CHAPTER 4

SIMULATIONS

4.1 Introduction

This chapter focuses on finding relationship between optimal data packet size and the performance metrics/parameters for throughput efficiency and energy efficiency (with implicit energy per useful bit metric) under different bit error rate (BER) and varying distances between a source-sink pair. The findings were obtained through a simulation of underwater channel model implemented in ns2-MIRACLE (Multi-InteRfAce Cross-Layer Extension library) package [50] running on Ubuntu platform. The ns2-MIRACLE package can be downloaded from this link *https://telecom.dei.unipd.it: /tlcrepos/nsmiracle-dev/trunk.*

Due to its simplicity in implementation and ease of control the Aloha MAC protocol was used throughout the simulation works. Since it was to be a one-hop data transmission so the stop-and-wait ARQ handshaking protocol at PHY layer was implemented between any source-sink pair of nodes for transmission of data packets. The simulations and the related outcomes described in this chapter are in parts to verify the viability of the equations analyzed in section 3.2 and to support the data packet size optimization algorithm proposed in section 3.3 for UWA communications.

The network simulator used in this simulation is ns2 version 2.34 – a discrete event simulator aimed at networking research. ns2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. The MIRACLE package in fact is an add-on to ns2 in that ns-2.34 does not provide any framework for underwater acoustic communication networking.

MIRACLE is a set of libraries written to add on to the functionalities of ns2. It provides an embedded engine for handling cross-layer messages and enables the co-existence of multiple modules within each layer of the protocol stack. Multiple IP, link layers, MACs or physical layers for instance can be specified and used within the same node. The MIRACLE framework facilitates wired networks as well as a mixture of wired and wireless scenarios in ns2. Moreover, due to its modularity, the code comes to be portable, re-usable and extensible [54].

4.2 The Simulation

4.2.1 General Scenario

Figure 4.1 shows a general scenario of the underwater environment set up for the simulation. The environment is made simple to avoid complications in the essential investigations. To avoid the complications in acoustic wave reflection near the water surface and the reflection near the bottom surface, a cluster of 100 nodes is placed in the middle of a body of water with dimension of 2km x 2km x 200m. Should the readers be interested in more advanced investigations, the effect of acoustic wave reflection and refraction on optimal data packet size shall be included in the simulation.

The depth of 200m was chosen to simulate the shallow water (based on the commonly accepted shallow water definition) environment. In line with the proposed model for one hop data transmission, somewhere at the middle of the cluster is a sink to collect data packets from other source nodes. The maximum distance between the sink and a node is set to 1km. Thus a dimension of 2km by 2km is deemed sufficient. It follows that the maximum transmission range of the nodes is to 1km. The distance between the sink and the various source nodes in the cluster will be varied from a minimum of 10m to a maximum of 1km in the course of the simulations.

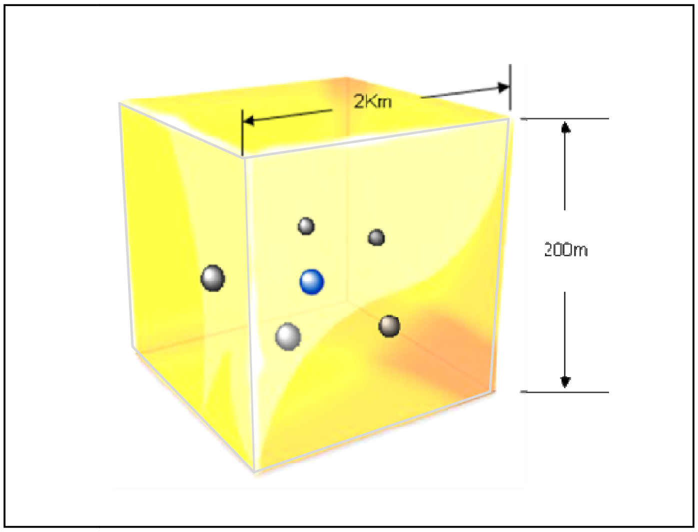


Figure 4.1: General scenario showing a cluster of nodes set up for the simulation

4.2.2 Simulation Setup

In the simulation two nodes are created (one as a transmitter/source and the other as a receiver/sink) at any one time for a one hop data packet transfer with one constant bit rate (CBR) module per layer. A bidirectional module/link connects the two nodes. A single CBR packet flow is then started from one node to the other. Figure 4.2 shows the layers involved in both nodes for a packet flow under the ns2 MIRACLE framework.

The first layer is the Constant Bit Rate (CBR) layer. Packet traffic is generated here and the schedule of the generation and message passing to the next layer is done at this layer too. For example, the packet is generated at a constant bit rate of 150byte/packet and sent to the next corresponding layer in a calculated flow.

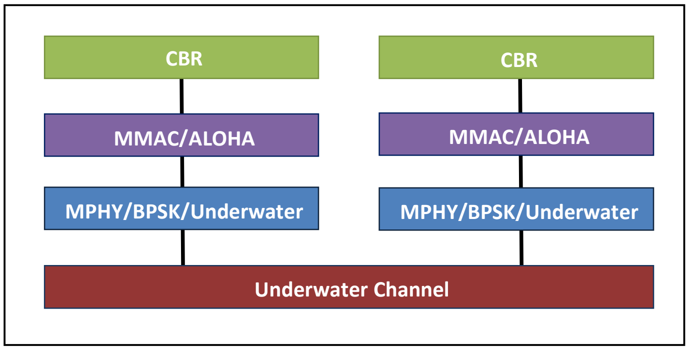


Figure 4.2: Ns2 MIRACLE layered framework

The second layer is the MMAC layer. The original second layer in ns2 is MAC layer but in ns2 with the Miracle package, it is called Miracle MAC (MMAC). The difference between the original MAC and the MMAC lies in the transmission and in the receiving functions. In MMAC it will call different sub modules related to the configured underwater channel in the process of the simulation. This layer concerns with how the packet is transmitted and specified. In this work, the MMAC is specified to be MMAC/ALOHA.

Going a level down is the physical layer called Miracle Physical (MPHY) layer. MPHY/BPSK/Underwater layer is a special class that helps simulating the real underwater environment. This layer is specified to use Binary Phase Shift Keying (BPSK) modulation in the underwater modules. In customizing the modules in both layers (MMAC and MPHY) for being able to be invoked at the time of simulation, the script parameter definition of Tool Command Language (abbreviated as TcL and pronounced "T-C-L" or "tickle") is needed such as this code snippet: `set phy($id) [new Module/MPhy/UWShannon]`. Sample of more complete TcL codes can be found in the appendix.

Tool Command Language is a powerful interpreted programming language developed by John Ousterhout. One of the main strengths of Tcl is that it can be easily extended through the addition of customized TcL libraries. It is commonly used for prototyping applications as well as for developing Common Gateway Interface (CGI) scripts, though it is not as popular as Perl scripting language in terms of CGI development.

