Islamic University of Technology

Effective Data Extraction from Underwater Using Sensor-Delay Tolerant Network (S-DTN)

A Thesis Presented to

The Academic Faculty

By

Nasif Ahmed Sharif, Student ID: 092436

Shihab Nayeem Ahmed, Student ID: 092409

Redwan Ahmed, Student ID: 092430

In Partial Fulfillment

Of the Requirement for the Degree

Bachelor of Science in Electrical and Electronic Engineering

Academic Year: 2012-2013

A Dissertation on,

Effective Data Extraction from Underwater Using Sensor-Delay Tolerant Network (S-DTN)

Submitted By

Nasif Ahmed Sharif (092436)   Shihab Nayeem Ahmed (092409)   Redwan Ahmed (092430)

Approved By

Dr. Md. Shahid Ullah
Head of the Department & Professor
Department of EEE, IUT

Dr. Khandokar Habibul Kabir
Thesis Supervisor & Associate Professor
Department of EEE, IUT

Abstract

Delay-tolerant networking (DTN) is a term invented to describe and encompass all types of long-delay, disconnected, disrupted or intermittently-connected networks, where mobility and outages or scheduled contacts may be experienced. 'DTN' is also used to refer to the Bundle Protocol, which has been proposed as the one unifying solution for disparate DTN networking scenarios, after originally being designed solely for use in deep space for the 'Interplanetary Internet.' We have evaluated the network to be used in underwater data extraction purposes.

Underwater terrain is very different from the terrestrial terrain, as it poses more amounts of obstructions where the normal protocols of networking tend to fail. DTN addresses this very problem through the hop-by-hop networking technique to extract data from deep sea and transport the data to the onshore sites for further analysis.

This paper has been aimed to provide the best possible solution model that can be designed using DTN to extract data from challenged underwater terrain.

Preface

The undergraduate thesis, "Effective Data Extraction from Underwater Using Sensor-Delay Tolerant Network (S-DTN)" has been written for the completion of Bachelor of Science degree at Islamic University of Technology, Bangladesh. This thesis work and writing has been done during the year 2013 under the supervision of Dr. Khandokar Habibul Kabir, Asst. Professor of the department of Electrical and Electronic Engineering.

We would like to dedicate this thesis to our supervisor Dr. Khandokar Habibul Kabir. Without the dedicated help of him we won't be able to complete this work. We are also grateful to all of our well-wishers, who provided their perpetual support towards accomplishing this task successfully.

Table of Contents

CHAPTER

1

Introduction

1.1 Introduction

The oceans alone cover about 70% of our planet and along with rivers and lakes are critical to our well-being. It is very much needed to understand their behavior and their eco system to fully appreciate and capitalize on the underwater system for betterment of human society. Underwater information can be related to the water temperature, pressure, seismic movements all of which contribute to human understanding and interpretation of data to compute the weather, map out oil rigs, prevent against earthquakes and tsunamis. It can also be used to monitor and model the behavior of underwater ecosystems to understand their behavior even further. The conditions in which this form of data collection takes place is very expensive and with the existing mechanisms it is not efficient enough for the money being spent behind the projects. In this thesis paper we attempt to simulate such a real world scenario where underwater data extraction takes place using the existing nodes but implementing a new form of networking that we call Sensor Delay Tolerant Network (S-DTN). Delay Tolerant Network (DTN) is a new communication paradigm that can span across multiple networks and cope with harsh conditions not envisioned in the Internet (TCP/IP) protocol model.

1.2 Overview

Underwater networks consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. Underwater sensor network can be used to collect data, monitor pollution, and explore the underwater environment, mineral resources and aquatic life under the sea.

1.2.1 Assumption

Our model makes use of the assumption that underwater sensor nodes and vehicles possess self-configuration capabilities i.e. they are able to coordinate their operation by exchanging configuration, location and movement information, and relay the monitored data to an onshore station.

1.2.2 Node characteristics

The sensor network consists of static and mobile underwater sensor nodes. The nodes communicate point-to-point with each other, and they broadcast using an acoustic protocol. The mobile nodes hover above the static nodes for ‘data muling’, and they can perform network maintenance functions.

1.2.3 Network selection

We want to choose the most widely used network protocol, TCP/IP for communication in the underwater scenario. But there are some assumptions regarding the characteristics of TCP/IP:

- Existence of an end-to-end path between sender and receiver.
- Maximum round trip between any pair of nodes is not excessive.
- End-to-end packet drop probability is small.

But underwater sensor network falls under the category of 'challenged networks' characterized by extremely limited end node power, memory capacity and are prone to discontinuous connection.

Hence an alternative networking protocol- 'Delay Tolerant Network' is our approach to account for the assumptions/limitations of TCP/IP.

Characteristics of Delay Tolerant Network:

- No end-to-end connectivity required.
- Long/Variable delays can be overcome.
- Presence of storage for every router.

1.3 Motivation

Our primary motivation is to make the extraction of information from underwater more effective and simple. Existing network protocols can be used effectively to extract data from onshore landscape .But the offshore scenario is not much effective .In underwater network, the links are not based on radio waves or optical waves. The only fruitful technology available is acoustic communication. But as we will shortly see there are some drawbacks regarding acoustic communication.

1.4 Research challenges

Major challenges in the design of Underwater Acoustic Networks are:

- Battery power is limited and usually batteries cannot be recharged
- The available bandwidth is severely limited
- Channel characteristics, including long and variable propagation delays, multi-path and fading problems
- High bit error rates
- Underwater sensors are prone to failures.

CHAPTER

2

Background Studies

2.1 Underwater data extraction - the traditional approach

The traditional approach for monitoring the underwater environment is to deploy underwater sensors that record data during the monitoring mission, and then recover the instruments.

Disadvantages of traditional approach:

- Real time monitoring is not possible. This is critical especially in surveillance or in environmental monitoring applications such as seismic monitoring. The recorded data cannot be accessed until the instruments are recovered, which may happen several months after the beginning of the monitoring mission.
- No interaction is possible between onshore control systems and the monitoring instruments.
- If failures occur, it may not be possible to detect them before the instruments are recovered. This can easily lead to the complete failure of a monitoring mission.
- The amount of data that can be recorded during the monitoring mission by the sensors is limited by the storage capacity of the sensors.

Hence, to account for the aforementioned drawbacks of the traditional approach, an underwater sensor network needs to be deployed which will allow real time monitoring, remote configuration and interaction with onshore control systems. This is traditionally obtained by connecting the sensors via wireless acoustic communication protocol.

2.2 UWASN - Underwater Acoustic Sensor Network

Wireless underwater acoustic networking is the enabling technology for Underwater Acoustic Sensor Networks (UW-ASNs) which consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. To achieve this objective, sensors and vehicles self-organize in an autonomous network which can adapt to the characteristics of the ocean environment.

2.2.1 Components for network formation

The list of required components are as follows:

Node

It is an active electronic device attached to a network capable of sending, receiving or forwarding information over a communications channel.

AUV

It is unmanned autonomous vehicle,that receives information from the nodes and relays the information to the base station above the sea level.

Static Nodes

Their position is fixed. They are anchored to the sea bed. There is no need for notification of location.

Kinetic Nodes

These nodes move with the current of water. Every node stays awake. Information is individually passed from these nodes to AUV.

A group of sensor nodes are anchored to the bottom of the ocean with deep ocean anchors. By means of wireless acoustic links these sensor nodes are connected to a central kinetic node, sometimes known as an underwater sink. These sinks relays data from the ocean bottom network to an Autonomous Underwater Vehicle (AUV).The data is relayed by the AUV to the surface station, which is above the sea level.

2.2.2 Why acoustic?

In terrestrial sensor network, communication is basically radio based. This is due to the relatively low power needed to transmit radio messages and basically the omnidirectional nature of radio propagation. Unfortunately, the majority of the electromagnetic spectrum is significantly attenuated by seawater, making radio communication impractical in underwater networks. Optical communication might be an exception. The primary advantage of optical communication is the higher theoretical data rate due to the higher frequency signal, while the disadvantages are range and line-of-sight operation. They are also affected by scattering. Hence in underwater networks, wireless communication is typically based on acoustic links.

2.3 Limitations of UWN with respect to terrestrial network

Many researchers are currently engaged in developing networking solutions for terrestrial wireless ad hoc and sensor networks. Although there are many recently developed network protocols for wireless terrestrial sensor networks, the unique characteristics of the underwater acoustic communication channel, such as limited bandwidth capacity and variable delays, require for very efficient and reliable new data communication protocols.

The main differences between terrestrial and underwater sensor networks can be summarized as follows:

- Underwater sensors are more expensive than terrestrial sensors.
- In underwater networks the sensor nodes are more sparsely dispersed.
- The information between the nodes cannot be correlated due to higher distance between them.
- Higher power is needed in underwater communications due to higher distances and for more complex signal processing at the receivers.

2.3.1 Assumptions of TCP/IP in terrestrial network

The communication model of the Internet is based on some inherent networking assumptions. These include:

- Existence of a continuous, bidirectional end-to-end path between two nodes
- Relatively short round-trip delays
- Symmetric data rates
- Low error rates.

These assumptions led to the design of a store-and-forward approach: intermediate nodes receive small fragments of information (packets) and forward them to next hop as fast as possible. Each packet is only transiently stored in a network device.

2.3.2 Sensor networks: A class of challenged networks

These networks are frequently characterized by extremely limited end-node power, memory, and processing capability. In addition, they are envisioned to exist at tremendous scale, with possibly thousands or millions of nodes per network. Communication within these networks is often scheduled to conserve power.

