted; given a channel that can transmit 200bps to 2000bps signal provided BER set up to 10-6. To summarize, slower bit transmission (at least 200bps) offer about 90% successful deliveries and the consideration is less collision happen during the transmission. An increase in the packet size transmitted then deteriorates the throughput efficiency. In faster transmission, 2000bps, the result show lower efficiency although it in the small packet size.

2.2 Data Packet Size and Throughput Efficiency

Based on data link layer, the format of an packet size is as in Figure 2 below and assume that each data packet consists of a total of N composes of N_l data bits plus the N_{oh} data packet overhead bits $(\alpha + \tau)$. So the Total is:

$$N = N_l + N_{oh} \quad (1)$$

Figure 2 Data Packet Format

At a minimum, N_{oh} equal the number of bits used for trailer (checksum). If the packet is transmitted by a bit rate of R then the packet duration, T_p can be evaluated as $T_p = NT$, where $T = 1/R$ is the duration of a single packet bit. In general, a synchronization preamble is added to the packet when a packet is transmitted. Let the duration of this preamble be $\boldsymbol{T_{syn}}$.

Now the propagation delay in a communication link can be expressed as $T_d = l/c$, where d is the distance between the transmitter and the receiver. c is the nominal speed of sound under water with a nominal value of 1500 m/s. Therefore, the total time needed to transmit a group of g packets and reception of the corresponding group of acknowledgments can be written as,

$$T(g) = g(T_p + T_{ack}) + T_w \qquad (2)$$

Where $T_w = 2(T_{syc} + T_d)$ is the total waiting time in the stop-and-wait protocol. It is noted that in a normal communication the duration of an acknowledgment is usually negligible with respect to the packet duration i.e. $T_{ack} << T_p$. For best efficiency, the time-out of the stop-and-wait protocol in transmitting a group of g packets should be equal to the round-trip time $T(g)$.

So, with a given a set of physical layer parameters (P_e, R, d) where P_e is the probability of packet error, R is the bit rate, and d is the distance between transmitter and the receiver; the throughput efficiency can be written in the form of [20],

$$\eta = (1 - P_e)^{N_l + N_{oh}} \frac{N_l}{N_l + \mu} \qquad (3)$$

Where,

$$\mu = N_{oh} + \frac{T_w R}{g} = \mu_o + \frac{2}{gc} lR \qquad (4)$$

It can be seen that the η expression is a function of packet length N_l. Therefore the optimal value of throughput efficiency can be evaluated by differentiating η with respect to N_l and equate it to zero i.e. $d\eta/d_{N_l} = 0$, and the optimal packet size is obtained as,

$$N_{opt} = \frac{\mu}{2}\left[\sqrt{1+\frac{4}{\mu\rho}} - 1\right] \quad (4)$$

Of which $\rho = \ln\frac{1}{1-P_e}$

With the optimal packet size obtained, thus the optimal throughput efficiency is,

$$\eta_{opt} = (1-P_e)^{N_{opt} + N_{oh}}\left(\frac{N_{opt}}{N_{opt} + \mu}\right) \quad (5)$$

2.3 Energy Efficiency

In data communications, energy is consumed during transmission of the data energy expended at the transmitter) and when framing and error correction is performed. So in communication energy can be generally taken as the sum of the energy required to transmit the data and the energy required to perform encoding and decoding of the data. Therefore the *Energy Efficiency* can be expressed as below:

$$\eta = \frac{k_1 N_l}{k_1(N_l+N_{oh}+\tau)+k_2+E_{dec}}(1-PER) \quad (6)$$

Where $(1 - PER)$ is the packet acceptance rate i.e. the data reliability rate.

For initial simplicity, optimal packet size is derived without error control i.e. τ and E_{dec} are considered as 0 (τ is the packet trailer bits, E_{dec} is energy needed for decoding) and a packet is said to be erroneous when one or more data bits are in error. With independent bit errors, a packet can be reliably received with a probability of $(1-p)^{l+\alpha}$ where p is the raw channel bit error rate. With these, the efficiency equation can now be written as,

$$\eta = \frac{k_1 N_l}{k_1(N_l+N_{oh}+\tau)+k_2+E_{dec}}(1-p)^{N_l+N_{oh}} \qquad (7)$$

This equation allows us to derive the optimal packet length l with respect to η. By taking the derivative of $d(\eta)/dN_l$ and equates it to 0 the optimal packet length is derived as,

$$N_{l_{opt}} = \frac{\sqrt{C_0^2 - \frac{4C0}{\ln(1-p)}} - C_0}{2} \quad \text{Where } C_0 = \alpha + k_2/k_1$$

2.4 Bit Error Rate (BER)

One of the changes that digital communications systems has brought to wireless transmission is the need for good end-to-end performance which is usually quantified by the bit error rate (BER). It quantifies the reliability of the entire radio system. BER starts off as a simple concept with a definition of,

$$BER = N_E/N_T \qquad (9)$$

Where N_E is the number of error bits and N_T is the total number of bits sent.BER is considered insignificant if a strong signal can be relayed through an unperturbed communication link.

However it cannot be ignored when the link is imperfect or noisy and a certain level of signal-to-noise ratio needs to be maintained over the link. In ARQ when it is used over relatively high BER links their performance is sensitive to the packet size. This implies that there is a need in choosing a correct packet size based on BER. The optimal packet size for ideal Selective Repeat (SR) ARQ scheme is given by [11] as,

$$k_{opt} = \frac{-h\ln(1-p)-\sqrt{-4hln(1-p)+h^2\ln(1-p^2)}}{2\ln(1-p)} \quad (10)$$

Where p is the known BER and h is the overhead bits per data packet.

2.5 NS2 and NSMIRACLE

Network simulator is simple an event-driven simulator tool that is widely known for studying the dynamic nature of communication network. NS2 is an object simulation consists of two languages which are C++ and OTcl (Object Oriented Tool Command Language). C++ mainly used to define the internal mechanism of simulation object, and OTcl uses to set up simulation by assembly and configuring the object as well as scheduling decret event [23].

NSMIRCALE is a set of add on library designed to enhance the functionalities provided by the Simulator ns2. NS-miracle provides an engine for handling cross-layer message and, at the same time, enables the coexistence of multiple modules within each layer of the protocol stack. The NS-MIRACLE framework facilitates the implementation and the simulation of modern communication system in ns2; moreover, due to its modularity, the code will be portable, re-usable and extensible.

CHAPTER 3

METHODOLOGY

The proposed project comprises the following phases:

3.1 Phase 1 – Review all and conduct all related works on Underwater Acoustic (UWA) Communication

Write a literature reviews and analysis on UWA channel modeling and data packet size optimization. It is a solid document that helps the research in the related area and scopes. The review will be focus channel modeling which is a tool to support the optimization techniques or approaches. The outcome of this phase would be the preliminary framework of the intended optimization algorithm

3.2 Phase 2 – Searching and identification of tools and parts for UWA setup.

The new optimization mechanism will be designed by using the framework and model that has been worked out from phase 1 as basis. Therefore, searching and identifying the relevant tool and development kits both software and hardware is necessary. The components must be ready for the subsequence tasks.

3.3 Phase 3 – Design and development of optimization algorithm

This is the most important part of the research because the new optimization technique is to be designed and modeled, and analyze. Moreover, the traditional system design cycle/approach would be adopted. Simulation will be used in order to detect and rectify the new optimization technique. This will be enabling the researcher to determine the architectural constraints, limitations and the effectiveness of the solution addressed earlier.

3.4 Phase 4 – Evaluation, testing and better enhancement.

Compare and evaluate the new designed optimization technique to any known similar or related works by the other researchers in the UWA community. It would be a critical benchmarking and a comprehensive report would be conducted. Phase 2 and 3 would be repeated if there are any shortcoming, limitations, constrains, error, etc discovered. At the end of this phase, a detail report would be produced.

CHAPTER 4

RESULT & DISCUSSION

4.1 Development Tool

In order to obtain the optimal packet size base on *Throughput*, various tools below are used to carry out the simulation.