The transmitter CBR module, acting as an agent, generates data packet of the required size. The Aloha MAC protocol, for is simplicity nature, is deployed in the MMAC layer for media access in this simulation with the aim to investigate the fundamental relationships between packet size and the performance metrics as described in section 4.1.

The MPHY layer uses BPSK modulation to send the data packet over the underwater channel to the receiver/sink. The underwater channel is configured with typical Shannon channel characteristics which include power level at the source, the bandwidth of the channel, the noise level, the link delay, etc.

The simplified sequence for packet transmission (Tx) and reception (Rx) events in the MIACLE PHY and MAC framework is shown in Figure 4.3. For readers' information, the details of the classes found in MIRACLE package are readily available in [51].

For simulation works, packet sizes of few bytes to thousands of bytes can be generally generated. However there are situations at which packet sizes of more than thousand bytes may be required. The distance between the source node and the sink node varies from 10m to 1000m with various increment steps depending on the position of the source and the sink nodes.

The transmitting node is set to a carrier frequency of 8.2 kHz with the signal bandwidth of 6 kHz. The link is having a capacity of 100kbps with a DropTail queue. It is emphasized here that all these values are chosen according to the common underwater acoustic transmission practices by considering that the outcomes of the simulation shall not be too far off from the practical situations.

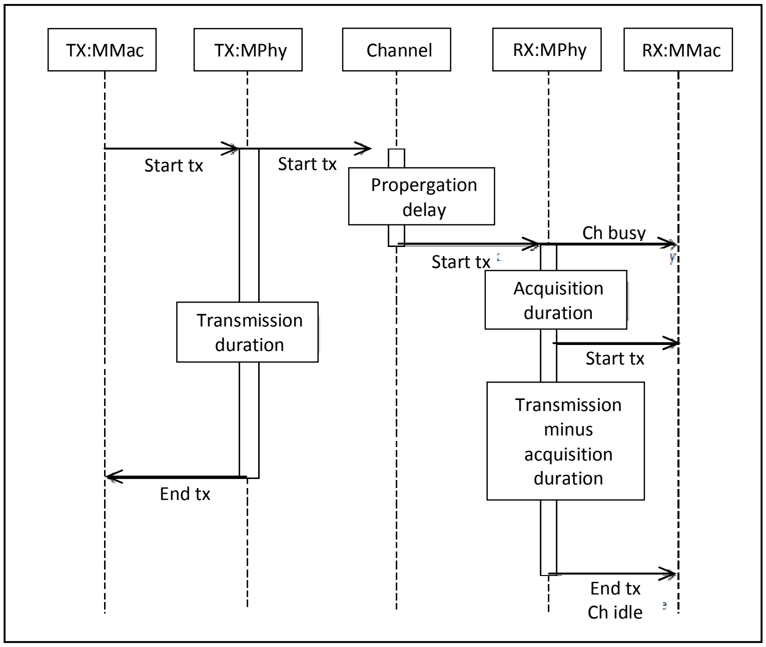


Figure 4.3: Simplified sequence diagram for packet transmission (Tx) and reception (Rx) events in the Miracle PHY and MAC framework [51]

In general, the basic throughput definition from [18] has been adopted in the simulation, or otherwise it would be explicitly described in various specific situations. Other specific setups or configurations and parameters needed for the simulation of each of the metrics including throughput efficiency, energy efficiency and energy per useful bit will be highlighted in each of the following subsections.

4.3 Results and Discussions

Three sets of simulations with Aloha MAC protocol were conducted via the ns2 simulator. The underwater channel model was, of course, based on the ns2-MIRACLE classes or libraries in the package. All the simulations results discuss in this section are aimed to directly or indirectly verify the results of the analyses presented in sections 3.2 to see if they support the optimization algorithm proposed in section 3.3. In consistent to the descriptions in the previous chapter, the outcomes of the simulations would be consolidated to construct four look up graphs (or databases) needed to support the implementation of the proposed algorithm.

The first set of graphs relates packet sizes to BERs for different header length. The next set relates packet sizes to throughput efficiency for various BERs. The third set relates packet sizes to the packet header length and transmitter/receiver constants, under various BERs. The final set is packet sizes to energy efficiency with implicit energy per useful bit element for various BERs.

It should be mentioned here that although four sets of graphs were consolidated, the proposed algorithm shall not use the third set of graphs in computing the optimal data packet size. The third set was consolidated merely to verify the feasibility and the applicability of the proposed algorithm. Therefore only three sets of graphs (or databases) are actually needed for the proposed optimization process.

4.3.1 Data Packet Size and Bit Error Rate

This simulation was based on equation (3.11) described in subsection 3.2.2 on page 48. For convenience and easy reference, the equation is restated her.

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

This equation shows that the optimal packet size k_{opt} is the function of BER (p) and packet header length (h). Note the natural logarithmic characteristic. The simulation in this subsection is to verify and investigate how the UW channel actually copes with this logarithmic characteristic. Some of the crucial parameters used in this simulation include:

Channel type	: Underwater Shanon
Packet Size	: 10 - 2000 bits
Header length	: 40 bits
Distance	: 100m - 1000m
Channel Frequency	: 8.2 kHz
Signal Bandwidth	: 6 kHz
MAC Protocol	: Aloha
Constant bit rate interval	: 0.01s, 0.03s, 0,05s

Part of the simulated output data are shown in Table 4.1 with the range of BERs from 0.0001 to 0.01 and from 0.0051 to 0.006. The k_{opt} values and their related BERs listed in Table 4.1 are plotted as shown in Figure 4.4. Please take note that the k_{opt} entries with their corresponding BERs listed in the Table 4.1 are reflected in the left hand portion of Figure 4.4.

An extended version of Figure 4.4 is presented in Figure 4.5 using a more complete set of data in Appendix A. The graph in Figure 4.4 shows the relationship of k_{opt} and BERs for a header length of 40 bits and it manifests the logarithmic characteristic of the k_{opt} with respect to BERs.