2.3.3 Major challenges in the design of underwater acoustic networks

The major challenges in the design of Underwater Acoustic Networks are as follows:

- Battery power is limited and usually batteries cannot be recharged
- The available bandwidth is severely limited
- Channel characteristics include long and variable propagation delays, multi-path and fading problems
- High bit error rates
- Underwater sensors are prone to failures.

2.4 Introduction of delay tolerant network

A delay-tolerant network (DTN) is a network of regional networks. It is an overlay on top of regional networks. The communication characteristics are relatively similar within a communication region.

2.4.1 Bundle layer

The unit of information exchange in a DTN is a **bundle**. A DTN **node** is an entity with a bundle layer. A node may be a *host*, *router*, or *gateway* (or some combination) acting as a source, destination, or forwarder of bundles. A router forwards bundles **within** a single DTN region while a gateway forwards bundles **between** DTN regions.

In a typical network, applications communicate using a common set of network layers. In a DTN, the bundle layer is placed below the application layer and hides the communication layers specific to that network/region, as depicted in Figure 1. A network-specific convergence layer is used underneath the bundle layer as to interface with each different network layer protocol used.

The unique characteristic of the bundle layer is the support for in-transit storage. Bundles received from a sender can be stored in an intermediate node for an excessive amount of time (minutes, hours, or even days). These store operations are performed by the network stack, at the bundle layer. The in-transit storage is the means to overcome the delays and disruptions induced while a bundle moves hop by hop to its final destination. This allows it to avoid costly end-to-end retransmissions due to errors or timeout; and to allow information exchange between two nodes that share no end-to-end communication path at any given time moment. The bundle protocol defines the custody operation which allows an intermediate node to handle bundle delivery to final destination on behalf of a more distant sender.

So the characteristics of Delay Tolerant Network can be summed up as:

- No end-to-end connectivity required.
- Long/Variable delays can be overcome.
- Presence of Storage for every Router.

2.5 Formation of UW-SDTN

We apply the concept of underwater acoustic sensor network (UWASN) in collaboration with DTN to form Underwater Sensor Delay Tolerant Network (UW-SDTN). In this approach we first deploy static sensor nodes. These static nodes are anchored to the ocean bottom or sea bed and are used for data extraction. A group of static nodes surrounds a central node. This static central node is used to relay the data accumulated / collected from the static corner sensor nodes to the surface station using multi-hop paths. The multi-hop path is established by using Autonomous Underwater Vehicles (AUVs). Data is transferred from the central node to the AUVs which relay it to a boat at the surface. Transmission of data is completed when data is ultimately received at the surface sink. To minimize transmission delay, multiple AUVs and boats can be used.

2.5.1 Advantages of this approach

- Less energy consumed as only certain nodes (central) need to send long range signals.
- Less probability of data overflow as the corner nodes pass information to the central node, having higher storage capacity.
- Data correlation is possible as the sensor nodes are placed in small groups in a particular area.

2.5.2 Assumptions for the formation of S-DTN

- Instantaneous data transfer between the different components of the network
- Uniform sea floor
- Movement of AUV is not affected by water current flow or marine life
- Boat speed constant
- A 2-dimensional model to simplify a 3-dimensional real world situation

CHAPTER

3

Our Proposed Network

3.1 Objective

The primary objective of using the network is to collect underwater data, sense and record them through the deployed sensors and transport them to sinks located on-shore. At first, it is essential to decide which part of the vast ocean is needed to be monitored. This is depended on the application for which the network is to be established at the first place.

3.2 Real world scenario

Real World Scenario describes a real world setting where we are interested to implement the network.

3.2.1 Placement of sensor nodes

In our chosen scenario, we would like to monitor ocean around few hundred meters off the coast. Thus the sensors are needed to be deployed in such a way that they form a line that is almost parallel to the coastline. Around each central sensor node, six corner nodes are placed considering a small radius, in a circular path, on the circumference of the central node with equal distances among them as depicted in fig 3(a). Each corner node will record data and transfer them to its respective central node. Number of central nodes to be placed is subject to the length of ocean to be monitored and the maximum allowable node-to-node separation.

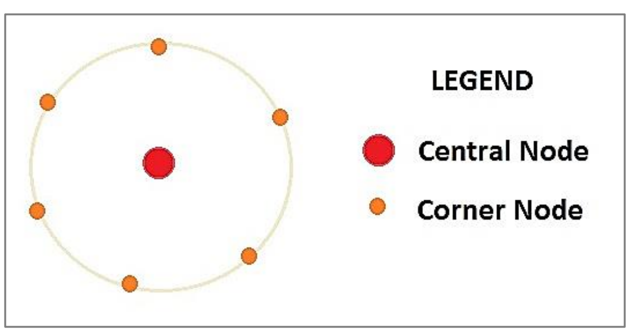

Fig. 3(a)

Knowing the depth of water at that particular distance from the coastline is of paramount importance. The seabed is not uniform and also there are waves which makes it very difficult to measure the actual depth at every point along the line. In that case, simplification of the scenario and an approximation of the depth can be very useful for the sake of advancing with the analysis.

3.2.2 Communication in the network

In underwater network, using acoustic waves is the most preferred form of communication. Since the channel is cause to experience strong signal attenuation and low data rates compared to terrestrial communication, selection of sensor ranges of all acoustic modems are vital to designing a network.

3.2.2.1 Purpose of AUV

When the ocean depth is significantly high, it is impractical to assume that every central node will forward data to the boat on the surface in a single hop. Due to the limited transmission range of sensor modems, at least one intermediate node has to be there to make up for this constraint. This is where the Autonomous Underwater Vehicle (AUV) comes in. It is basically a crewless, untethered submersible robot which operates independent of direct human control.

The AUV is programmed to move backwards and forwards over a particular path, keeping a fixed height from the sea floor. It is important to note that the AUV has to be within the range of at least one sensor node at all times as it moves along the path. Furthermore, the AUV has to have sufficient transmission range so that it can communicate with the node i.e. usually a boat above it. And if it fails to cover this distance another AUV can always be placed above the existing AUV to take care of the range gap.

3.2.2.2 Movement of boat

The boat is considered to use two forms of network – acoustic and radio waves. Acoustic to communicate with the AUV and radio waves for terrestrial communication with the sinks on-shore. Suppose there exists two on-shore sinks at point A and point B, lying on the coastline, separated by few kilometers of distance between them. We think of a boat to start from point A as it moves roughly in a parabolic path to reach the shore at point B as shown in Fig. 3(b) and Fig. 3(c), cruising above line where nodes were deployed and participating in communication with AUV and sinks whenever they lie within range.

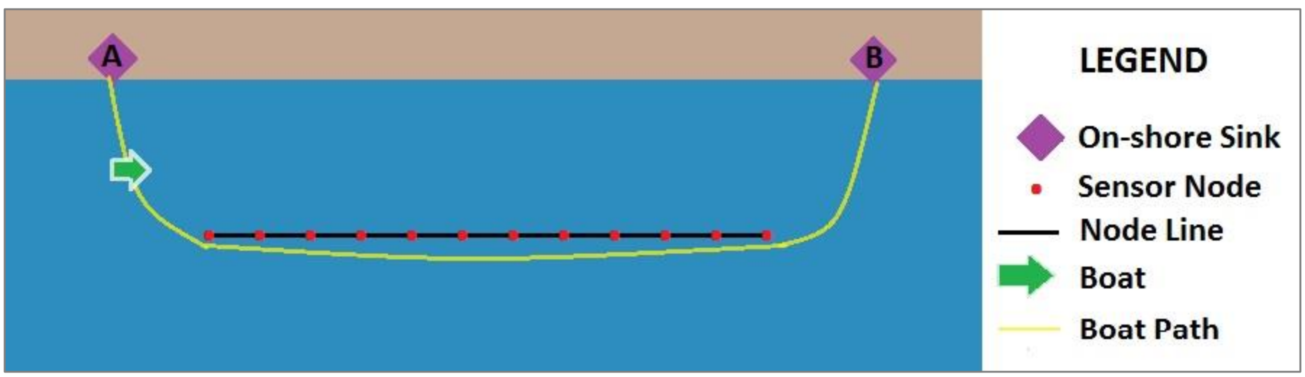

Fig. 3(b) Real world scenario, view from top

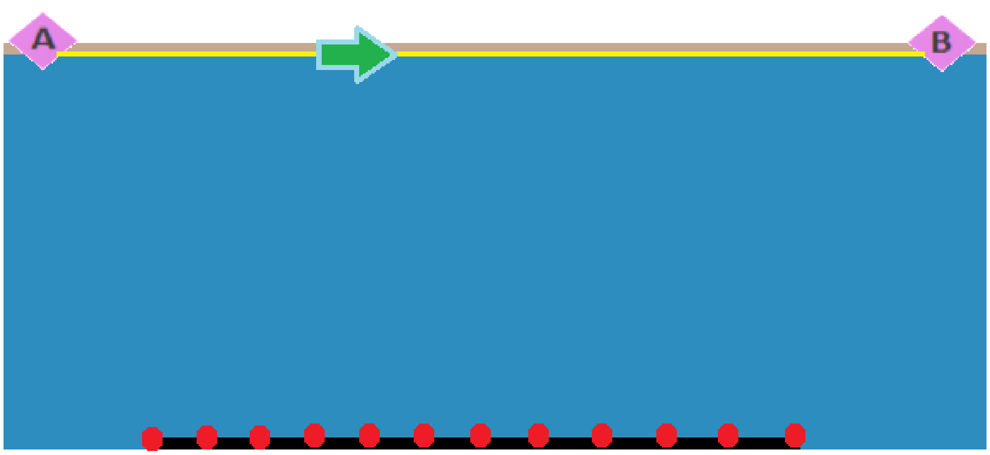

Fig. 3(c) Real world scenario, view from side

3.2.3 Application of delay tolerant network

Creating an effective network in this kind of challenged environment is quite troubling since an end-to-end connectivity cannot be assured at all times. Sensor node generates data continuously but cannot transmit it instantaneously because an AUV is not always there within its range to receive it. Therefore sensor node needs to store the data volume in its storage until it sees an AUV. However, this kind of operation is not supported by the existing TCP/IP protocol. The necessity of a "store-and-forward" approach of data transfer is met by a new standard in communication called Delay Tolerant Network (DTN).