Operating platform : Ubuntu 9.10
Development platform: ns-2.34, MATLAB 2009a, ns-Miracle, Underwatermiracle.
Tools : Low-power acoustic modem, underwater, speaker, hydrophones, sound mixer, water tank, server

4.2 Node Model and Configuration

For the simplicity, the performance of the network is measured between two nodes. The configuration of the node is carried out in 3 layers (Media layers) of OSI Model. They are Network layer, Data link layer and Physical layer. The figure 3 below shows the node configuration.

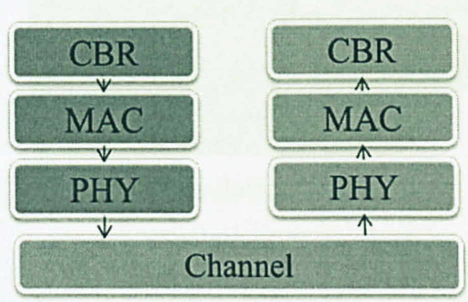

Figure 3 Node Model

The descriptions of each layer are:

- CBR (Constant Bit Rate) with specific *Bit Rate*
- MAC[1] : Using MPHY/BPSK/Underwater PHY
- Channel: Underwater Chanel

There are two MAC protocols that will be used. They are **ALOHA** and **CSMA**.

4.3 Simulation Scenario and Settings

Figure 12 shows the general scenario of the underwater environment set up for simulations. A cluster of 100 nodes is placed in the middle of a body of water with a dimension of 2km x 2km x 200m. This is to avoid reflection effects near the water surface and the water bottom. The depth of 200m is chosen to simulate the shallow water environment. One sink collects data packets from other nodes. Distance range between the sink and a node is 100m to 1km. The maximum transmission range of the nodes is to be 1km.

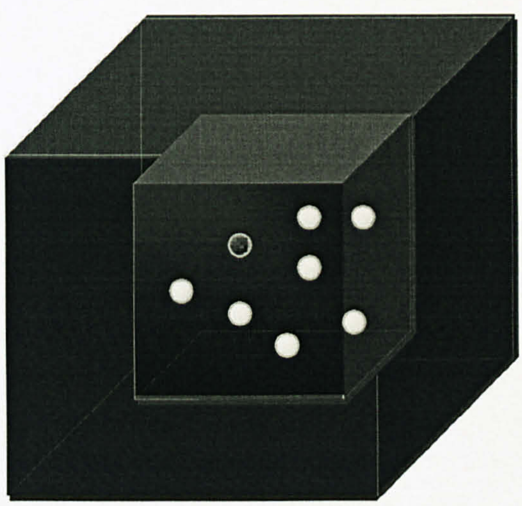

Figure 4 Simulation Scenario

[1] MPHY/BPSK/Underwater PHY is a special classes that help to simulate the real underwater environment

There are many parameters that we have to define the value before we carry out the simulation. In Table 1, we list down all the parameter across the scenarios in both MAC layers.

Property	Value
Transmission Power	5.2481e13
Bandwidth	6e3
Frequency	8.2e3
Queue Size	5
Link Delay	0.01

Table 1 Simulation Parameter

4.4 Experimentation

The tools used are two SAM-1 Miniature Acoustic Modems, two servers, a DolphinEAR hydrophone and one Aquasonic AQ339 Underwater Speaker. An application is developed in Visual Basic to capture the data sent and received by the acoustic modems. Since we don't much control over the data link layer, we made the CSMA protocol in Application layer.

Modem: Range: 250m – 1000m

Frequency: 7 Hz to 22 000 Hz

The spectrogram is for the measure the noise to calculate the BER.

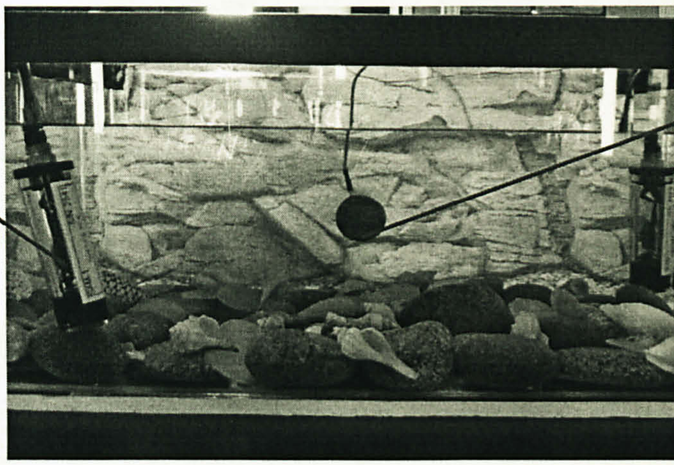

Figure 5 UWA Experiment in Aquarium

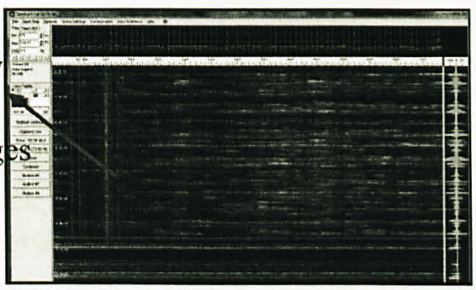

Figure 6 : Spectrum Lab

Figure 7 : Experiment at pond

The application directly controls the serial port of both the transmitter and the receiver. By implementing CSMA protocol, when there is data to send, the transmitter will send out a RTS (Request to send), if the channel is clear, the receiver will send out a CTS (Clear to send). The transmitter will then send the data. If no CTS is received after a specified timeout and number of trials, the data packet is discarded.

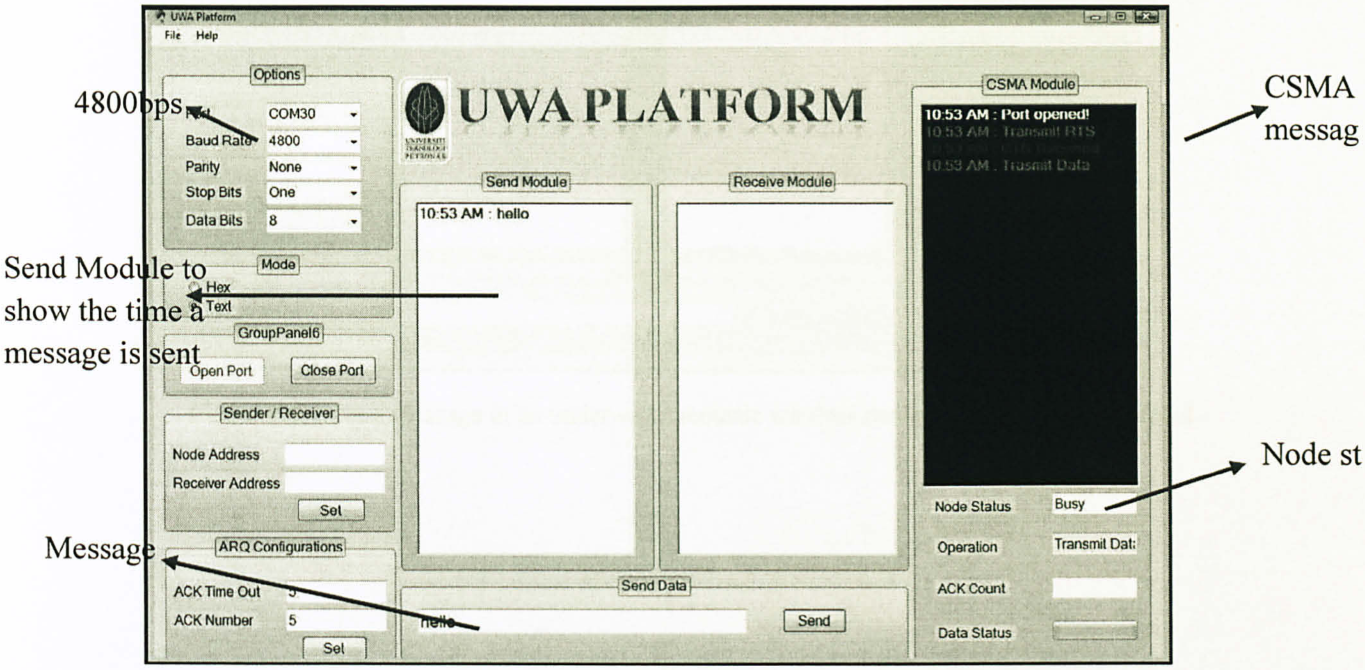

Figure 8 Send a message in an underwater acoustic wireless network using CSMA protocol

This same application is used in another laptop to view the received messages. Since the link is half duplex, only one party can send or receive at any point of time else the network will be jammed. Before launching, the laptop must be connected to the acoustic modem via a serial-to-USB converter.