Table 4.1: Partial data obtained from k_{opt} simulation for header of 40 bits

BER	k_{opt}	**BER**	k_{opt}	**BER**	k_{opt}	**BER**	k_{opt}	**BER**	k_{opt}
0.0001	612.1234	0.0011	169.5892	0.0021	116.4861	0.0031	91.73385	0.0041	76.63188
0.0002	426.7439	0.0012	161.4216	0.0022	113.2767	0.0032	89.91482	0.0042	75.42259
0.0003	344.573	0.0013	154.2118	0.0023	110.2775	0.0033	88.17818	0.0043	74.25512
0.0004	295.5632	0.0014	147.7855	0.0024	107.4661	0.0034	86.51785	0.0044	73.12706
0.0005	262.0996	0.0015	142.0101	0.0025	104.8238	0.0035	84.92834	0.0045	72.03624
0.0006	237.3847	0.0016	136.7821	0.0026	102.3342	0.0036	83.40472	0.0046	70.98061
0.0007	218.1662	0.0017	132.02	0.0027	99.98285	0.0037	81.94251	0.0047	69.95831
0.0008	202.6664	0.0018	127.6582	0.0028	97.75748	0.0038	80.53765	0.0048	68.96759
0.0009	189.8209	0.0019	123.6433	0.0029	95.6471	0.0039	79.18646	0.0049	68.00684
0.001	178.9482	0.002	119.9314	0.003	93.64208	0.004	77.88556	0.005	67.07455

BER	k_{opt}	**BER**	k_{opt}	**BER**	k_{opt}	**BER**	k_{opt}	**BER**	k_{opt}
0.0051	66.16933	0.0061	58.3568	0.0071	52.22609	0.0081	47.24187	0.0091	43.08086
0.0052	65.28988	0.0062	57.67923	0.0072	51.68263	0.0082	46.79286	0.0092	42.70132
0.0053	64.43498	0.0063	57.0175	0.0073	51.15009	0.0083	46.35174	0.0093	42.32772
0.0054	63.60348	0.0064	56.371	0.0074	50.6281	0.0084	45.91829	0.0094	41.95988
0.0055	62.79433	0.0065	55.73912	0.0075	50.1163	0.0085	45.49227	0.0095	41.59768
0.0056	62.00653	0.0066	55.12133	0.0076	49.61436	0.0086	45.07347	0.0096	41.24094
0.0057	61.23914	0.0067	54.51709	0.0077	49.12196	0.0087	44.66169	0.0097	40.88955
0.0058	60.49127	0.0068	53.92591	0.0078	48.6388	0.0088	44.25673	0.0098	40.54335
0.0059	59.7621	0.0069	53.34731	0.0079	48.16458	0.0089	43.85839	0.0099	40.20223
0.006	59.05086	0.007	52.78084	0.008	47.69903	0.009	43.46649	0.01	39.86605

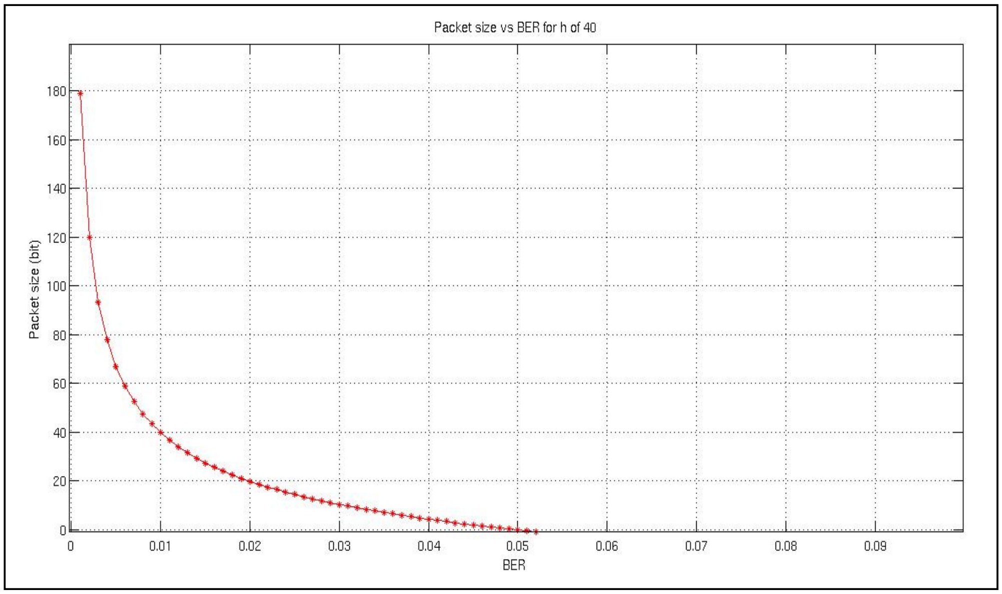


Figure 4.4: Packet size against BER in linear scale

The graph in Figure 4.4 is plotted in linear scale for small intervals of BERs to see the non-linear (logarithmic) relationship between the optimum packet size and the BERs. A sharp exponential drop in packet size is observed with BER approaching 0.01. Interestingly, at the BER of 0.01 no payload bit can be sent in that the whole packet length is now occupied by the entire header bits (40 bits in this case). From the plot in Figure 4.4 it is observed that the channel/link quality would affect the optimum packet size *exponentially*.

To see an extended picture of packet size against BERs the data collected in Appendix A were plotted out in a log scale as shown in Figure 4.5. The graph in Figure 4.5, again, shows an exponential drop in packet size when the BER (the link quality indicator) deteriorates. With a good quality link, say for BER of 0.001 and less, packet size can vary from around 200 bits to near 2000 bits. This is well above the header length of 40 bits which certainly shall result in high throughput.

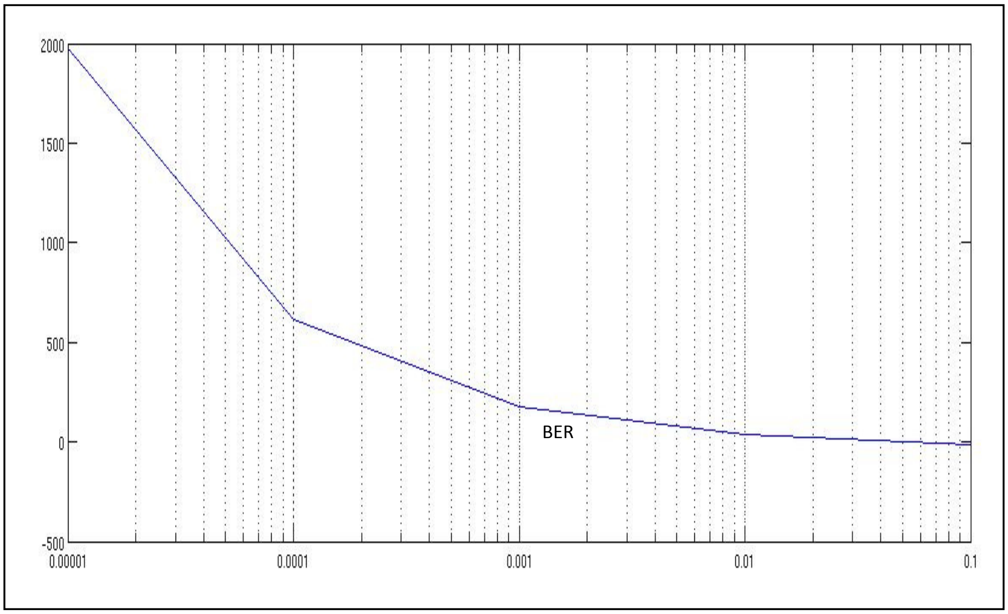


Figure 4.5: Packet size against BER in log scale

To see an even more comprehensive relationship of packet size against BERs, Figure 4.6 is plotted with a set of graphs relating packet size to different BERs under different header length. The full set of data is shown in Appendix C. This is the sample set of graphs that the proposed optimization algorithm will consult as part of the process in computing the optimal packet size for different BER metrics. Note that a header length of 160 bits is the standard length used in the Request-To-Send (RTS) data packet for the conventional stop-and-wait ARQ protocol.