In simple terms, each nodes that holds data is called "custodian" of those data. The custodian keeps the data in its storage until it receives a proper acknowledgement after successfully forwarding the data to the next node. The next node now becomes the custodian, this is called "custody-transfer". Likewise, the data moves along the network until it reaches the end point receiver, i.e. in our case the sinks located on-shore. Application of DTN makes networking simple and possible, it minimizes packet loss and improves overall efficiency of the network.

CHAPTER

4

Building the Simulator

4.1 Background

To test for performance and effectiveness of our proposed network model, it is needed to be implemented in real world and only through vigorous experimentation and subsequent result analysis, the rationality of the network can be judged. Unfortunately, due to limitations of time and money it was not possible for us to carry out the experiments materially. But there is a plausible alternative approach, which is to create a much simplified simulated environment of the real world scenario where the network is chosen to be implemented. This inspired us to build an underwater network simulator from scratch, particularly for our real world scenario.

4.2 Simulator developed in Netlogo

For realistic imitation, the network model has to be three-dimensional i.e. 3D to account for distances in all three directions. But for simplification of modeling and through consideration of few assumptions, it seemed reasonable to do the modeling in two-dimension i.e. 2D.To develop our simulator, we have used *Netlogo.* It is basically an agent-based programming language and integrated modeling environment. *NetLogo* is a free and open source software, under GPL license. It is written in Scala and Java and runs on the Java Virtual Machine. Netlogo can be downloaded from the website http://ccl.northwestern.edu/netlogo

The figure in the following page Fig. 4(a) shows a sample simulator world view, formed after configuring different variables during setup procedure.

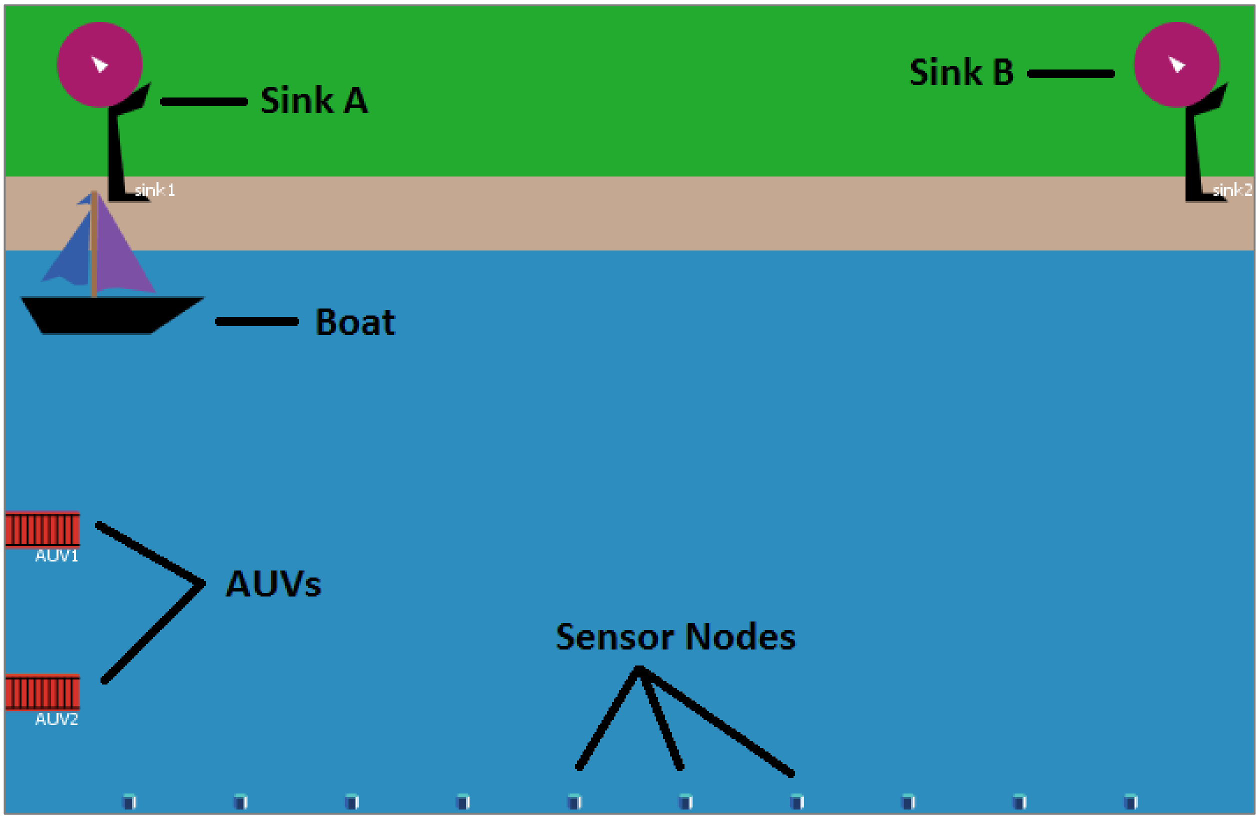

Fig. 4(a) World view in simulator

4.2.1 The Interface

Another important part of the simulator is its interface which is like a control panel where all different sliders, chooser, buttons and switches exist. Fig. 4(b) illustrates the graphical user interface (GUI) and part of the interface consists of monitors and plot viewer as depicted in Fig. 4(c).

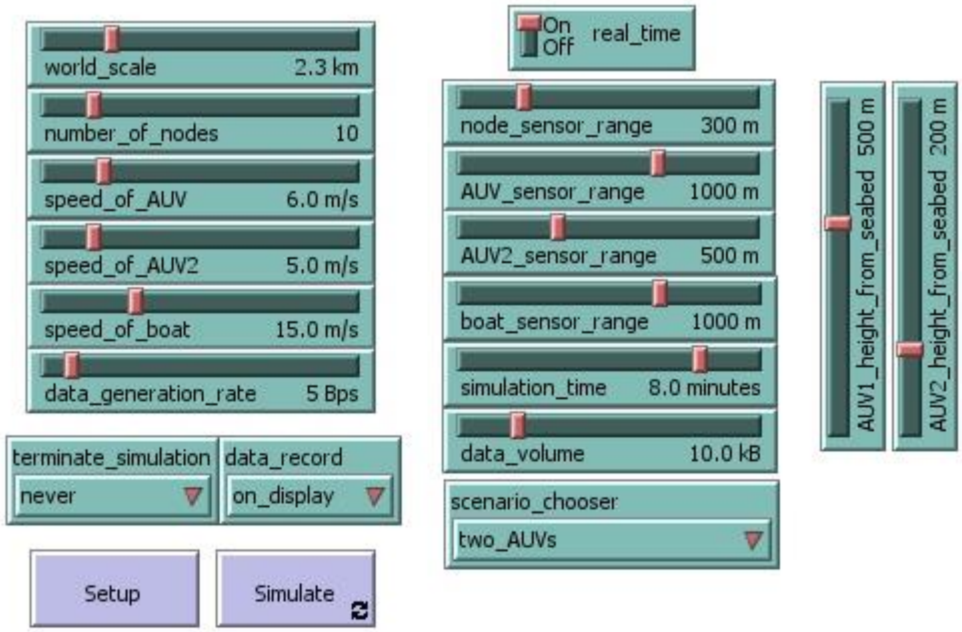

Fig. 4(b) Graphical User Interface – Setup and Control Panel

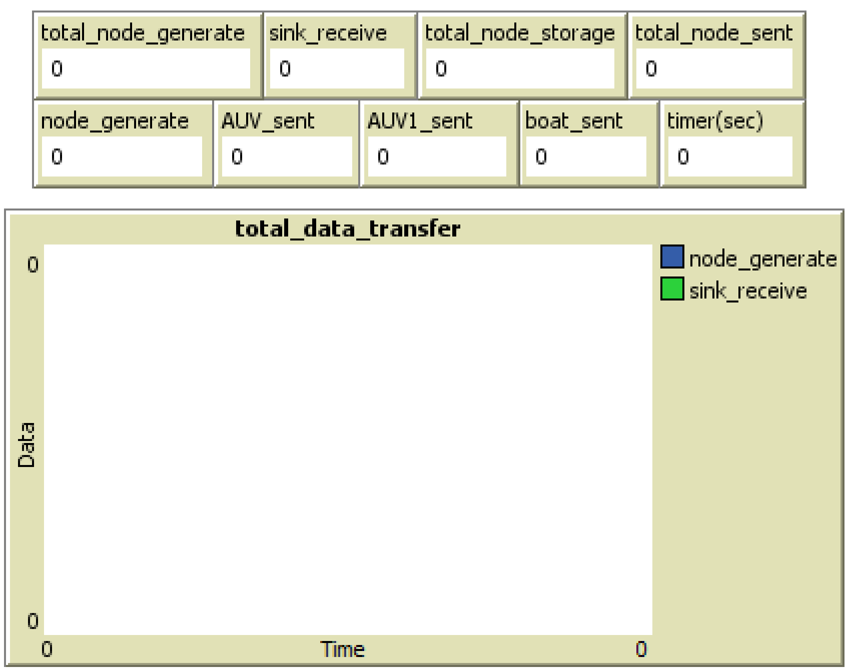

Fig. 4(c) Graphical User Interface – Monitors and Plot Viewer

4.2.1.1 Sliders

Netlogo defines sliders as global variables, which are accessible by all agents. They are used in models as a quick way to change a variable without having to recode the procedure every time. Instead, the user moves the slider to a value and observes what happens in the model.