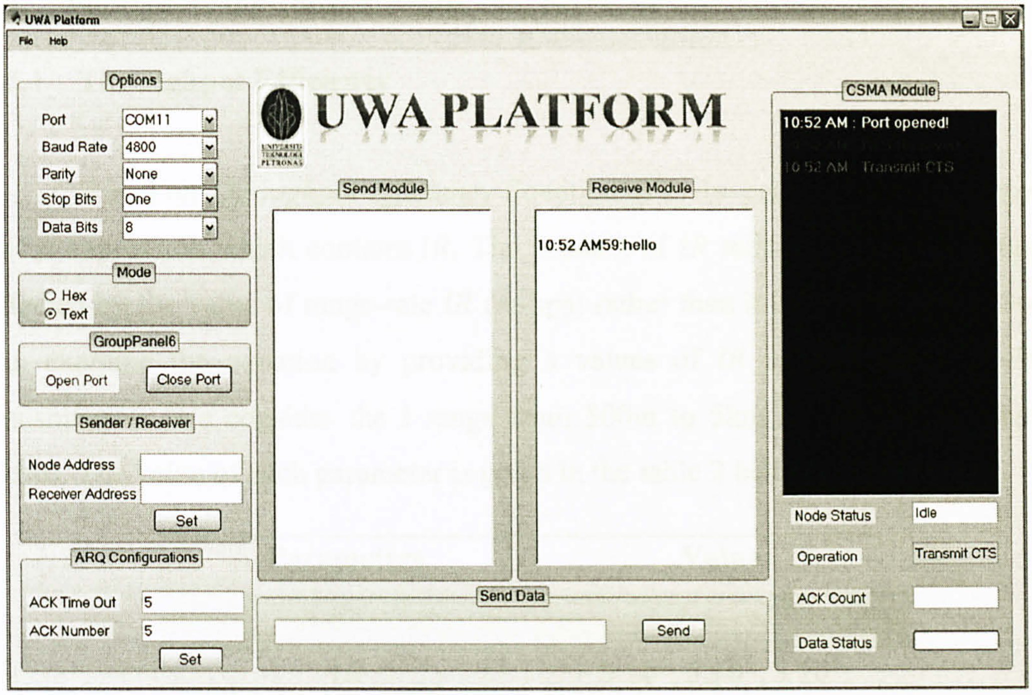

Figure 9Receive a message in an underwater acoustic wireless network using CSMA protocol

4.5 Simulation and Data Analysis

4.5.1 Throughput Efficiency

Based on *Throughput Efficiency* Equation of 2, the performance of it depends on μ expression which contains lR. The product of lR tells us that the equation is effected by the value of range-rate lR (m-bps) rather than it individual value. So we can examine the equation by providing 3 values of lR represents the low/high transmission. We consider the l range from 500m to 5km and R from 100bps to 1kbps. The value of each parameter is given in the table 2 below.

Parameters	Value
Pe	$10^{-3}, 10^{-4}$
lR	$5.10^4, 5.10^5, 5.10^6$
N_{oh}	8bits
T_{sync}	$16T$
g	1

Table 2 Throughput Efficiency Parameters

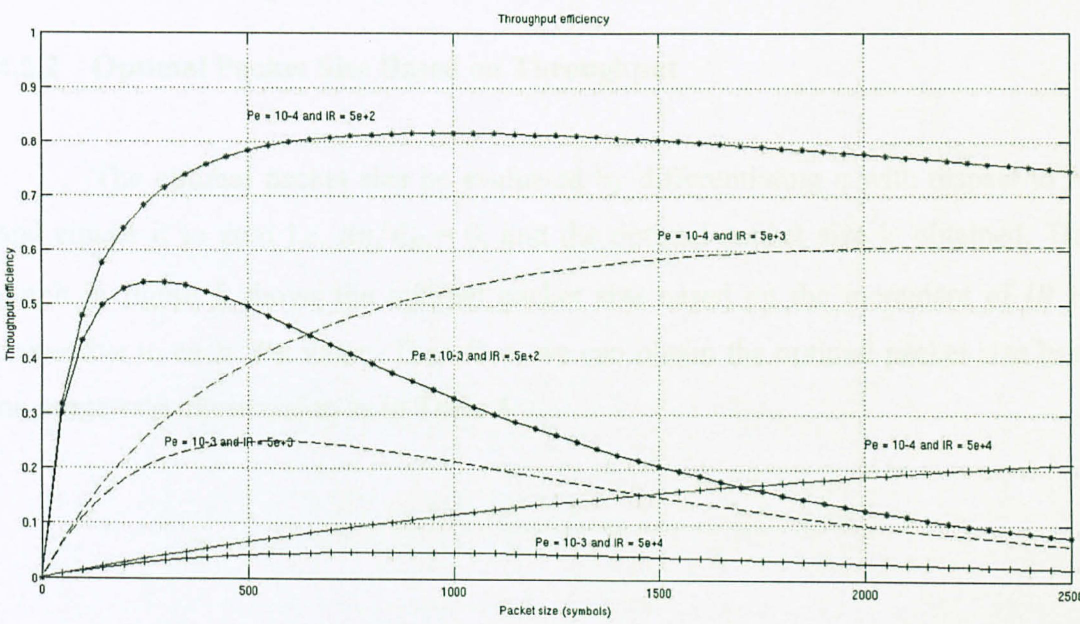

Figure 10 Throughput Efficiency

We can easily identify the packet size of each graph based on the optimal throughput. The packet size that we get is the optimal packet size. The table 3 below list all the optimal packet size based on each graph.

	Pe =10e-4			***Pe=10e-3***		
	$lR = 5.10^4$	$lR = 5.10^5$	$lR = 5.10^6$	$lR = 5.10^4$	$lR = 5.10^5$	$lR = 5.10^6$
Throuput Efficiency	*1050*	*2500*	*2500*	*300*	*550*	*900*
Packet Size	*0.8166*	*0.667*	*0.2133*	*0.5421*	*0.2504*	*0.0477*

Table 3 Optimal Packet Size from Throughput Efficiency

The graph also tell us that at certain Pe and LR and after the optimal througput efficiency, the throughput efficiency drop down slowly and vice versa. So there are some condition that some range of packet size can be considered as optimal packet size. The reasons why we need to have range of optimal packet size is because later we will put all the three parameters together (BER and Energy Efficiecny) and different between each optimal point will be able to justify and therefore all of three parameters can be optimized.

4.5.2 Optimal Packet Size Based on Throughput

The optimal packet size be evaluated by differentiating η with respect to N_l and equate it to zero i.e. $d\eta/d_{N_l}= 0$, and the optimal packet size is obtained. The graph in figure 6 shows the optimal packet size based on the increment of lR in respective to each $\boldsymbol{Pe}$ value. Therefore, we can obtain the optimal packet size base on range-rate transmission as in Table 4.