It is understood that under this stop-and-wait handshaking protocol the source node will transmit an RTS packet to the sink node to check if a link can be established between them before any packet is transmitted. In the proposed algorithm this RTS packet will double its function as a test data packet for the source node to test/compute the quality of the link thus obtaining the link BER value.

Please take note that the top most graph/line in Figure 4.6 is to be used as a reference graph to explain the principle used in the proposed algorithm and the rest of the plots will be used for purpose of the comparative study in the next chapter.

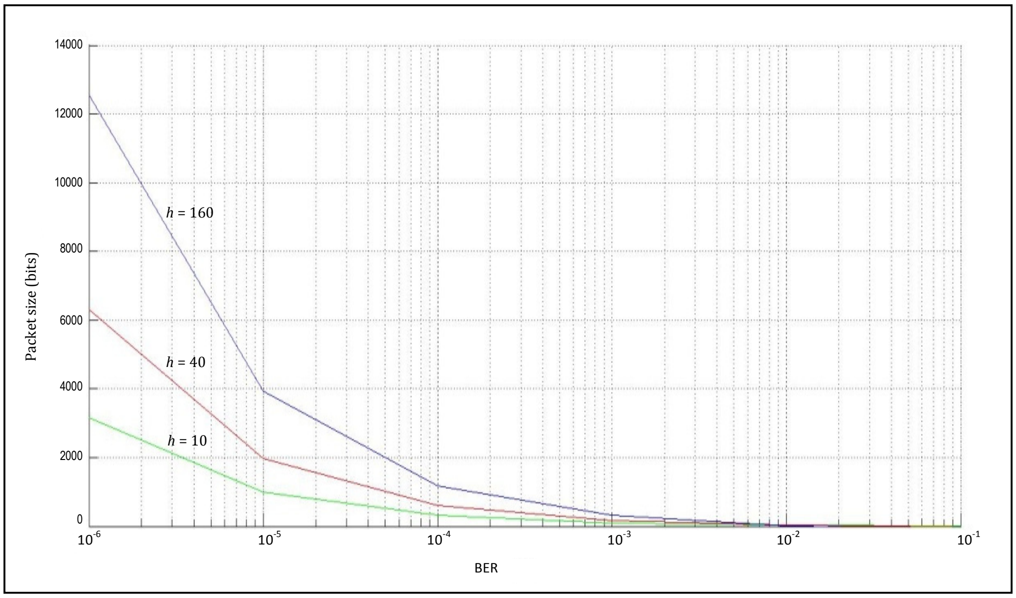


Figure 4.6: Packet size against BERs with different header length (*h*)

For a quicker, but probably a coarser referencing or indexing into Figure 4.6 to find a packet size under a certain link BER, a simplified data set can be obtained from Figure 4.6 for this purpose. For example the simplified data set for a header length of 40 bits is obtained as shown in Table 4.2. This simplified data set stores BERs in an incremental step of a decade. These large increment steps in BERs can make the computation of optimum packet size practically much faster. This is to say that it may not be necessary to use a BER down to a detail of, say 3×10^{-4}, when 1×10^{-4} will do. This is acceptable because if a 1×10^{-4} BER is considered not good enough then let alone the BER of 3×10^{-4}.

It is essential to be informed that, for practical implementation, the packet size composed for actual transmission may use the truncated value as shown in the last column of Table 4.2. However a round-off value may be considered if a user is not in favor of truncation. The author would recommend truncation since it is easier to implement and will not make much difference in comparison to the value obtained from rounding process.

Table 4.2: A simplified data set

BER	k_{opt} (bits)	Truncated value of k_{opt}
10^{-2}	39.86605	39
10^{-3}	178.9482	178
10^{-4}	612.1234	612
10^{-5}	1979.8950	1979
10^{-6}	6304.5221	6304

4.3.2 Data Packet Size and Throughput Efficiency

The optimal packet size for different throughput efficiencies denoted by equation (3.9) in subsection 3.2.1 page 47 was the main reference used in this simulation. Equation (3.9) is restated here for easy reference,

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

where,

$$\mu = N_{oh} + \frac{T_W R}{g} = N_{oh} + \frac{2}{gc} lR \tag{4.1}$$

Take note that μ is related to lR (range-rate) product which has a dimension of m-bps. lR denotes the product of distance l (in meter) between a source-sink pair and the data transmission rate R (in bps) between the source node and the sink node. Note that lR is a factor that affects the throughput efficiency.

It is explicit to see in the equation (3.9) that N_{opt} is a function of range-rate product (lR) and the BER (p) of the communication link. Some of the crucial simulation parameters used in this simulation include:

Channel type : Underwater Shanon
Channel frequency : 8.2 kHz
Signal bandwidth : 6 kHz
Queue size : 5
Link delay : 0.01s
BER (ρ) : 10^{-3} , 10^{-4}
Distance (l) : 500 m to 5 km
Rate (R) : 100 bps to 1000 bps
Header (N_{oh}) : 100 bits
No. of group (g) : 1
MAC : Aloha

The simulation of this N_{opt} results in a set of graphs shown in Figure 4.7. These graphs depict a relationship between optimum packet size as a function of range-rate for different p (BER). The graphs plotted in Figure 4.7 are actually representing only part of the simulation output data consolidated in Appendix K. Only the relationship of packet size against the lR products qualified by the p of 10^{-3} and 10^{-4} are shown here for illustration purpose. A more detailed set of graphs will be discussed in the next chapter for results analysis.

It is straightforward to observe in Figure 4.7 that low quality link does not permit large packet size. It was also observed by the author that by keeping the distance l between the source-sink pair constant, example in static nodes deployment, and under a certain value of BER, the packet size seems to be fairly linearly increasing with an increasing R. However the packet size increases at a faster rate if the link p is low.