In our simulator, we have used sliders to control the following variables:

World Scale – The Netlogo world is made up of cells called patches. Each patches have its own identity defined by its co-ordinates. In our model, there are a total of 1683 patches, 51 patches along x-axis and 33 patches in the y-axis, giving the world frame a rectangular shape. The horizontal axis is scaled by a scaling factor, calculated by using the value (in km) set by this slider, so that each patch defines real world length in meters.

scaling_factor = (world-width / (world_scale * 1000))

World-width means the total number of patches that exist in the horizontal axis. In our case, it is 51.The vertical axis is scaled by using the same scaling factor.

Number of Nodes–The user can place any number of sensor nodes by sliding the number_of_nodes slider before setting up the simulator. Number of sensor nodes has to be even numbered. The in-built algorithm evenly distributes them in the seabed space. The spacing between every adjacent node is same.It is to be noted that, only central nodes are considered by the simulator, in fact there exists six additional corner nodes surrounding every central node.

Speed of AUV 1– The user has the opportunity to set speed_of_AUV of his/her choice through use of this slider. The speed is calibrated in meters per second using the same scaling factor used to define world dimension. AUV 1 lies above AUV 2, so naturally AUV 1 is considered to have greater speed than AUV 2. Generally, speeds of AUV are much lower than other watercrafts like boats, submarines etc.

Speed of AUV 2 - The user has the opportunity to set a speed of his/her choice using speed_of_AUV2 slider. Again, the speed is calibrated in meters per second using the same scaling factor used to define world dimension. AUV 2 lies below AUV 1, so naturally AUV 1 is considered to have greater speed than AUV 2. It is AUV 2 that communicates directly with the nodes and perform 'data muling'.

Speed of Boat–The speed of boat is to be defined by the user. Again, the speed is calibrated in meters per second using the same scaling factor used to define world dimension. There can be more than one boat in the scenario, in that case, all boats are assumed to cruise at the same speed as set by speed_of_boat slider.

Data Generation Rate–Sensors in node are programmed to sense and record numerous underwater data. We assumed these generated data by each node as chunk of bytes that are created periodically each second. By the data_generation_rate slider user can arbitrarily set this rate in unit of bytes per second.

Node Sensor Range - All underwater devices and vehicles use acoustic waves for communication. Acoustic waves have limited reach well defined by the range parameter of the acoustic modem used in the system. If any one of the two systems move away while communicating, the link between them would be disrupted as their separation distance exceeds the reach of the low-range device. Each node creates link with AUV 2 as the vehicle becomes available within the range of node which is defined by node_sensor_range parameter. Node modem range is assumed to be lower than AUV 2's range. Although the user has absolute freedom to select any range of his/her choice for this parameter. Data transfer occurs only when node and AUV 2 have an established link between them. The simulator represent links by a thick yellow lines as shown in Fig. 4(d).

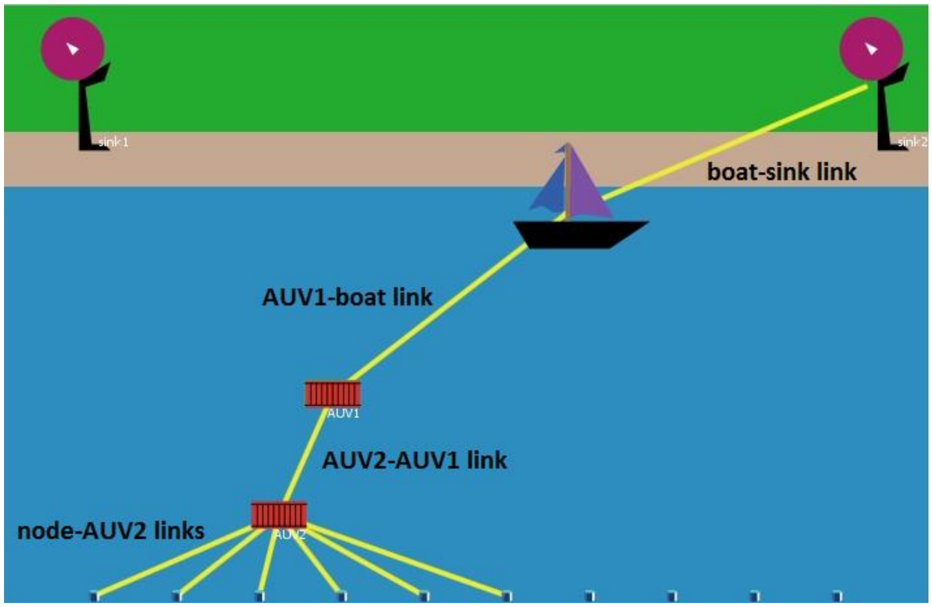

Fig. 4(d) Existing links are presented by thick yellow lines

AUV 2 Sensor Range – AUV 2 connects with both sensor nodes and AUV 1. Data bytes generated by the nodes are stored in each node's storage until AUV 2 is within reach, hence data is forwarded to AUV 2's storage. Usually, multiple node-AUV 2 links are existent at all times. These links are continuously being made and broken with the movement of AUV 2. By default AUV 2 is set to have higher range than sensor nodes. Thus these links' existence depends upon the each node's transmission capability. On the other hand, AUV 2 itself forwards its stored data to the next hop, in this case AUV 1, whenever AUV 2-AUV 1 link is established. This link's existence depends upon AUV 2 sensor range as set by the user using AUV2_sensor_range slider.

AUV Sensor Range - AUV 1 communicates with both boat and AUV 2. Data bytes collected by AUV 2 from the nodes are stored in AUV 2's storage until AUV 1 is within its reach, hence data is forwarded to AUV 1's storage. This way AUV 1 becomes the custodian of the successfully received data. By default AUV 1 is set to have higher range than AUV 2. Thus this link's existence depends on AUV 2's transmission capability. On the other hand, AUV 1 itself forwards its stored data to the next hop i.e. the boat whenever boat-AUV1 link is established. This link's existence depends upon AUV 1 sensor range as set by the user using AUV_sensor_range slider.

Boat Sensor Range - As mentioned earlier, boat uses two forms of communication using both acoustic waves in underwater communication with AUV 1 and radio waves in Wi-Fi terrestrial communication with sinks on-shore. Data bytes stored in AUV 1 storage is received by boat whenever boat is available within AUV 1's reach. The quality of terrestrial communication is much better in comparison with underwater network. The user specified boat_sensor_range parameter actually defines the Wi-Fi range of boat-sink links made in the air.

AUV 1 Height from Seabed – The position of AUVs are imperative to designing an underwater network. Proper placement of AUV is made through exploitation of the ocean's depth. This would maximize the utilization of the discontinuous links that are created. Users can define AUV1_height_from_seabed value in meters using this slider.

AUV 2 Height from Seabed - Users can define AUV2_height_from_seabed value in meters using this slider. Again, the placement of AUVs are imperative to designing an underwater network. Distance of AUV from sea floor along with sensor ranges governs the duration of survival of link between two bodies. By identifying depth of ocean, one can easily calculate the distance of AUV 2 from water surface. Almost all AUVs in practice have limited depth of operation, this reason accentuates the importance of setting this parameter sensibly.

Simulation Time – Once any simulation is run, it is needed to be terminated either manually or automatically. A running simulation can terminate spontaneously when situation arise as set by a predetermined condition. The simplest way to do this is to affix a duration of time for which simulation will run and stop when timer expires. User can set simulation_time slider in minutes to do the above.

Data Volume – There is another approach for terminating a running simulation. For instance, if the user wants to see how much time it takes to transfer a known volume of data, he/she can easily do that by setting a fixed amount of data with the help of data_volume slider for reception at the sink. When sink_receive parameter reaches data_volume value, simulation expires instantaneously irrespective of amount of data actually been generated by the nodes at that particular time.

4.2.1.2 Switches

In Netlogo, switches are a visual representation for a true/false global variable. User may set the variable to either on (true) or off (false) by flipping the switch.

In our simulator, we have used only one switch:

Real Time Switch–The simulator has been developed by keeping the ease of use and flexibility in mind. While setting up the simulator, users can easily implement determined values for variables before actually running the simulation. But provision is there to manipulate several different parameters while simulation is already on the run. Through the use of real_time switch, i.e. by setting it to 'On' one may vary sensor ranges and speeds of AUVs, boat etc. But during normal operation the switch is set to 'Off'.

4.2.1.3 Choosers

In Netlogo, choosers let user to choose a value for a global variable from a list of choices, presented in a drop down menu. The choices may be strings, numbers, booleans, or lists.

In our simulator, we have used the following switches:

Scenario Chooser - Choosers are basically drop-down menus from where user has to select any one of the choices on display. Scenarios are different models of network implemented in the same world space to test for their effectiveness. We have created four basic scenarios for the user to choose from. However, creating additional scenarios is pretty straightforward once user can think of one. With little programming knowledge, an advanced user is supposed to be capable of doing so. The built-in scenarios in scenario_chooser include basic – involving just one AUV, two AUVs- involving two AUVs one above another in layers, three AUVs –same as two AUVs case but involving an additional AUV in the bottom layer to take care of other half of the nodes and finally two_boats – it is exactly same as three_AUVs scenario but involving 2 boats instead of one, the second boat is considered to be cruising in from opposite direction.