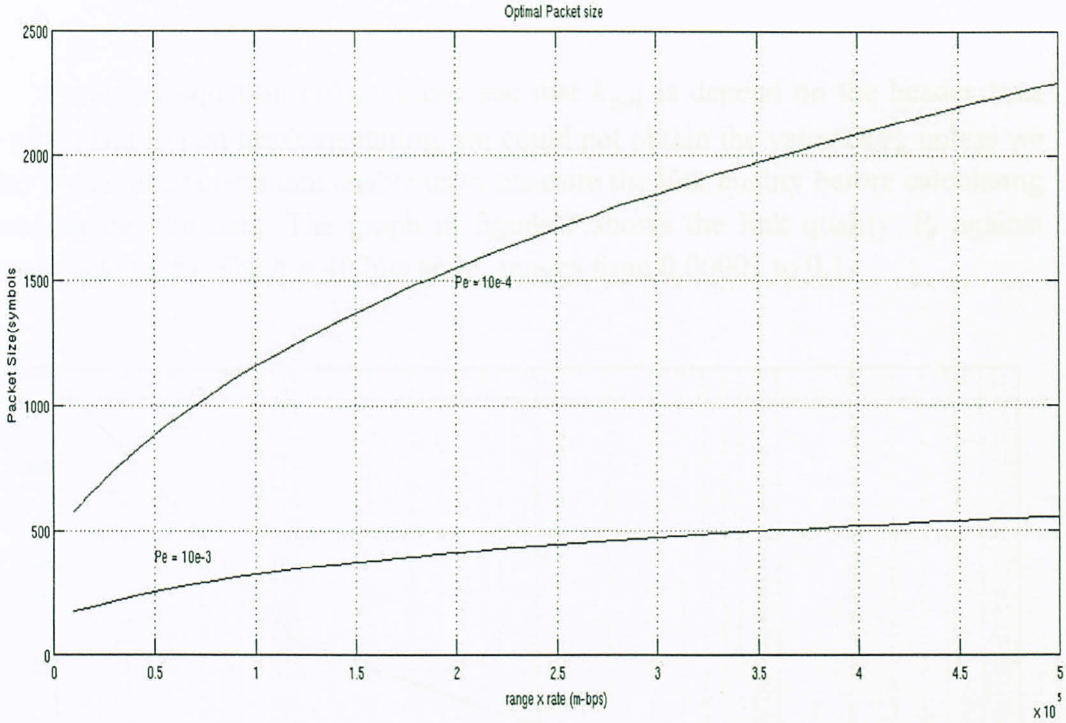

Figure 11 Optimal Packet Size based on Throughput

lR	$P_e = 10e-4$	$P_e = 10e-5$
9.0e+4	1106	310.9
1.3e+5	1290	354.1
1.6e+5	1490	381
2.2e+5	1615	425.6
2.8e+5	1792	461.8
3.6e+5	1997	501.3

Table 4 Optimal Packet Size from Graph

4.5.3 Optimal Packet Size Based on BER[2]

From the equation (10), we can see that k_{opt} is depend on the header h as well as P_e. But in real implementation, we could not obtain the value of P_e unless we deploy some mechanism that enable us to measure the link quality before calculating or sending out the data. The graph in figure 7 shows the link quality P_e against optimal packet size. The h = 40 bits andP_e ranges from 0.00001 to 0.1.

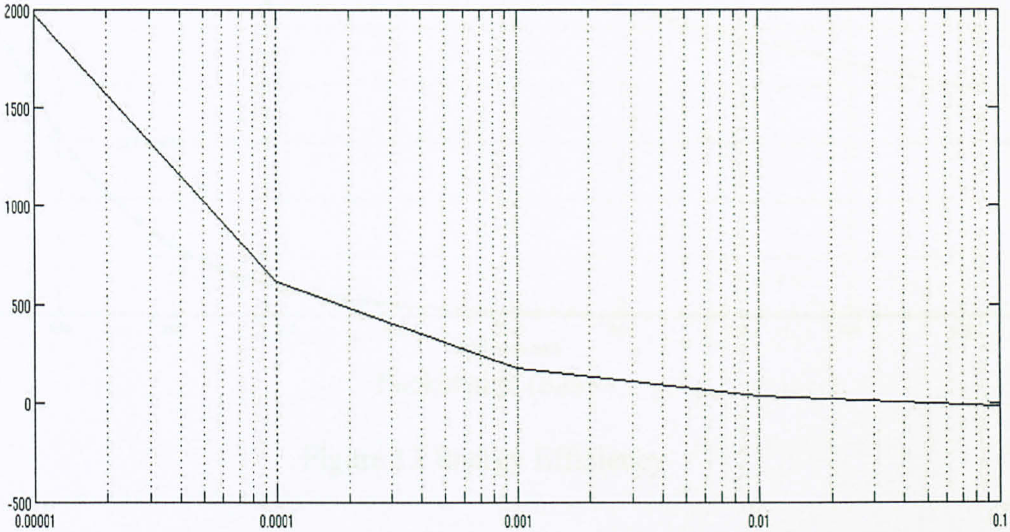

Figure 12 Optimal Packet Size based on BER

P_e	k_{opt}
0.00001	**2000**
0.0001	**621.1**
0.001	**178.9**
0.01	**39.87**
0.1	**0**

Table 5 Optimal Packet Size based on BER

[2] This is a join research topic. The Optimal packet size based on BER is researched by Mr. **Mohd Shafwan Bin Abdullah.**

4.5.4 Optimal Packet Size Based on Energy Efficiency[3]

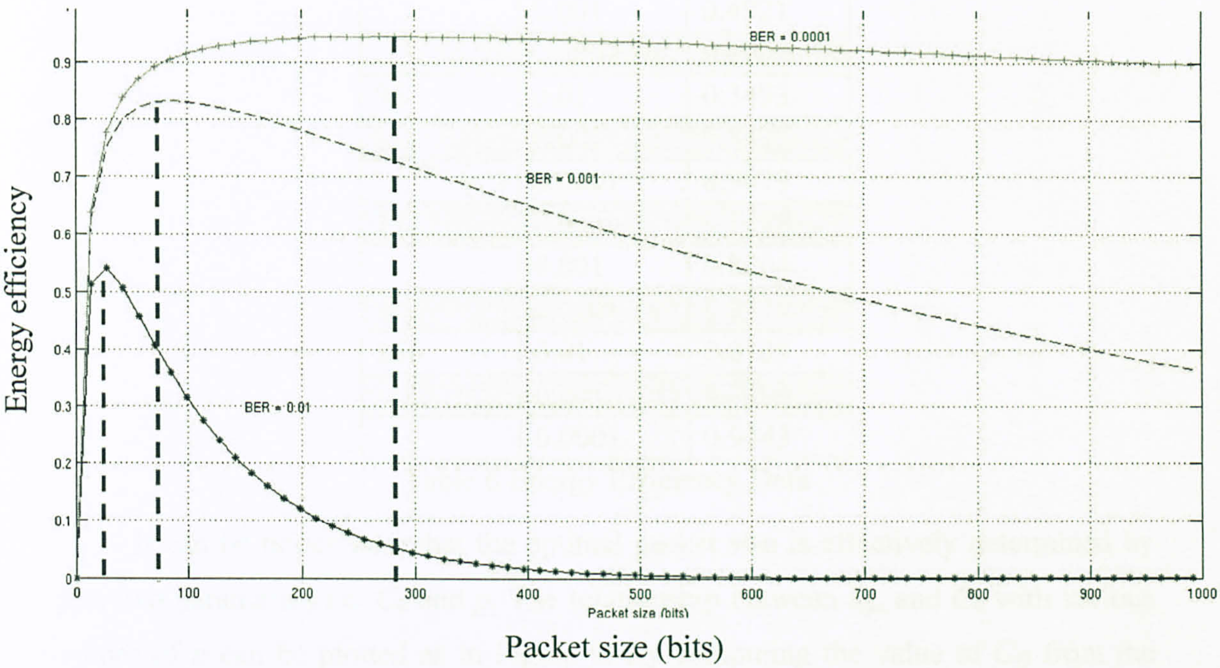

Figure 13 Energy Efficiency

Figure 8 shows the relationship between energy efficiency and packet size under different link qualities, quantified with bit error rates. The table 6 lists all the optimal value that obtains from Figure 8.