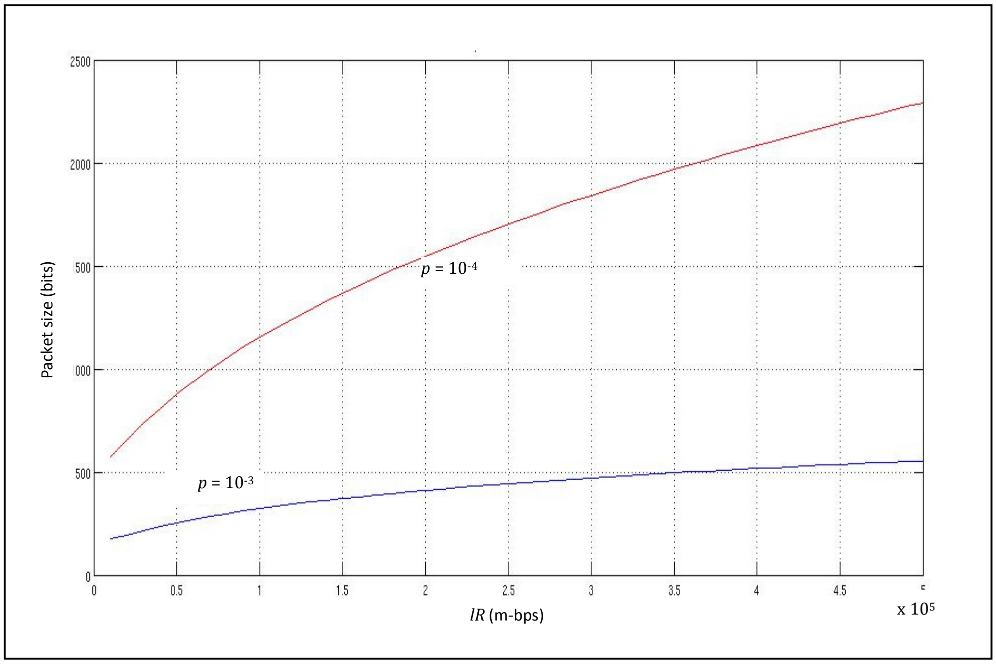


Figure 4.7: Packet size against range-rate (*lR*) with different BER (*p*)

The optimal packet size obtained from Figure 4.7 can be use to plot the throughput efficiency graphs as shown in Figure 4.8. That is, by substituting the N_{opt} variable in equation (3.10) on page 47 with values from Figure 4.7 a set of throughput efficiency graphs is thus plotted. For the purpose of referencing, equation (3.10) on page 47 is given here as,

As can be seen from Figure 4.8, and as expected, link with high BER will have lower peak throughput efficiency than the one with low BER. It is mainly due to the smaller optimal packet size allowed in low quality link. For instance, with constant *lR* product of 5e+2 but different p of 10^{-4} (curve 1) and 10^{-3} (curve 2) in Figure 4.8, the peak throughput efficiency for p of 10^{-4} is approximately 1.5 times better than p of 10^{-3}.

Moreover, at this peak efficiency the optimal packet size for p of 10^{-4} is doubled that for p of 10^{-3}. The reason for this phenomenon is that the high BER is certainly to produce more packets lost in the course of transmission and as a result, the performance of peak throughput efficiency is pulled down.

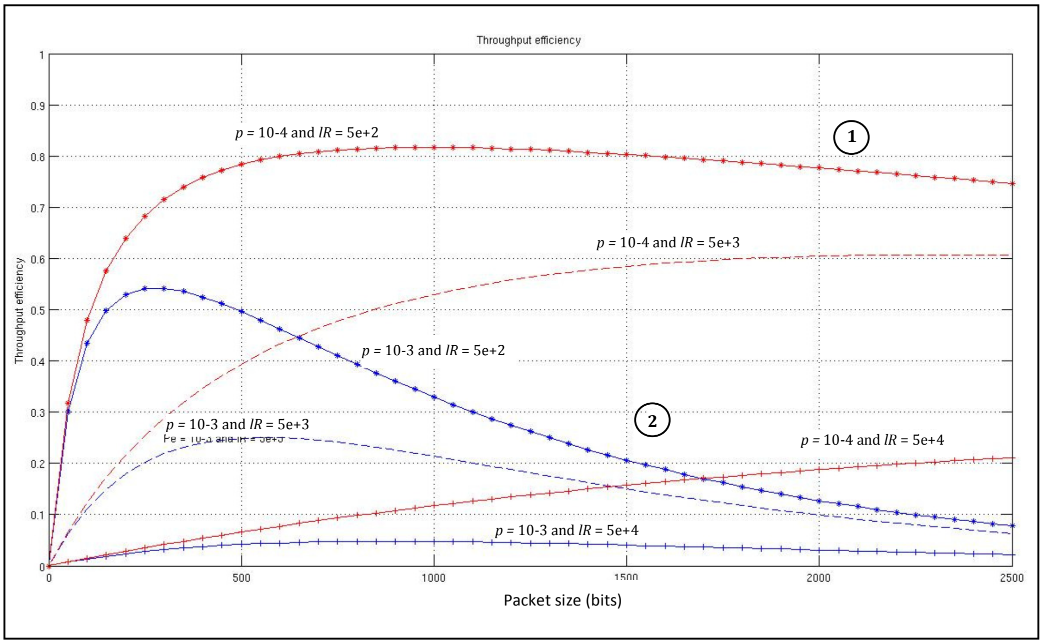


Figure 4.8: Throughput efficiency against N_{opt} under different p and lR products

Another interesting observation in Figure 4.8 is that with a constant BER i.e. by maintaining the link quality at a certain level, the peak throughput efficiency will drop when the lR product increases. For example, comparing the curves with p of 10^{-4} but with different lR of 5e+2, 5e+3, and 5e+4 the peak efficiency drops from 0.8 to 0.6 and down to 0.2. This issue can be explained by the fact that as the distance l increases (and/or the transmission rate of R is getting higher), the chances for data packets being dropped (due to higher bit errors) will also increase thus bringing along with it a poorer peak throughput efficiency.

It should also be noticed in Figure 4.8 that for a good quality link, for example for p of 10^{-4} (curve 1), the high throughput efficiency seems to be maintaining almost at a constant peak value. In other words the throughput efficiency is maintained very nearly at its peak even though the optimal packet size is increasing. This may suggest a fact that data packets with large size can be transmitted in a high quality communication link to attain high throughput efficiency.

4.3.3 Data Packet Size and Energy Per Useful Bit

Energy per useful bit was investigated in the perspective of energy efficiency in data transmission between a pair of source-sink nodes. Considering the limited battery power for underwater sensor nodes, energy efficiency is investigated to understand the role of optimal packet length in meeting the power constraint issue. By adopting the energy per useful bit (EPUB) definition for radio wave transmission proposed in [52] and [53], the relationship between data packet size and EPUB in the simulated underwater environment is shown in Figure 4.9. The adopted EPUB definition [52] is stated below here as,

$$EPUB = \left(\frac{N_l + N_{oh}}{N_l}\right)(P_{TX} + \xi \cdot P_{RX})T \qquad (4.2)$$

where,

N_l	is the data length bits in a packet
N_{oh}	is the overhead bits in a packet
T	is the bit time in seconds
P_{TX}	is the power of the transmitter in mW
P_{RX}	is the power of the receiver in mW
ξ	is the average proportion of time spent in receive mode divided by the time spent in transmit mode.