Data Record - The two most important information from simulation are the input i.e. the total volume of data generated by nodes and output i.e. the total data received by the sinks on-shore with respect to time. In the simulator, there are two ways to have this information. If data_record is set to 'on_display' mode, these information are shown in monitors and plot viewer in numerical and graphical representations respectively. Whereas by selecting 'file_save' option, the user can have the chance to export information in comma separated values (.csv) format in tabular form. CSV files can be read by Microsoft Excel or Matlab, where further analysis can be carried out with user's freedom.

Terminate Simulation – As mentioned earlier, there are two ways to terminate a running simulation automatically. Choosing between 'elapsed_time' and 'data_volume' option, users can stop simulation either by setting simulation_time or data_volume sliders respectively. The third option is "never" as it signifies the simulation is required to be stopped manually by clicking 'Simulate' button on the interface.

4.2.1.4 Buttons

Netlogo defines button as either *once* or *forever*. When user clicks on a once button, it executes its instructions once. The forever button executes the instructions over and over, until user clicks on the button again to stop the action.

In our simulator, we have used the following two buttons:

Setup Button - With the hit of the setup button, blank modeling environment of Netlogo gets filled with the desired underwater network scenario. Setup operation goes through many procedures to finally build the world according to user's input in the interface. To name a few, setup button resets the timer, zeros all data count, check for status of real time switch, data record mode, and most importantly models ocean, shore, creates sinks, boat, AUVs, nodes etc. Once setup button is pressed, the simulator is ready to perform simulation on the scenario displayed on-screen.

Simulate Button – This button essentially starts the simulation. Behind this button works many procedures like one that is responsible for moving the AUVs, launching the boat, generating data in each nodes, performing data transfers among linked bodies etc. Simulate button is a *forever* button, pressing this button again would promptly halt the running simulation.

4.2.1.5 Monitors

Netlogo describe monitors as display that shows value of any reporter. The reporter could be a variable, a complex reporter, or a call to a reporter procedure. Monitors automatically update several times per second.

In our simulator, we have used the following monitors:

node_generate – A Monitor displays real time value of an assigned variable while the simulation is running. In our simulator, each node is programmed to generate a chunk of bytes every second. The node_generate monitor displays the total data generated by each node with the passage of time.

total_node_generate – If total number of nodes in the model is greater than one, which is the case in practical networks, the total_node_generate monitor shows the sum of data generated by the all the nodes in the network combined. This value may be considered as the input of the system.

total_node_generate = (node_generate * number_of_nodes)

sink_receive – Data transmission is considered successful once it reaches final destination terminal through the intermediaries in the system. In our case, data obtained by either of the sinks is considered as received data, and this can be thought of as output of the system. Therefore, sink_receive monitor displays the total data received by the both the sinks combined.

total_node_sent – AUV 2 collects data from sensor nodes as it moves along its path. The sum of all data bytes sent by all the nodes combined is displayed in the total_node_sent monitor.

AUV_sent - AUV 1 collects data from AUV 2 when AUV 1-AUV 2 link is existent. The data bytes sent by AUV 2 to AUV 1 above it is displayed in the AUV_sent monitor.

AUV1_sent – The boat collects data from AUV 1 when boat-AUV 1 link is existent. The data bytes sent by AUV 1 to boat above it is displayed in the AUV1_sent monitor.

boat_sent – The on-shore sinks accumulates data from boat as the boat moves along its path. The sum of data gathered by both the sinks is displayed on the boat_sent monitor.

total_node_storage – As previously mentioned, nodes can only transfer data when an AUV is within reach. To keep hold of its unremittingly generated data, nodes requires to store data immediately after generation. The total_node_storage monitor displays total stored data of all nodes combined.

timer – The timer is triggered by the hitting the setup button. It shows elapsed time after setup button has been pressed in seconds. This value is vital to keep track of the simulation duration.

4.2.1.6 Plot

Netlogo defines plot as a graphical view of data the model is generating.

We have used only one plot window in our simulator.

total_data_transfer – The total_data_transfer window represents two graphs namely total_node_generate and sink_receive on the same time scale. The graphs are plotted in real-time as the simulation progresses.

CHAPTER

5

Data Collection & Result Analysis

5.1 Technical Specification

To make our simulation realistic, we have gone through many technical specifications of practically existing sensor nodes, AUVs etc. We fetched data from the tech sheets and implemented those into simulator.

Given below is a summary of the properties of nodes, AUVs and boat modem,

Nodes:

- Maximum Operating Depth: 200m~1km
- Acoustic Modem Range: 250m~1km
- Bit rate: 25~100bps

Autonomous Underwater Vehicle:

- Maximum Operating Depth: 200m~1km
- Acoustic Modem Range: >1km
- Bit rate: >320kbps
- Speed: 1-4 knots

Boat:

- Acoustic Modem Range: >1km
- Bit rate: >320kbps
- Speed: 10-25 knots
- Boat to sink connection: Wi-Fi

5.2 Simulation Parameters

For running our simulation, we input the following values of different parameters:

Dimensions:

Length of ocean = 2.3 kilometers

Ocean depth = 1037 meters

Separation between sinks = 1804 meters

Node to node gap = 205 meters

Depth of AUV2 from surface = 837 meters

Depth of AUV1 from surface =537 meters

Speeds:

Speed of AUV1 = 6m/s = 11.66 knots

Speed of AUV2 = 5m/s = 9.72 knots

Speed of boat = 15m/s = 29.15 knots

Sensor ranges:

Node sensor range =300 meters

AUV sensor range = 1000 meters

AUV2 sensor range= 500 meters

Boat sensor range = 1000 meters

Others:

Number of nodes = 10 units

Node data generation rate = 5 bytes per second

5.3 Scenario Comparison

Using our simulator we basically created four scenarios to test for their performance and effectiveness.

5.3.1 Scenario 1: One AUV and one boat

In scenario 1, only one AUV and one boat is present for data transfer see Fig. 5(a) and Fig. 5(b). Given that the range of AUV is limited, boat-AUV link exists for only a very short amount of time.

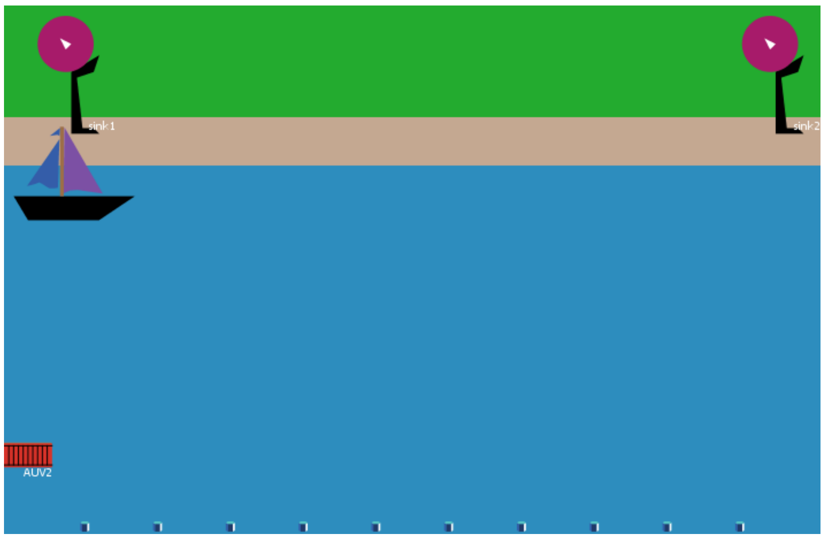

Fig. 5(a) Scenario 1: One AUV and one boat

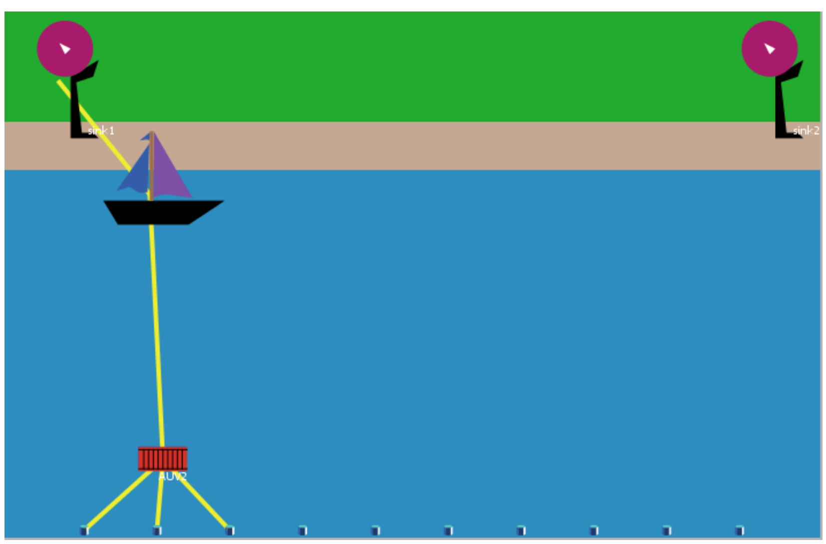

Fig. 5(b) Scenario 1, a moment during simulation, links shown

5.3.2 Scenario 2: Two AUVs and one boat

In scenario 2, two AUVs are in stack format and a boat is present for data transfer. So to compensate for the limited range of AUV acoustic signals we stack two AUVs to move across the entire length of our selected sea bed, see Fig. 5(c) and Fig. 5(d).