The same as Throughput Efficiency, by taking the derivative of $d(\eta)/dl$ and equates it to 0 the optimal packet length is derived as,

$$l_{opt} = \frac{\sqrt{C_0^2 - \frac{4C0}{\ln(1-p)}} - C_0}{2}$$

where,

$C_0 = \alpha + k_2/k_1$

---

3 This is a join research topic. The Optimal packet size based on Energy Efficiency is researched by Ms. **How Mei Le**

Packet size (bits)	Bit error probability	Energy Efficiency
16	0.01	0.4257
	0.001	0.4921
	0.0001	0.4992
96	0.01	0.3493
	0.001	0.8327
	0.0001	0.9079
176	0.01	0.1628
	0.001	0.8004
	0.0001	0.9379
256	0.01	0.0739
	0.001	0.7499
	0.0001	0.9443

Table 6 Energy Efficiency Data

It can be notice here that the optimal packet size is effectively determined by just two parameters i.e. C_0 and p. The relationship between l_{opt} and C_0 with various values of p can be plotted as in Figure 9. By computing the value of C_O from the radio equipment parameters and getting the BER, p from the channel, this plot may allows us to obtain the approximate optimal packet size for a reasonable range of radio parameters k_1 and k_2 with some header bit of $\alpha = 16$ bits.

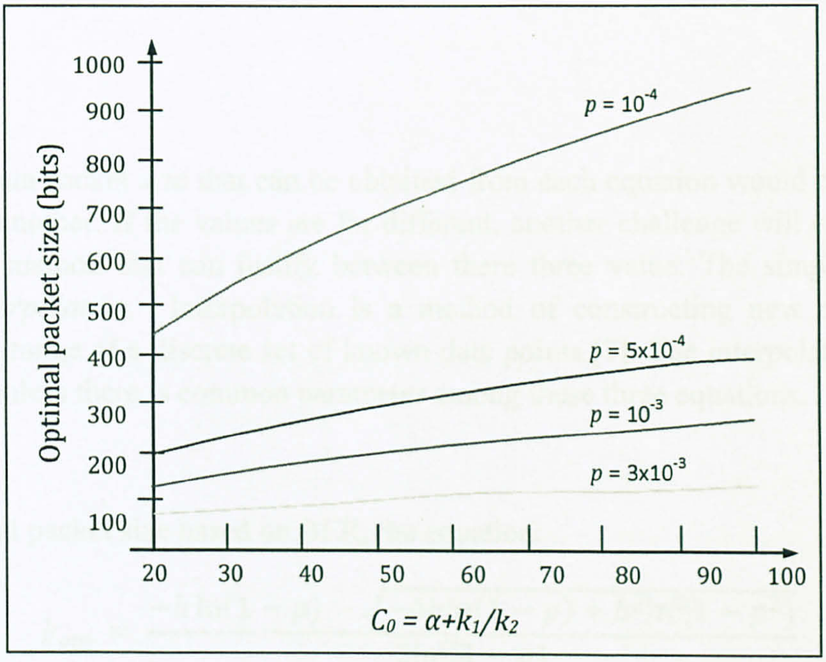

Figure 14 Optimal Packet Size based on Energy Efficiency

4.6 The Optimal Packet Size based on Three Parameters (Throughput, BER, and Energy Efficiency)

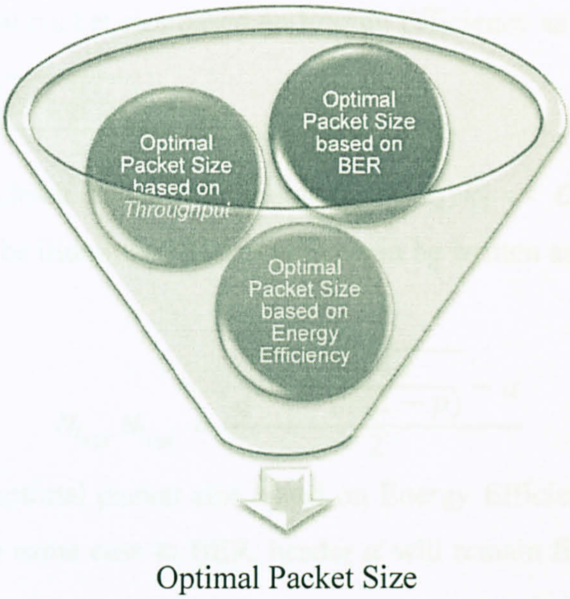

Figure 15 Interpolation of three matrices

Figure 10 would best describe the final optimal packet size that one should send in order to optimize all the three parameters.

Optimal data packet size that can be obtained from each equation would be different from one another. If the values are far different, another challenge will exist. There are many method that can justify between there three value. The simplest one is *linear interpolation.* Interpolation is a method of constructing new data points within the range of a discrete set of known data points [7]. The interpolation can be obtained unless there is common parameter among these three equations.

For optimal packet size based on BER, the equation:

$$k_{opt} = \frac{-h\ln(1-p) - \sqrt{-4hln(1-p) + h^2\ln(1-p^2)}}{2\ln(1-p)}$$

is depend on h and p. Since the data packet that is being sent has a fix header so h can remain fixed and only p remain change due to the link condition.

Secondly, the optimal packet size based on Energy Efficiency as express in equation:

$$N_{l_{opt}} = \frac{\sqrt{C_0^2 - \frac{4C0}{\ln(1-p)}} - C_0}{2} \quad \text{Where } C_0 = \alpha + k_2/k_1$$

Depends on $\boldsymbol{C_0}$. $\boldsymbol{C_0}$Itself depends on α and k_1, k_2. $k_2/k_1 \ll \boldsymbol{C_0}$.Therefore the value of k_2/k_1 can be illuminate. The equation can be written as

$$N_{l_{opt}} N_{l_{opt}} = \frac{\sqrt{\alpha^2 - \frac{4\alpha}{\ln(1-p)}} - \alpha}{2}$$

Now we can have optimal packet size based on Energy Efficiency is depending on header α and p. The same case to BER, header α will remain fixed. Therefore, only p remain change.

Finally, when come to optimal packet size based on *Throughput Efficiency* the equation:

$$N_{opt} = \frac{\mu}{2}\left[\sqrt{1 + \frac{4}{\mu\rho}} - 1\right]$$

Has one extra important parameter that take part in the expression ofμ is lR.

The interpolation between three equations might able to obtain by set *Throughput* a higher priority. This means that the computation of interpolation must be done for a known lR. For real practice lR is usually obtain by the default value of routing table. As a result, optimal packet size will be obtained. There exists another complication due to the behavior of complex expression of these three equations. Optimal range can be identified accordingly to the priority of each equation.

4.7 Proposed Algorithm

The algorithm is proposed in accordance to the latest progress of this research. The algorithm will be as below:

Connect the link

Test the link condition

Acquired: BER;

With acquired parameter, get the optimal packet size

- *Search for possible Optimal Packet Size based on BER*
- *Search for possible Optimal Packet Size based on Energy*
- *Search for possible Optimal Packet Size based on Throughput (Range or fixed value)*

Compute the interpolation

Assemble the data packet with Optimal Size;

Transmit the packet;

4.8 Data Look Up Table

The ability of the computation might be a barrier in some modern. To make it easy, data look up table will provide the optimal packet size that allow the microprocessor simply using a normal search to obtain the value. The data will be list down all the frequent lR values. The table below represents the look up table structure.

lR	P_{e1}	P_{e2}	P_{e3}

Table 7 Look Table

Sample data of LR

Sample data of packet size

CHAPTER 5

CONCLUSION

The three metrics namely throughput, efficiency and BER play an important role in the performance of Underwater Acoustic Network. The long propagation delay, high likelihood of the data looses, slow speed of transmission give a huge challenge for researchers. A better method to identify the optimal packet size should be proposed in order to achieve the optimal efficiency of the network. The existence of simulation tool, NS2 in particular, can help researcher to have a better understanding of the behavior of underwater network. There are some constrains that researcher face during their simulation which are the required knowledge in C++, TCL and most importantly understanding the infrastructure of the simulation. Without those understanding, researcher will found not be able to simulate the real underwater environment.

The algorithm that has been proposed needs a lot of improvement. The interpolation that has been proposed also required a better understanding in other mathematical methods that would hope give us a better result. The UWA and NS2 community in UTP is relatively small which required a lot of self research and self understanding that can contribute to prolong and slow down the speed of research.