Based on the above definition, listed below are some of the essential parameters used for the simulation:

B_l : 0 – 2500 bits
B_h : 40 bits
T : 1/8.2 kHz
P_{TX} : 8000 mW
P_{RX} : 80 mW
ξ : 0.8

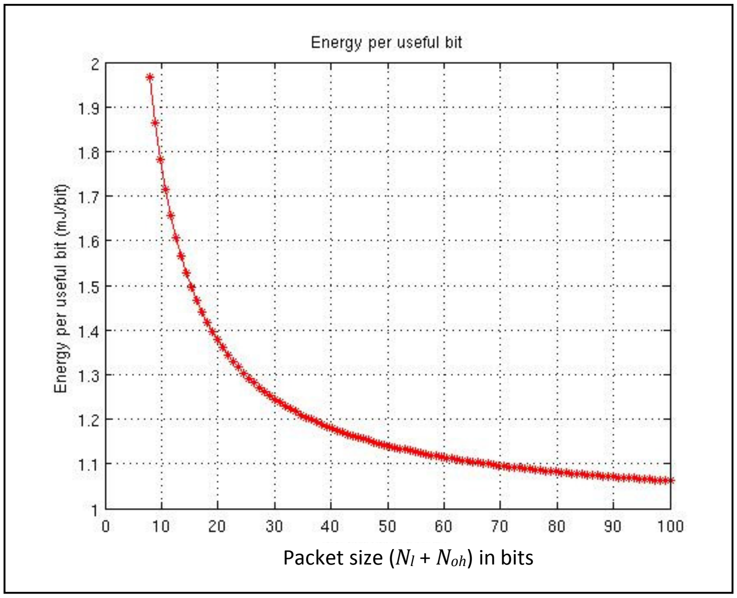


Figure 4.9: Energy per useful bit against packet size

The plot in Figure 4.9 clearly shows that if the bits are correctly delivered to the sink node, the small packet size will have high EPUB. It can be explained by the fact that since small packet tends to have less chances of being corrupted (bit lost) over the link, the EPUB tends to be high. However as the packet length gets larger for just over tens of bits the EPUB drops very quickly to a more constant level and slowly approaching 1 mJ/bit.

From a practical perspective, data packet sizes used in most of the transmission are very often more than hundreds or thousands of bits. For reader's information the header bits in the standard RTS packet is 160. EPUB is thus seemed not to be a very helpful metric to quantify the energy usage or constraint in data bits transmission. However EPUB is a useful parameter when comparison of BER performance is involved.

In this context, the energy efficiency equation (3.15) described in subsection 3.2.3 on page 55 would be the main reference for the simulation works described in this subsection for finding the relationship between energy efficiency and packet sizes. Equation (3.15) (page 55) depicts that energy efficiency is a function of packet length l and p (the communication link BER). Implicitly it involves the EPUB element. The equation is restated here for easy reference as

$$\eta = \frac{k_1 l}{k_1(l+\alpha)+k_2}(1-p)^{l+\alpha}$$

where,

l is the packet length
p is the BER
α is packet trailer bits

In the simulation it is assumed that the source and the sink nodes are of homogeneous type leading them to have the same equipment constants i.e. $k_1 = k_2$. So the energy efficiency term in the η equation i.e. the term:

$$\frac{k_1 l}{k_1(l+\alpha)+k_2} \tag{4.3}$$

can be approximated to $l/(l + \alpha)$ for $(l + \alpha) >> 1$. This is acceptable since in most of the practical applications, packet length is more than hundreds of bits. It is also similarly in line with the basic definition of energy efficiency where l should be larger than α. In simulating this energy efficiency, some of the essential parameters used are listed here:

Channel type	: underwater Shanon
Channel frequency	: 8.2 kHz
Signal bandwidth	: 6 kHz
Queue size	: 5
Link delay	: 0.01s
BER (ρ)	: 10^{-2} , 10^{-3} , 10^{-4}
Header (α)	: 40 bits
Length (l)	: 0 – 1000 bits
MAC	: Aloha

The simulation output is shown in Figure 4.10 in which the graphs in this figure strongly depict high energy efficiency for low BER. The energy efficiency to link with BER of 10^{-4} is almost two folds than those with BER of 10^{-2}. The efficiency drops very sharply for high BER when the packet length is increased beyond the peak.

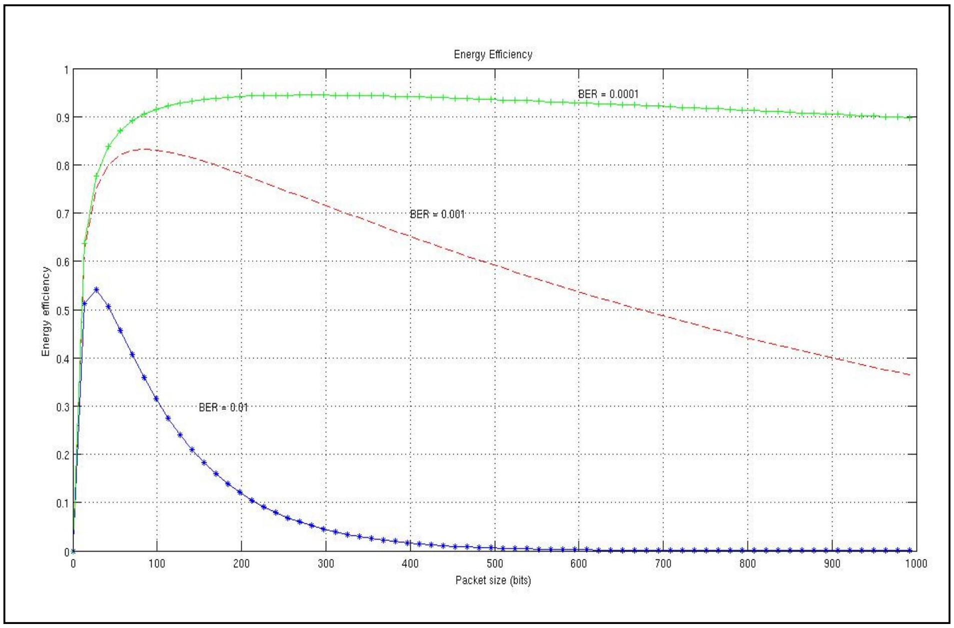


Figure 4.10: Energy efficiency against packet size under different BERs

This is practically true because the probability of packets being corrupted is high and the demand for retransmissions consequently increases and more energy is wasted then. Hence it is not surprising to observe that the energy efficiency gradually lessens more gently beyond the peak performance for the links with low BERs. The large packet length/size in good quality link in turn is able to attain higher energy efficiency than links with poor quality.

Now let's get back to the context of energy efficiency by considering equation (3.16) in subsection 3.2.3 on page 55. It relates the optimal packet size to the BER. The optimal packet size is using the derivative of the efficiency equation of (3.15). Equation (3.16) is restated as below with $C_0 = \alpha + k_2/k_1$.