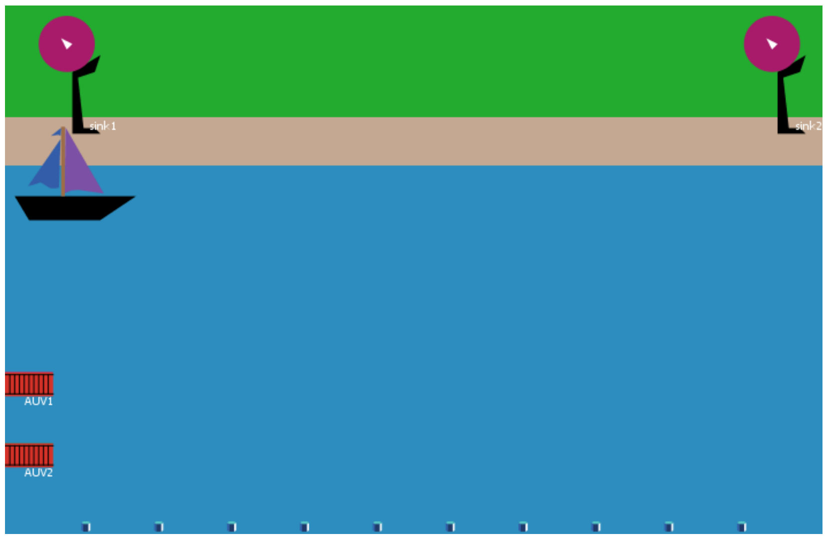

Fig. 5(c) Scenario 2: Two AUVs in layers and one boat

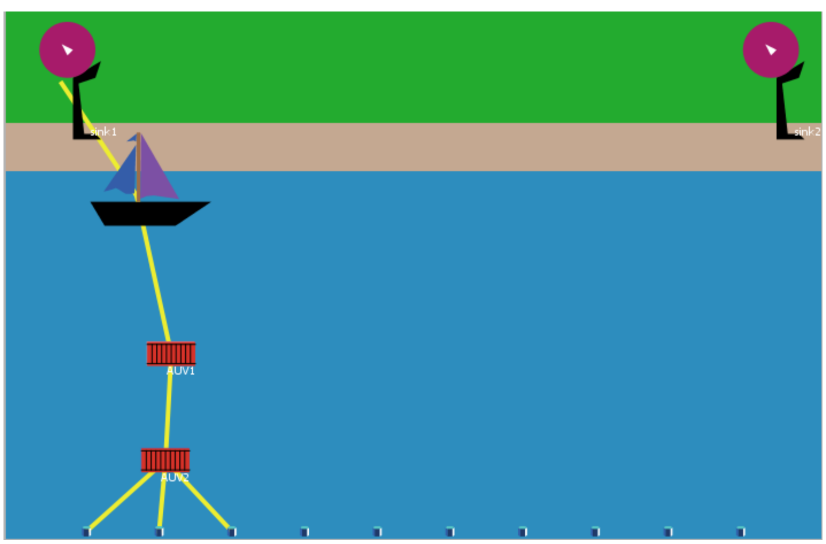

Fig. 5(d) Scenario 2, a moment during simulation, links shown

5.3.3 Scenario 3: Three AUVs and one boat

In the third scenario, two AUVs are present side-by –side in the first layer, with an additional one in stack in the second layer, see Fig. 5(e) and Fig. 5(f). This has been done to increase the amount of communication between the nodes and the AUV, given the fact that speed of AUVs are much slower than boat.

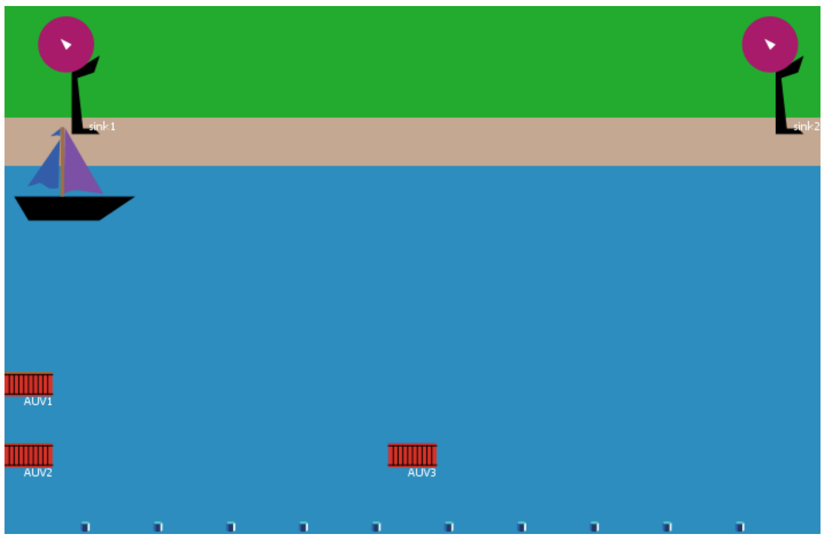

Fig. 5(e) Scenario 3: Three AUVs and one boat

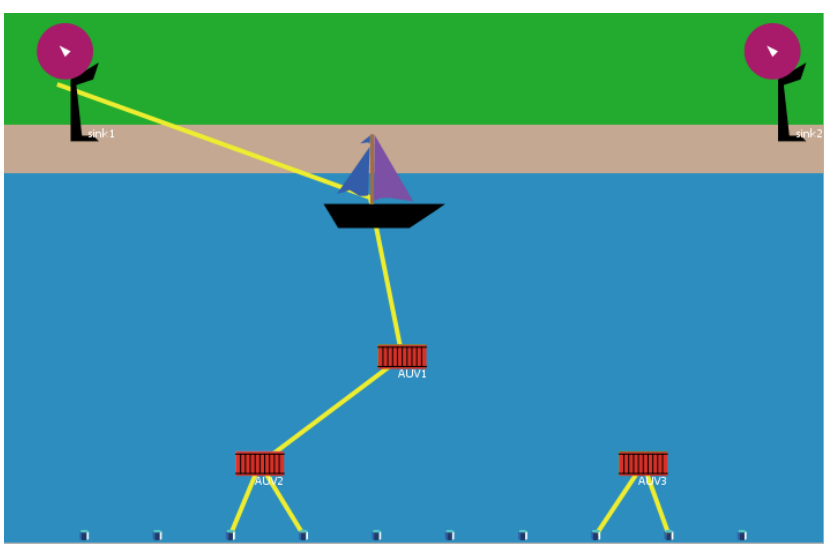

Fig. 5(f) Scenario 3, a moment during simulation, links shown

5.3.4 Scenario 4: Three AUVs and two boats

Scenario 4 is similar in construct to scenario 3, with an additional boat present to increase overall data transfer rate. The second boat is assumed to start its cruise from the opposite end. The idea for this scenario is to increase the amount of communication in the entire system.

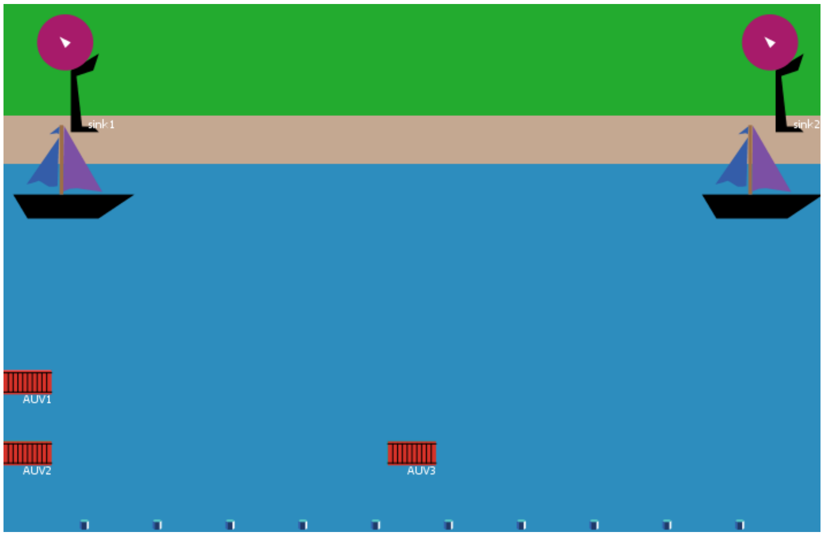

Fig. 5(g) Scenario 4: Three AUVs and two boats

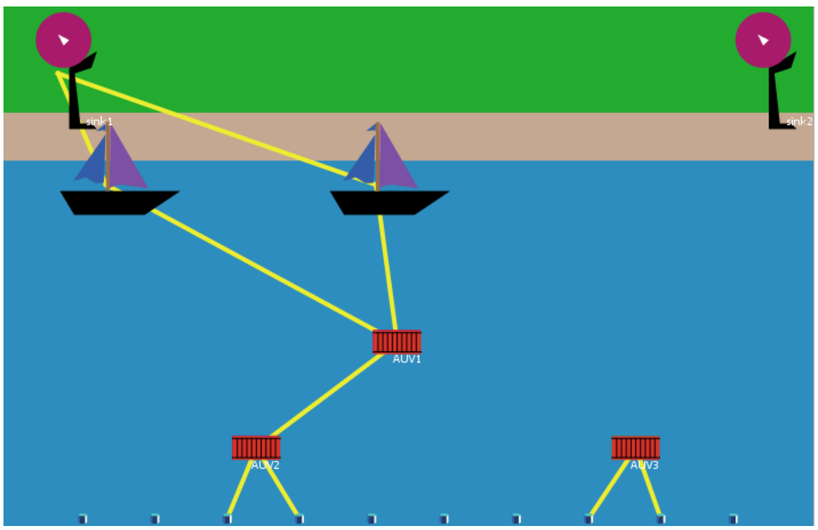

Fig. 5(h) Scenario 4, a moment during simulation, links shown

5.4 Methods of analysis

The 4 different scenarios were compared on the basis of 2 approaches:

- In one approach, the simulation time was kept fixed and the % data received at the sink, the end node, was measured by our simulator for each scenario respectively.
- In another approach, the volume of data received at the sink is kept fixed, and the corresponding time for this data transfer is measured for each scenario.