REFERENCES

[1] Basagni S., Petrioli C., Petroccia R., Stojanovic M. 2009, *Choosing the Packet Size in Multi-hop Underwater Networks*, Northern University, Boston, MA

[2] Vuran M.C., Akyildiz I.F. 2008, *Cross-layer Packet Size Optimization for Wireless Terrestrial, Underwater, and Underground sensor Networks*, in Proc. IEEE INFOCOM '08, Phoenix, Arizona.

[3] DEI GRUPPO TELECOMUNICAZIONI, 2009. Retrieved March 8, 2010 from DGT website http://telecom.dei.unipd.it/ns/miracle/doxygen/

[4] Antonio B., *A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks,* IEEE, Porto, PORTUGAL

[5] Nitin J., Samir D,. Asis,. *A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Maultihop Wireless Networks.* U.S.A 2006

[6] Jinsung L,. Junhee L,. Yung Y,. *Implementing Utility-Optimal CSMA.* South Korea.

[7] Paul M,. Mikael S,. Amir Q,. *Comparison between Aloha and CSMA in Multiple hop ad Hoc networks.* Institut National De Recherche En Informatique Et En Automatique. Mars 2004. Germany

[8] Rajar, J.*Wireless Ad Hoc and Sensor Networks.* Springer, 2007, 217-239

[9] Deniel E., Muriel M., Milica S. *Underwater Acoustic Networks: Channel Models and Networks Coding based Lower Bound to Transmission Power for Multicast, 2008*

[10] Basagni S., Petroili C., Stiojanovic M., *Choosing the packet Size in Multi-hop Underwater Networks,* Northern University, Boston, Ma, Jan2009.

[11] Ian F., Dario P., Tommaso M. *Underwater acoustic sensor networks: research challenges,* Georgia Institute of Technology, Atlanta, February 2005.

[12] James P. *Acoustic Propagation Consideratoin for Underwater Acoustic Communications Network Development,* Woods Hole Oceanographic Institution, Woods Hole, MA

[13] Warren L. Sumit R., Nathan P. *Underwater Acoustic Communication Performance Modeling in Support ad Ad Hoc Network Design,* University of Washington,

[14] Albert F., Michele Z. *Modeling the Underwater Acoustic Channel in ns2,* University of Illinois, Undersity of Padova, Italy.

[15] Wei L., Haibin Y. *Experiment Research on Underwater Acoustic Sensor Network.* Shenyang Institute of Automation, Chinese Acedemy of Sciences, China. IEEE, 2007

[16] Borja P., Milica S., *A MAC Protocol for Ad-Hoc Underwater Acoustic Sensor Networks.* Massachusetts Institute of Technology, Cambridge, MA. IEEE, 2006

[17] Qilian L., Xiuzhen C., *Underwater Acoustic Sensor Networks: Target Size Detection and Performance Analysis.* IEEE, 2008

[18] Jing H., Jiangou H., Maohua R., *Design and Simulation of an Underwater Acoustic Network.* College of marine, Northwestern polytechnical University, China. IEEE

[19] Ethem S., Milica S., John P,.*Underwater Acoustic Networks.*IEEE, 2000

[20] John G., Geoffrey X. *Analyzing the Performance of Multi-hop Underwater Acoustic Sensor Networks*. IEEE, 2007

[21] Paul B., Homer B., *Acoustic Communication Channel Modeling for the Baltic Sea.* Space and Naval Warfare Systems Center, San Diego, California. IEEE

[22] Yue W. *A Tutorial of 802.11 Implementation in ns-2. MobiTec Lab, CUHK*

[23] Warren L, J. Sumit R,.*Underwater Acoustic Communication Performance Modeling in Support of Ad Hoc Network Design.*MTS, 2007

[24] Joe R., Dale G. *Underwater Acoustic Communications and Networks for the US Navy's Seaweb Program.* IEEE, 2008

[25] Nicola B., Marco M. *Spectrum-aware Channel and PHY layer modeling for NS3.* ACM, 2009

[26] Nitthita C., Wee S,. Kee C. *Aloha-based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks.* IEEE, 2007

[27] Peng X., Zhong Z. *Aqua-Sim: An NS-2 Based Simulator for Underwater Sensor Networks*. MTS: 2009

[28] Yu Y., Peng B. *The Protocol of Physical Layer for Underwater Acoustic Network.*IEEE, 2010

[29] Jing H., Jianggou H. *Design and Simulation of an Underwater Acoustic Network.* College of Marine, Northwestern Polytechnical University, China

[30] Issariyakul H. *Introduction to Network Simulator NS2.* Springer, 2009.

[31] *Aqua-Sim.* http://ubinet.engr.uconn.edu/mediawiki/index.php/Aqua-Sim

[32] Sankarasubramaniam Y., Akyildiz I. F., McLaughlin S. W. 2003, "Energy efficiency based packet size optimization in wireless sensor networks," in *Proc. IEEE Internal Workshop on Sensor Network Protocols and Applications*, pp. 1-8

[33] E. Modiano, "Data Link Protocols for LDR MILSTAR Communications", Lincoln Laboratory, Communications Division Internal Memorandum, October 1994.