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

Again, with homogeneous equipment for the source-sink pair C_0 is approximated to α, the packet trailer (or checksum bits) which may be kept as a constant. Thus l_{opt} in this case is a sole function of p, the BER.

The following parameters were used to simulate the relationship between l_{opt} and ρ in the context of energy efficiency:

Transmitter power	: 5.248mW
Channel type	: underwater Shanon
Channel frequency	: 8.2 kHz
Signal bandwidth	: 6 kHz
Queue size	: 5
Link delay	: 0.01s
BER (ρ)	: $3\text{x}10^{-2}$, 10^{-3} , $5\text{x}10^{-4}$, 10^{-4}
Header (α)	: 20 – 100 bits
MAC	: ALOHA

The outcomes of the simulation are shown in Figure 4.11. A straightforward manifestation from the graphs in Figure 4.11 is that a high quality link (low *p*) allows larger packet size in spite of the header length and the rate of increase for packet size under good link quality to be much faster than links with poor quality.

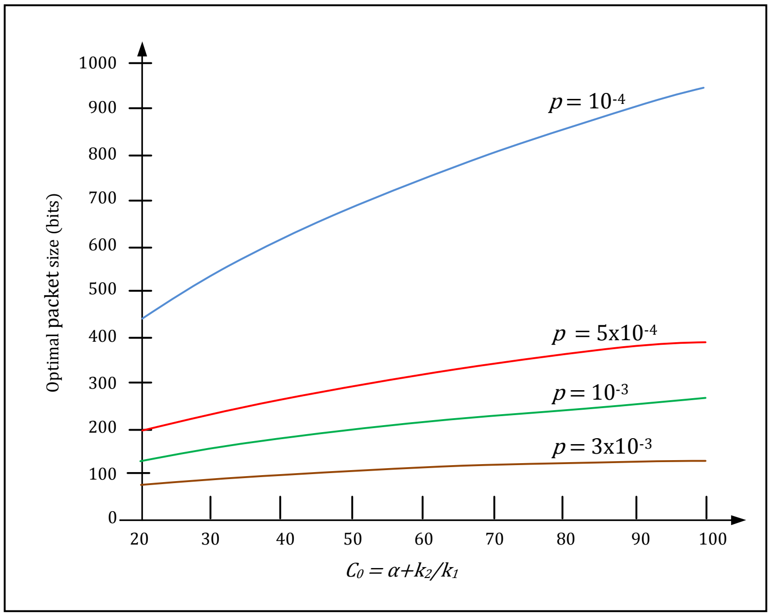


Figure 4.11: Optimal packet size against header length with different BERs

4.3.4 Optimal Packet Size Search Algorithm

The outcomes of the simulations described above have led the author to develop a data packet size optimization algorithm with performance metric of throughput efficiency and energy efficiency qualified under different bit-error-rate. The outline of this algorithm is presented in section 3.3.1 on page 57. The algorithm is proposed as follow.

Prerequisites:

A database consists of three data sets belonging to the three look-up graphs similar to Fig. 4.6, Fig. 4.7, and Fig. 4.10 are already in place. In practical case this database can be constructed using the data collected through an actual data collection or measurements in the designated sensor network area where the sensor nodes have been deployed (or to be deployed). Alternatively, the data sets could be constructed by simulation means for the body of water where the sensor nodes are to be deployed. Of course, other actual underwater environment factors from the designated areas need to be considered for the simulation.

The constructed database shall then be loaded into the memory of the underwater nodes as the “knowledge acquired” from measurements at the intended areas of sensor deployment. These are considered as the core knowledge to be used for finding the optimal packet size for effective and efficient data transmissions. It is worth to be mentioned that for UW nodes with memory constraint perhaps only simplified data sets shall be loaded into it but, of course, could be on the expense of reduced communication effectiveness.

The Algorithm

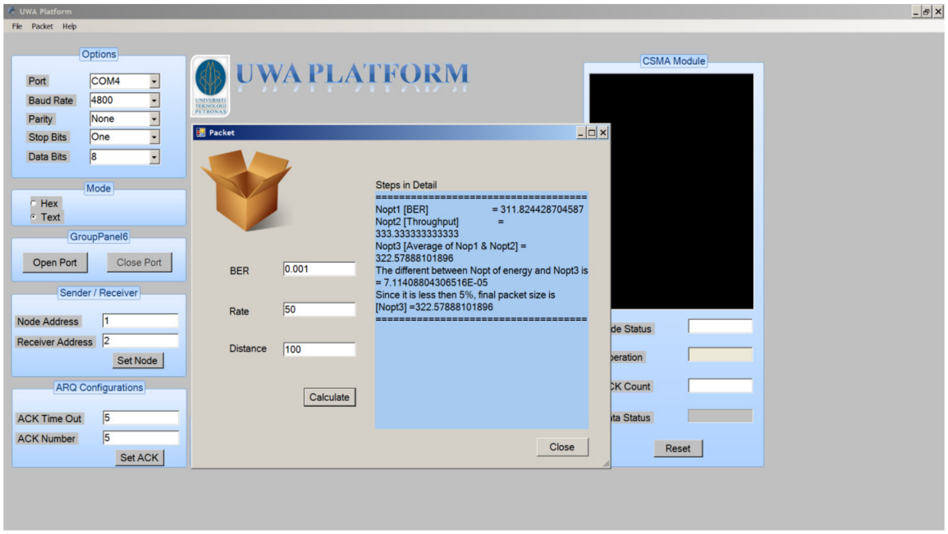
```
/*three data sets are denoted as $D_{46}$ , $D_{47}$ , and $D_{410}$ to represent Fig. 4.6, Fig. 4.7,
/*and Fig. 4.10 respectively
/* Source node and sink node are of homogeneous type
/*test packet is essentially the RTS packet format with header length ($h$) of 160 bits
{
1. Source node:
2.   send(test_packet) to the sink with predefined
        bit rate ($R$);
3. Sink:
     ack_and_return(test_packet);
4. Source:
     with returned packet computes:
        {
```

```
5.      BER ($p$);
6.      distance ($l$);
7.      with $p$ indexed into $D_{46}$ to acquire $N_{opt1}$;
8.      with $lR$ product indexed into $D_{47}$ to acquire $N_{opt2}$;
9.      $N_{opt}$ :=  average($N_{opt1}$ , $N_{opt2}$ );
10.     with $N_{opt}$ indexed into $D_{410}$ to acquire the energy
             efficiency ($\eta$);
11.     compute: difference between $\eta$ and $\eta_{opt}$ from $D_{410}$;
12.     if (difference) < (5%) then
13.     packetsize := $N_{opt}$
14.     else
          {
15.       with $p$ indexed into $D_{410}$ to obtain packet size
           ($N$) corresponds to max $\eta$ ;
16.       packetsize := average($N$, $N_{opt}$);
17.       } end_if
        }
18. Source: assemble data_packet with packetsize ;
19. Source: send (data_packet);
20. }End
```

The coding of this algorithm can be found in Appendix B and the samples of the algorithm run are presented as screenshots shown in Figure 4.12. By referring to the pseudo code above, the data packet size is firstly obtained with reference to the link *BER* in line 7. Then the packet size is qualified by *throughput effic*iency in line 8 and 9. The packet size at last is qualified by *energy efficiency* via lines 10 to 16. Hence, the author apparently called this algorithm as: "***Data Packet Size Optimization Algorithm With Performance Metric of Throughput Efficiency and Energy Efficiency Under Different Bit-Error-Rate***" or in short simply as "***2Q Optimization Algorithm***" because the optimum packet size is qualified by two performance metrics of throughput efficiency and energy efficiency.