5.4.1 Constant time analysis

Total data generated by nodes and total data received at sinks are plotted with time. The plots from our simulation for the 4 scenarios for constant time analysis are as follows:

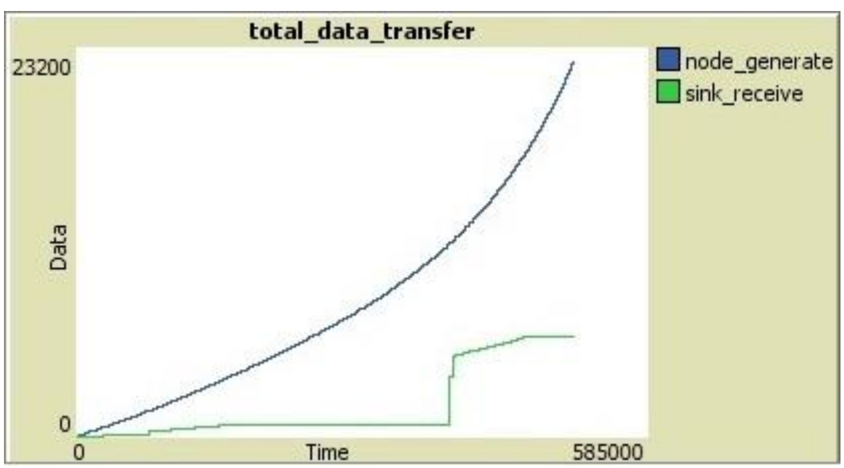

Fig. 5(i) Plot for constant time analysis of scenario 1

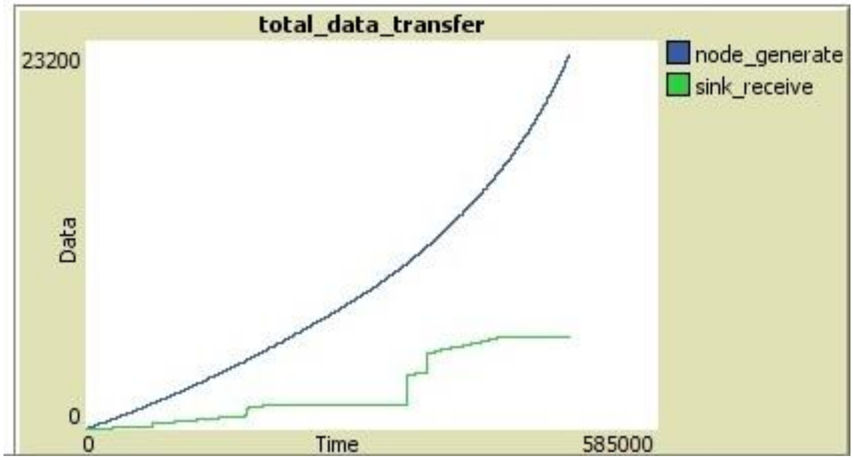

Fig. 5(j) Plot for constant time analysis of scenario 2

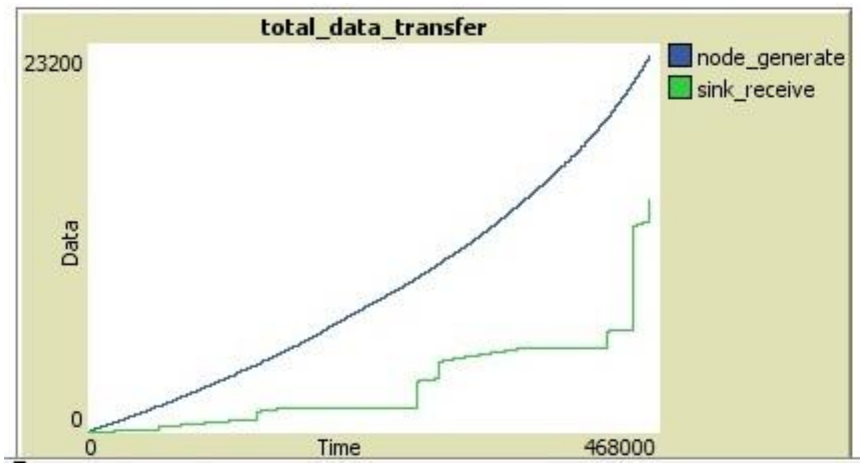

Fig. 5(k) Plot for constant time analysis of scenario 3

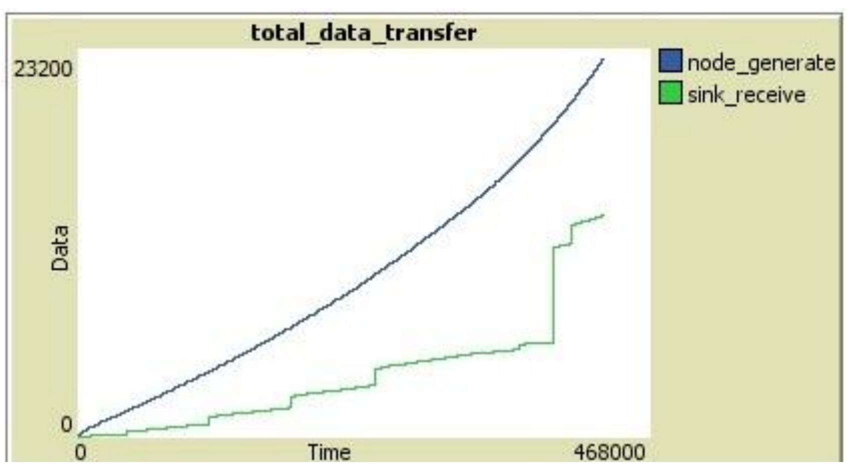

Fig. 5(l) Plot for constant time analysis of scenario 4

In Constant time analysis, for all scenarios, simulation time was considered 8 minutes. Since the simulation time was fixed, amount of total data generated by the nodes was same for all cases. On the basis of different amount of received data, we calculated percentage data transferred for all 4 scenarios as follows:

$$\text{Data Transferred} = \frac{\text{Amount of Data Received at sink node}}{\text{Amount of Data Generated}} \times 100$$

Here, amount of data generated = simulation time (in sec) * number of nodes * data generation rate

= (8 * 60) * 5 * 10 bytes

= 24000 bytes

Scenario 1

Amount of data received at sink node = 5855 bytes

∴ % Data Transferred = **24.4%**

Scenario 2

Amount of data received at sink node = 6380 bytes

∴ % Data Transferred = **26.58%**

Scenario 3

Amount of data received at sink node = 14095 bytes

∴ % Data Transferred = **58.73%**

Scenario 4

Amount of data received at sink node = 14845 bytes

∴ % Data Transferred = **61.85%**

5.4.2 Constant data analysis

Total data generated by nodes and total data received at sinks are plotted with time. The plots from our simulation for the 4 scenarios for constant data analysis are as follows:

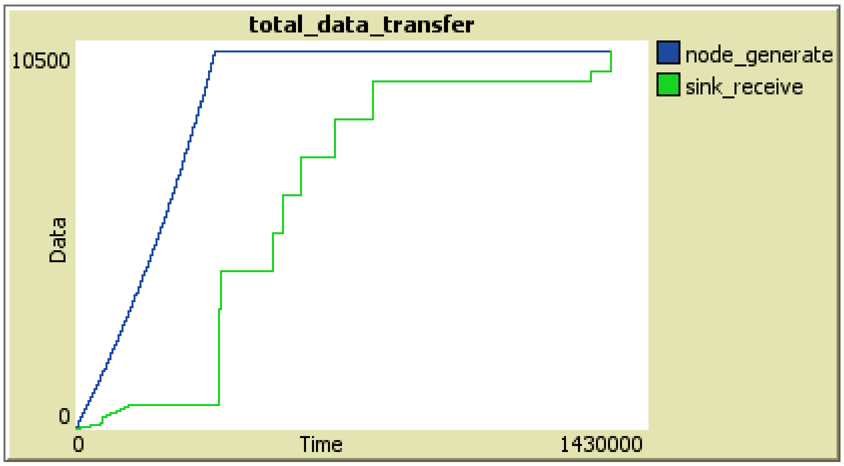

Fig. 5(m) Plot for constant data analysis of scenario 1

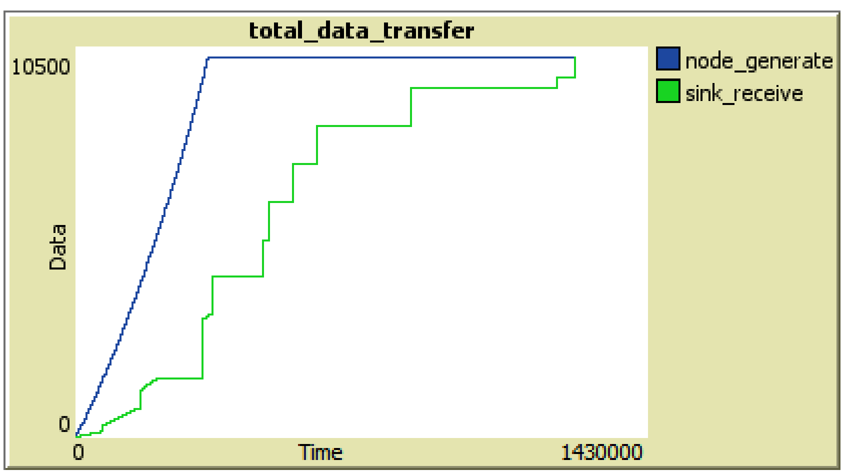

Fig. 5(n) Plot for constant data analysis of scenario 2

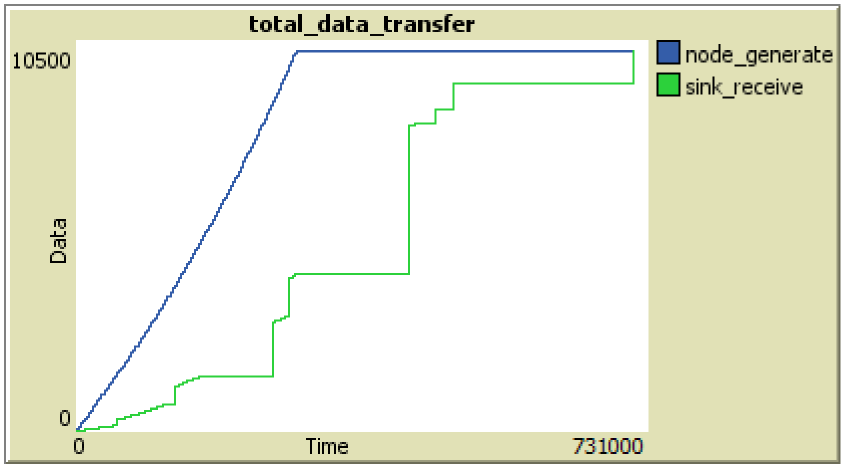

Fig. 5(o) Plot for constant data analysis of scenario 3

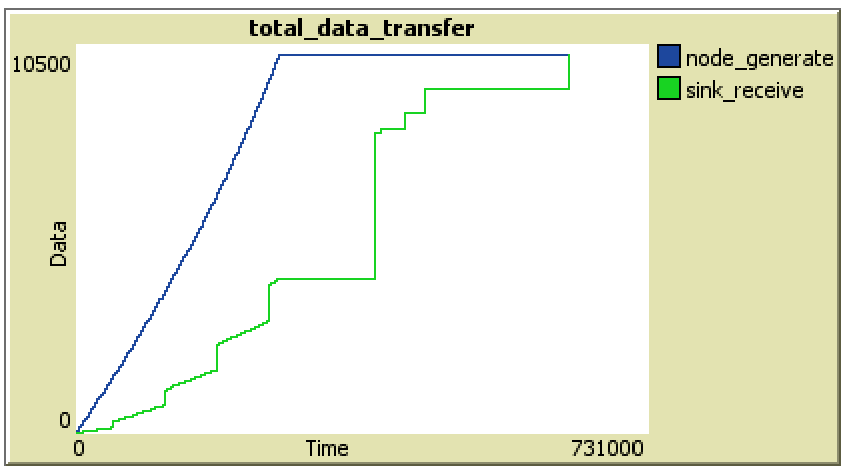

Fig. 5(p) Plot for constant data analysis of scenario 4

In Constant data analysis, for all scenarios, volume of data for generation and transfer was considered 10 kilobytes. Since the data generation was fixed, nodes stopped generating when combined node generation reached precisely 10240 bytes i.e. 10 kilobytes. We therefore measured the time required for data transfer for each scenarios.

$$\text{Time taken for data generation} = \frac{\text{Amount of data generated}}{\text{Number of nodes} * \text{Data generation rate}}$$

$$= \frac{10240}{10 * 5}$$

= 204.8 sec

Scenario 1

Total transfer time = 2166.743 sec = 36.112 minutes

∴ Time taken after generation completion = (2166.743 - 204.8) = **1961.943 sec**

Scenario 2

Total transfer time = 1891.045 sec = 31.517 minutes

∴ Time taken after generation completion = (1891.045 - 204.8) = **1686.245 sec**

Scenario 3

Total transfer time = 1234.04 sec = 20.567 minutes

∴ Time taken after generation completion = (1234.040 - 204.8) = **1029.240 sec**

Scenario 4

Total transfer time = 1240.561 sec = 20.676 minutes

∴ Time taken after generation completion = (1240.561 - 204.8) = **1035.761 sec**

5.5 Locating the optimum point

To locate the optimum point we use both the analysis of constant time and constant data. Using the constant time analysis we find that the amount of data transferred within a set period of 8 minutes between the sink node and the sensor nodes is highest in scenario 3 and in scenario 4.

Then we keep the total amount of data constant at 10 kilobytes and then run the simulation in the constant data analysis to find that the least amount of time required to transfer data between the nodes and sink is in scenario 3 as well followed by scenario 4.

After this we compare the scenarios from the time constant analysis by comparing the increasing of data transfer whenever we move from one scenario to the next by addition of different components to the network.

The following chart Fig. 5(q) shows a comparison of the four scenarios in terms of increase in percentage data transferred.

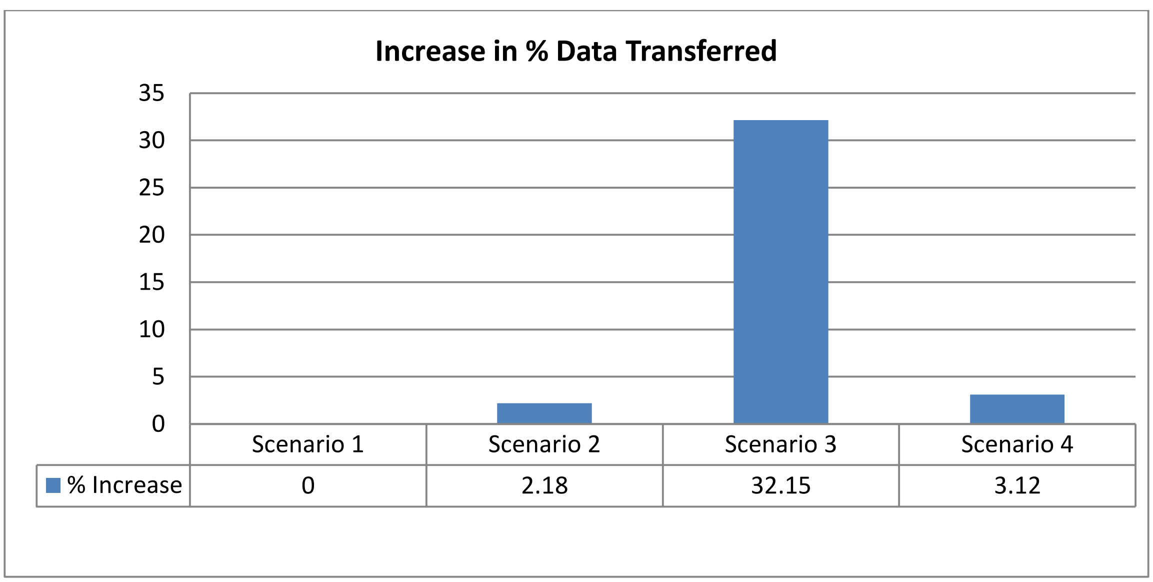

Fig. 5(q) Comparison of four scenarios

Highest increase in percentage data transferred is observed at scenario 3.

Thus we select **scenario 3** of three AUVs and one boat as the optimum point.

CHAPTER

6

Conclusion

Conclusion

DTN is a technology that is very new to the world of networking. Underwater data is an essential part of everyday life of human beings and every year billions of dollars are being spent to recover as much information as possible.

The thesis exploited this vast field of data availability and aimed to achieve the efficient point at which the data transfer would be done at maximum for the money being spent behind the projects.

The project can be implemented in different scenarios and can be modified according to the different requirements of the underwater terrain at different locations.

Furthermore, we tried to reach the optimum point of operation simulating two different sorts of operation to the four scenarios that we selected to understand the project even better.

References

[1] K. Fall. A delay-tolerant network architecture for challenged internets, in: Proceedings of ACM SIGCOMM'03, August 2003.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Computer Networks (Elsevier) Journal, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[3] The Delay Tolerant Networking Research Group, Available from <http://www.dtnrg.org>.

[4] E.M. Sozer, M. Stojanovic, J.G. Proakis, Underwater acoustic networks, IEEE Journal of Oceanic Engineering 25 (1) (2000) 72–83.

[5] I. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. In Elsevier Journal of Ad Hoc Networks, volume 3, pages 257–281, Mar. 2005.

[6] Akyildiz, I. F., Pompili, D., and Melodia, T. Challenges for efficient communication in underwater acoustic sensor networks. ACM Sigbed Review 1, 2 (july 2004).

[7] Heidemann, J., Li, Y., Syed, A., Wills, J., and Ye, W. Underwater sensor networking: Research challenges and potential applications. Tech. Rep. ISI-TR-2005-603, USC/Information Sciences Institute, July 2005.

[8] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," IEEE Communications Surveys & Tutorials, vol. PP, no. 99, 2011, pp. 1–34.

[9] Ccl.northwestern.edu. 2013. *NetLogo Home Page*. [online] Available at: http://ccl.northwestern.edu/netlogo/ [Accessed: 8 Oct 2013].

[10] En.wikipedia.org. 2013. *Delay-tolerant networking*. [online] Available at: http://en.wikipedia.org/wiki/Delay-tolerant_networking [Accessed: 8 Oct 2013].

[11] En.wikipedia.org. 2013. *Underwater acoustic communication*. [online] Available at: http://en.wikipedia.org/wiki/Underwater_acoustic_communication [Accessed: 8 Oct 2013].

[12] Ece.gatech.edu. 2013. *Underwater Acoustic Sensor Networks*. [online] Available at: http://www.ece.gatech.edu/research/labs/bwn/UWASN/ [Accessed: 8 Oct 2013].