APPENDEX A

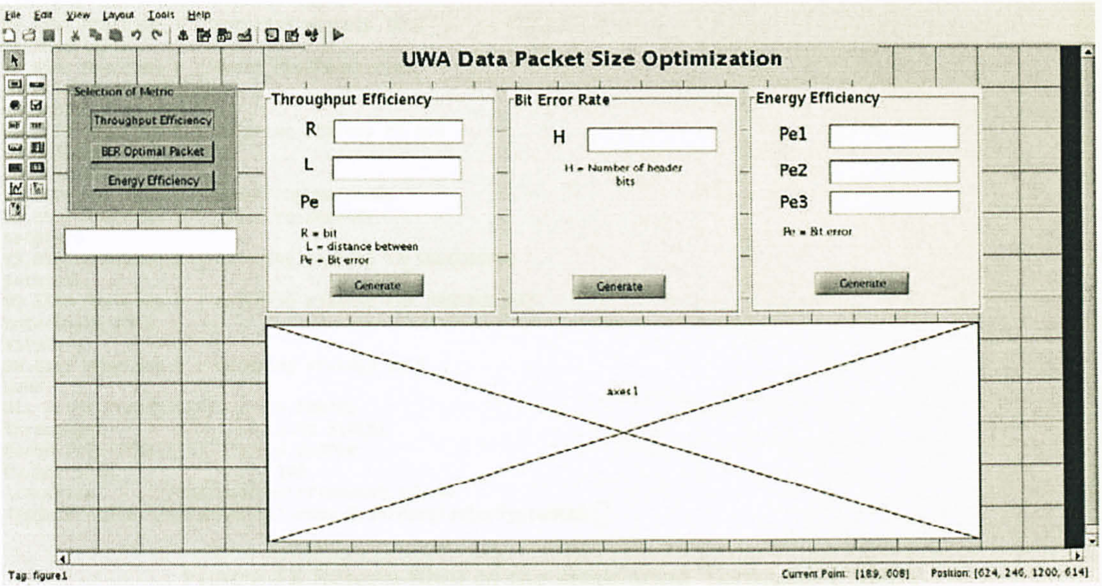

Figure 16 Design Graph Interface

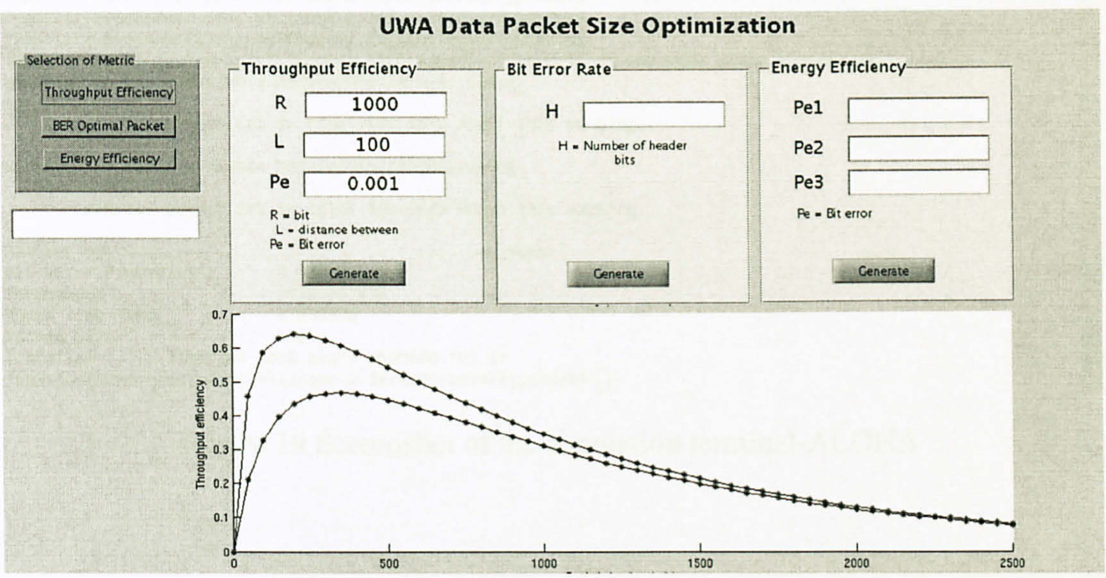

Figure 17 Graph Plotting

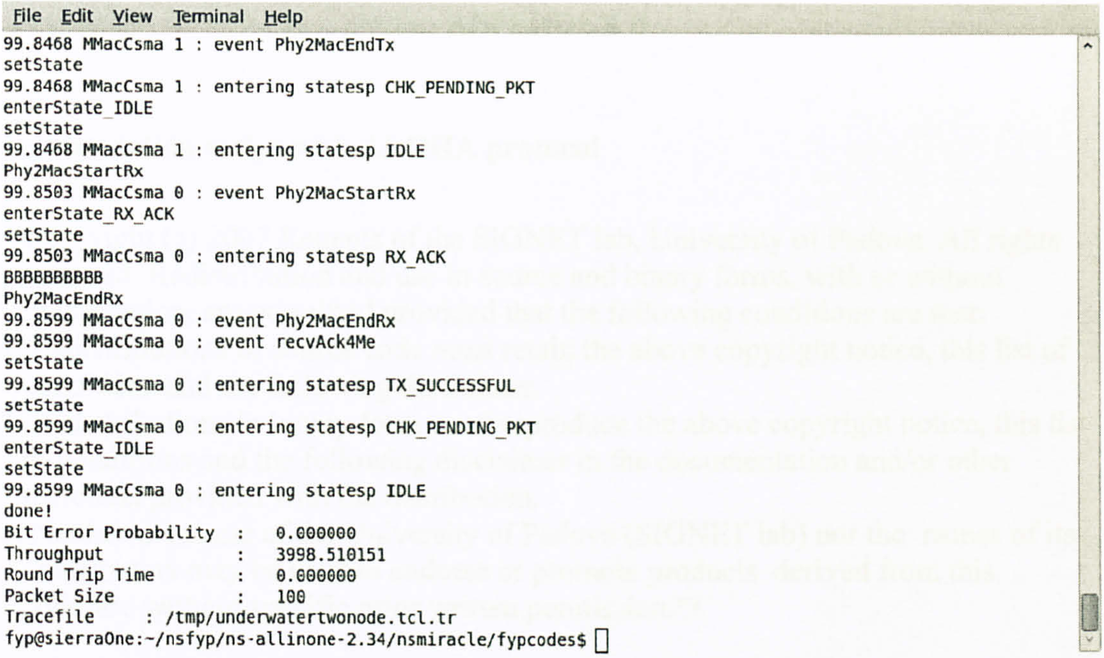

Figure 18 Screen Shot of the simulation Terminal-CSMA

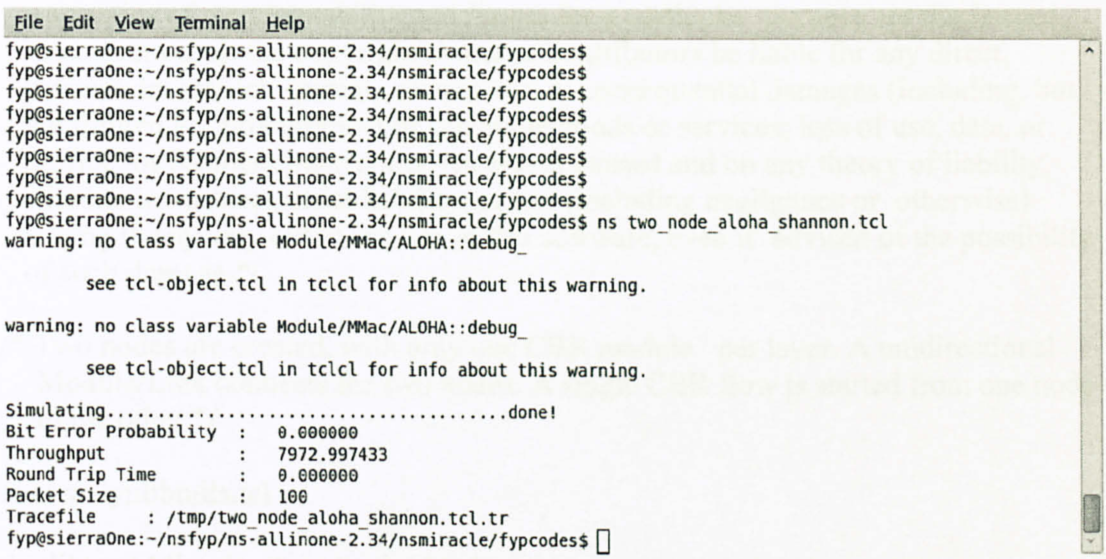

Figure 19 Screenshot of the simulation terminal-ALOHA

APPENDEX B

Tcl simulation script with ALOHA protocol


```
/* Two nodes are created, with only one CBR module per layer. A unidirectional
   Module/Link connects the two nodes. A single CBR flow is started from one node
   to the other.*/

source dynlibutils.tcl

dynlibload Miracle ../nsmiracle/.libs/
dynlibload miraclelink ../link/.libs/
dynlibload miraclecbr ../cbr/.libs/
dynlibload Trace ../trace/.libs/

# Module libraries

load libMiracle.so
load libmiraclecbr.so
load libMiracleWirelessCh.so
load libmphy.so
load libMiracleBasicMovement.so
```

```
load libmphytracer.so
load libcbrtracer.so
load libsinrtracer.so
load libverboseclcmntracer.so
load libuwm.so

proc finish {} {
   global ns tf rm opt cbr packetSize
   puts "done!"
   $ns flush-trace
   close $tf
   #set per [$rm getper]
   #set thr [$rm getthr]
   #set rtt [$rm getrtt]
   set per [$cbr(2) getper]
   set thr [$cbr(2) getthr]
   set rtt [$cbr(2) getrtt]
   puts "Bit Error Probability  :  $per"
   puts "Throughput             :  $thr"
   puts "Round Trip Time        :  $rtt"
   puts "Packet Size    :  $opt(packetSize)"
   puts "Tracefile     : $opt(tracefile)"
}

##########################################
# global allocations & misc options
##########################################

set opt(starttime)     0.0
set opt(stoptime)    100
set opt(txduration) [expr $opt(stoptime) - $opt(starttime)]
set opt(xmax)        5000.0
set opt(packetSize)  100
set channel [new Module/UnderwaterChannel]
set propagation [new MPropagation/Underwater]
set smask [new MSpectralMask/Rect]
$smask setFreq 8.2e3
$smask setBandwidth 6e3

MInterference/MIV set maxinterval_ 500
Module/MPhy/UWShannon set debug_ 0
Module/MPhy/UWShannon set TxPower_  5.2481e13

set ns [new Simulator]
$ns use-Miracle
```

```
set errorRNG [new RNG]
# seed random number generator according to replication number
if {$argc >= 1 } {
   set run [lindex $argv 0]
   puts "replication number: $run"
   for {set j 1} {$j < $run} {incr j} {
      $errorRNG next-substream
   }
}

#Module/CBR set debug_ 1
Module/CBR set packetSize_ $opt(packetSize)
Module/CBR set period_    0.1

set opt(tracefile)   "/tmp/${argv0}.tr"
set tf [open $opt(tracefile) w]
$ns trace-all $tf
set devnull [open  "/dev/null"  w]

##################################
# Procedure for creating nodes
##################################

proc createNode {id } {

   global channel propagation smask ns cbr position  node phy mac

   set node($id) [$ns create-M_Node]
   set cbr($id) [new Module/CBR]
   set mac($id) [new Module/MMac/ALOHA]
   set phy($id) [new Module/MPhy/UWShannon]

   $node($id) addModule 3 $cbr($id) 0 "CBR($id)"
   $node($id) addModule 2 $mac($id) 0 "MAC($id)"
   $node($id) addModule 1 $phy($id) 0 "PHY($id)"

   $node($id) setConnection $cbr($id) $mac($id) 1
   $node($id) setConnection $mac($id) $phy($id) 1
   $node($id) addToChannel $channel $phy($id)   1

   set position($id) [new "Position/BM"]
   $node($id) addPosition $position($id)
   $position($id) setX_ 0.0
   $position($id) setY_ 0.0
   $position($id) setZ_ -10.0

   $phy($id) setSpectralMask $smask
```

```
   $phy($id) setPropagation $propagation

   set interf($id) [new MInterference/MIV]
   $phy($id) setInterference $interf($id)

}

#################################
# create nodes
#################################

createNode 1
$position(1) setX_ 100.0
$ns at $opt(starttime)   "$cbr(1) start"
$ns at $opt(stoptime)   "$cbr(1) stop"

createNode 2
$position(2) setX_ 200.0
$position(2) set debug_ 543

$phy(1) setDestPosition $position(2)
$phy(2) setDestPosition $position(1)

set link [new Module/Link]
$link delay      0.01
$link bandwidth  1000000
$link qsize      5
$link connect $node(1) $cbr(1) 1 $node(2) $cbr(2) 1
#$link print-params

set rv [new RandomVariable/Uniform]
$rv set min_ 0
$rv set max_ 100
$rv use-rng $errorRNG
set em [new ErrorModel]
$em unit bit
$em set rate_ 5
$em ranvar $rv

$link addErrorModel $em
$cbr(1) set debug_ 0

set stop 20
$ns at [expr $stop + 1] "finish; $ns halt"

for {set t 0} {$t < $stop} {set t [expr $t + ($stop / 40.0)  ]} {