Principle of the proposed algorithm is explained here with reference to Figure 4.12 (a) and (b). Figure 4.12 (a) shows a link with reasonable BER quality (p) of 0.001 and a moderate lR product of 5x10^4 m-bps whereas Figure 27 (b) shows a link with same lR product of 5x10^4 but with a below average link quality of 0.01.

In Figure 27 (a), with p of 0.001 the algorithm first looks up the BER data set in the database and returns an optimal packet size of N_{opt1} = 311 bits. The algorithm afterward looks up in the throughput efficiency data set with the lR of 5x10^4 to return an optimal packet length of N_{opt2} = 333 bits. So the optimal packet size, up to now, is qualified only by *throughput efficiency* metrics under p of 0.001. The algorithm then takes the average of 311 (N_{opt1}) and 333 (N_{opt2}) bits to give a N_{opt3} of 322 bits. With the size of 322 bits the algorithm then looks up in the related energy efficiency, η (the 2nd performance qualifier) in the *energy efficiency* data set. This η is used to compared with the peak η in the energy efficiency data set to see whether the difference between them is greater than 5%. It is found to be less than 5% in this sample. Thus the final optimal packet size then is decided as 322 bits. It is to be mentioned here that ±5% is considered a good tolerance factor in common engineering terms.



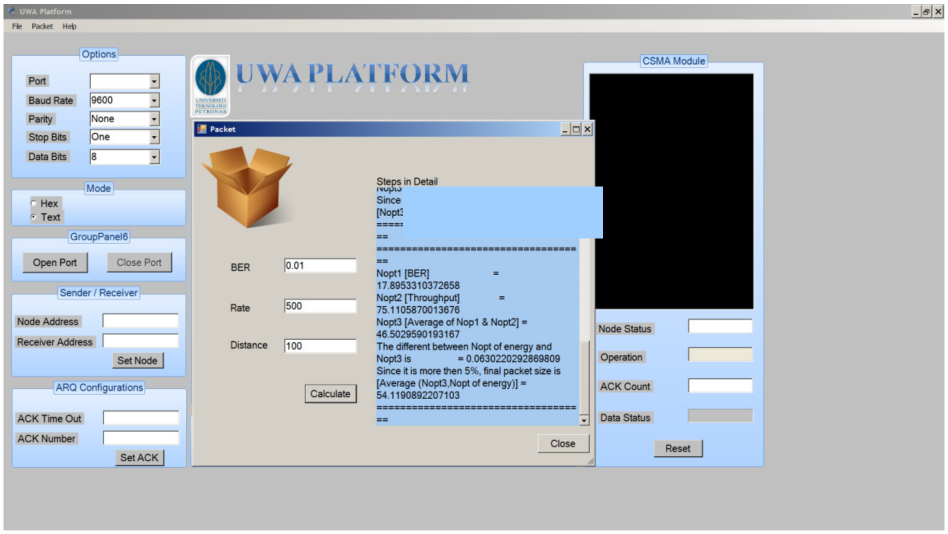
Port COM4
Baud Rate 4800
Parity None
Stop Bits One
Data Bits 8
Mode: ○ Hex ● Text
Node Address 1
Receiver Address 2
ACK Time Out 5
ACK Number 5

BER 0.001
Rate 50
Distance 100

Steps in Detail

```
========================================
Nopt1 [BER]                = 311.824428704587
Nopt2 [Throughput]           =
333.333333333333
Nopt3 [Average of Nop1 & Nopt2] =
322.57888101896
The different between Nopt of energy and Nopt3 is
= 7.11408804306516E-05
Since it is less then 5%, final packet size is
[Nopt3] =322.57888101896
========================================
```

(a) Difference of η and η_{opt} less than 5%



Baud Rate 9600
Parity None
Stop Bits One
Data Bits 8
○ Hex
● Text

BER 0.01
Rate 500
Distance 100

Steps in Detail

```
Nopt3
Since
[Nopt3
=====
==
========================================
==
Nopt1 [BER]                =
17.8953310372658
Nopt2 [Throughput]           =
75.1105870013676
Nopt3 [Average of Nop1 & Nopt2] =
46.5029590193167
The different between Nopt of energy and
Nopt3 is             = 0.0630220292869809
Since it is more then 5%, final packet size is
[Average (Nopt3,Nopt of energy)] =
54.1190892207103
========================================
==
```

(b) Difference of η and η_{opt} more than 5%

Figure 4.12: Samples of algorithm output

The same process is repeated for Figure 27 (b). Here, the difference in η is found to be more than the 5% threshold value i.e. at 6.3%, as the consequence, the final packet size needs one more adjustment with respect to the energy efficiency metric. The final optimal size is accordingly computed by taking the average of N_{opt3} and the packet size corresponding to the peak η in the energy efficiency data set to give an optimal size of 54 bits.

The explanation of the algorithm with the aid of the two screenshots clearly shows that the optimal packet size can be determined by the proposed algorithm using the two important performance metrics (under different BERs) of UWA communication link.

4.4 Chapter Conclusions

This chapter presents the outcomes of the simulation works conducted on ns-2 simulator with its MIRACLE package running on Ubuntu platform. The aim of the simulation is to verify and possibly to extend the discussions in chapter 3 on the relationships between optimal data packet size and the two important UWA communication performance metrics namely throughput efficiency and the energy efficiency under different communication BERs.

The simulated outputs are collected and used to construct a database which comprises of three important data sets relating optimal data packet size to the throughput efficiency, the energy efficiency, and BERs. The consolidated databases are used in the proposed algorithm to determine the optimal data packet size for underwater data packets transmissions. The principle of the algorithm is clearly described with two screenshots shown in Figure 4.12.

It should be noted that the screenshots shown here are the basic interface developed to test the viability of the proposed algorithm. The algorithm can be invoked by clicking the “Packet” option from the menu bar in the UWA Platform application. The smaller window on top of the UWA Platform is the interface showing the data entry boxes and the computed results from the proposed algorithm.