   $ns at $t "puts -nonewline . ; flush stdout"
```

```
}

puts -nonewline "Simulating"
$ns run
```

Tcl simulation script with CSMA protocol


```
/* Two nodes are created, with only one CBR module per layer. A unidirectional
   Module/Link connects the two nodes. A single CBR flow is started from one node
   to the other.*/

source dynlibutils.tcl

dynlibload Miracle ../nsmiracle/.libs/
dynlibload miraclelink ../link/.libs/
dynlibload miraclecbr ../cbr/.libs/
dynlibload Trace ../trace/.libs/

# Module libraries

load libMiracle.so
load libmiraclecbr.so
load libMiracleWirelessCh.so
load libmphy.so
load libMiracleBasicMovement.so
load libmphytracer.so
load libcbrtracer.so
load libsinrtracer.so
```

```
load libverboseclcmntracer.so
load libuwm.so

proc finish {} {
  global ns tf rm opt cbr packetSize
  puts "done!"
  $ns flush-trace
  close $tf
  set per [$cbr(2) getper]
  set thr [$cbr(2) getthr]
  set rtt [$cbr(2) getrtt]
  puts "Bit Error Probability  :  $per"
  puts "Throughput             :  $thr"
  puts "Round Trip Time        :  $rtt"
  puts "Packet Size            :  $opt(packetSize)"
  puts "Tracefile              : $opt(tracefile)"
}


##########################################
# global allocations & misc options
##########################################

set opt(starttime)  0.0
set opt(stoptime)   100
set opt(txduration) [expr $opt(stoptime) - $opt(starttime)]
set opt(xmax)       5000.0
set opt(packetSize) 40
set opt(HeaderSize) 40
set opt(BaseBackoffTime) 0.01
set opt(AckTimeout) 5
set opt(debug_states) 1
set opt(debug) 1

set channel [new Module/UnderwaterChannel]
set propagation [new MPropagation/Underwater]
set smask [new MSpectralMask/Rect]
$smask setFreq 8.2e3
$smask setBandwidth 6e3

MInterference/MIV set maxinterval_ 500
Module/MPhy/UWShannon set debug_ 0
Module/MPhy/UWShannon set TxPower_  5.2481e6
Module/MMac/CSMA set HeaderSize_ $opt(HeaderSize)
Module/MMac/CSMA set debug_ $opt(debug)
```

```
Module/MMac/CSMA set BaseBackoffTime_ $opt(BaseBackoffTime)
Module/MMac/CSMA set AckTimeout_ $opt(AckTimeout)
Module/MMac/CSMA set debug_states_ $opt(debug_states)

set ns [new Simulator]
$ns use-Miracle

set errorRNG [new RNG]
if {$argc >= 1 } {
   set run [lindex $argv 0]
   puts "replication number: $run"
   for {set j 1} {$j < $run} {incr j} {
      $errorRNG next-substream
   }
}

Module/CBR set packetSize_ $opt(packetSize)
Module/CBR set period_   0.2

set opt(tracefile)  "/tmp/${argv0}.tr"
set tf [open $opt(tracefile) w]
$ns trace-all $tf
set devnull [open  "/dev/null"  w]

##################################
# Procedure for creating nodes
##################################

proc createNode {id } {

   global channel propagation smask ns cbr position  node phy mac

   set node($id) [$ns create-M_Node]
   set cbr($id) [new Module/CBR]
   set mac($id) [new Module/MMac/CSMA]
   set phy($id) [new Module/MPhy/UWShannon]

   $node($id) addModule 3 $cbr($id) 0 "CBR($id)"
   $node($id) addModule 2 $mac($id) 0 "MAC($id)"
   $node($id) addModule 1 $phy($id) 0 "PHY($id)"

   $node($id) setConnection $cbr($id) $mac($id) 1
   $node($id) setConnection $mac($id) $phy($id) 1
   $node($id) addToChannel $channel $phy($id)   1

   set position($id) [new "Position/BM"]
   $node($id) addPosition $position($id)
```

```
  $position($id) setX_ 0.0
  $position($id) setY_ 0.0
  $position($id) setZ_ -10.0

  $phy($id) setSpectralMask $smask
  $phy($id) setPropagation $propagation

  set interf($id) [new MInterference/MIV]
  $phy($id) setInterference $interf($id)

}

##################################
# create nodes
##################################

createNode 1
$position(1) setX_ 10.0
$ns at $opt(starttime)   "$cbr(1) start"
$ns at $opt(stoptime)   "$cbr(1) stop"

createNode 2
$position(2) setX_ 30.0

$phy(1) setDestPosition $position(2)
$phy(2) setDestPosition $position(1)

set link [new Module/Link]
$link delay      0.01
$link bandwidth  1000000
$link qsize      5
$link connect $node(1) $cbr(1) 1 $node(2) $cbr(2) 1

#$link print-params

set rv [new RandomVariable/Uniform]
$rv set min_ 0
$rv set max_ 100
$rv use-rng $errorRNG
set em [new ErrorModel]
$em unit bit
$em set rate_ 5
$em ranvar $rv
$link addErrorModel $em
$cbr(1) set debug_ 0
set stop 100
```

```
$ns at [expr $stop + 1] "finish; $ns halt"

for {set t 0} {$t < $stop} {set t [expr $t + ($stop / 40.0)  ]} {
   $ns at $t "puts -nonewline . ; flush stdout"
}

puts -nonewline "Simulating"
$ns run
```