

Implementation of Link Stability
Based Routing for Underwater
Network

By
MS. TSU-CHI HUANG

Submitted in Partial Fulfillment of the
Requirement for the Degree of
Master of Science
in Telecommunications Science
Assumption University

March , 2007

Implementation of Link Stability Based Routing for Underwater Network

By

MS. TSU-CHI HUANG



**Submitted in Partial Fulfillment of the
Requirement for the Degree of
Master of Science
in Telecommunications Science
Assumption University**

March , 2007

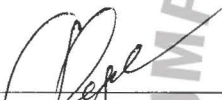
Faculty of Science and Technology

Thesis Approval


Thesis Title: Implementation of Link Stability Based Routing for Underwater Network
By: Tzu-Chi Huang
Thesis Advisor: Dr. Sergey Fedoseev
Academic Year: 3/2006

The Department of Telecommunications Science, Faculty of Science and Technology, Assumption University, Thailand, has approved this final report of the Twelve Credits course, TS 7000 Master Thesis, submitted in partial fulfillment of the requirements for the degree of Master of Science in Telecommunications Science (MS TS).


Approval Committee:




(Dr. Sergey Fedoseev)
Advisor



(Assoc. Prof. Dr. Chavdar M. Hardalov)
Committee




(Asst. Prof. Dr. Dobri A. Batovski)
Committee

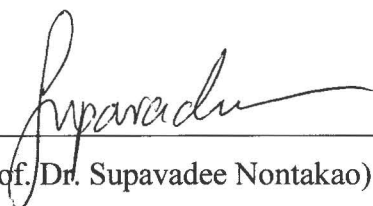


(Prof. Dr. Chidchanok Lursinsap)
Commission of Higher Education,
University Affairs

Faculty Approval:



(Asst. Prof. Dr. Dobri Batovski)
Director, MS TS



(Asst. Prof. Dr. Supavadee Nontakao)
Dean, Faculty of Science and Technology

May 2007

ACKNOWLEDGEMENTS

I would like to thank the instructors from St. Gabriel Telecommunications Research Laboratory, Assumption University, for giving me the opportunity to initiate and complete the work on this thesis.



ABSTRACT

Wireless underwater networks are being considered for a variety of applications. To make these applications viable, there is a need to enable reliable communications among underwater sensors and devices. Wireless underwater communication is a challenging task because most of the common communication means do not work well in water. Since acoustic communication systems provide very low bandwidth and radio waves do not propagate in water, underwater optical wireless communications are being considered.

The main focus of this thesis is to understand the performance of the underwater optical wireless communications. Computational experiments are performed to study a proposed topology control scheme in preserving a full connectivity network while reducing the risk of hot spots causing congestion in the network.

TABLE OF CONTENTS

| | |
|--|------|
| DECLARATION OF ORIGINAL WORK | ii |
| ACKNOWLEDGEMENTS | iii |
| ABSTRACT | iv |
| TABLE OF CONTENTS | v |
| LIST OF TABLES | viii |
| LIST OF FIGURES | ix |
| ACRONYMS | xii |
| CHAPTER 1 INTRODUCTION | 1 |
| CHAPTER 2 OVERVIEW AND BACKGROUND | 4 |
| 2.1 Literature Review | 4 |
| 2.1.1 Wireless Sensor Network (WSN) | 4 |
| 2.1.2 Free Space Optics (FSO) | 6 |
| 2.1.3 Optical Sources and Detectors for Underwater Communications | 7 |
| 2.1.4 Underwater Sensor Network Communication Architecture | 8 |
| 2.1.5 Underwater Sensors | 11 |
| CHAPTER 3 METHODOLOGY | 12 |
| 3.1 Network Platform | 12 |
| 3.2 Node Design | 12 |
| 3.3 Topology Control | 15 |

| | |
|---|----|
| 3.4 Set/Ring Construction Algorithm | 16 |
| 3.5 Overview of the Computational Experiments for Set/Ring Construction Algorithm | 20 |
| 3.6 Software Development | 26 |
| 3.7 Connectivity Matrix | 27 |
| 3.8 Unicast Routing | 28 |
| 3.9 Delay Calculation | 29 |
| CHAPTER 4 NUMERRICAL RESULTS | 30 |
| 4.1 Performance of the Sample Topology of 7 nodes | 31 |
| 4.1.1 Connectivity Matrix | 31 |
| 4.1.2 Delays in Sample Topology of 7 nodes | 34 |
| 4.1.2.1 Delay at Each Node | 34 |
| 4.1.2.2 End-to-End Delay | 36 |
| 4.1.2.3 Average Delay per Hop for Different Hop Number | 38 |
| 4.1.3 Throughputs in Sample Topology of 7 nodes | 40 |
| 4.1.3.1 Edge Throughput | 40 |
| 4.1.3.2 Average Throughput per Hop for Different Hop Number | 41 |
| 4.1.4 Frequency of Edge Overlapping | 42 |
| 4.2 Performance of the Sample Topology of 31 nodes | 43 |
| 4.2.1. Delays in Sample Topology of 31 nodes | 45 |
| 4.2.1.1 Delays at Each Node | 46 |
| 4.2.1.2 Average Delay per Hop for Different Hop Number | 48 |
| 4.2.2 Throughputs in Sample Topology of 31 nodes | 50 |
| 4.2.2.1 Edge Throughput | 50 |

| | |
|--|----|
| 4.2.2.2 Average Throughput per Hop for Different | |
| Hop Number | 51 |
| 4.2.3 Frequency of Edge Overlapping | 53 |
| CHAPTER 5 ANALYSIS AND DISCUSSION | 55 |
| CHAPTER 6 CONCLUSION | 60 |
| REFERENCES | 62 |
| APPENDICES | |
| A.1 Source Code for Set/Ring Constructions Algorithm | 64 |
| A.2 Source Code for Obtaining the Sample Topologies | 67 |
| A.3 Source Code for the Shell Sorting Algorithm | 70 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1.1 Spectral attenuation coefficient of ocean waters [4]. | 2 |
| Figure 2.1 FSO links when no direct link fiber optic network is possible [7]. | 6 |
| Figure 2.2 Architecture for 2D Underwater Sensor Networks. | 9 |
| Figure 2.3 Architecture for 3D Underwater Sensor Networks. | 10 |
| Figure 2.4 Internal architecture of an underwater sensor node. | 11 |
| Figure 3.1 Schematic diagram of the underwater diver network with three optical transmitters and receivers with a buoyant. | 13 |
| Figure 3.2 Schematic diagram of the underwater static network with four optical transmitters and receivers with a buoyant. | 13 |
| Figure 3.3 Schematic diagrams of the multi-section transmitters/receivers. | 14 |
| Figure 3.4 A basic optical node design. | 15 |
| Figure 3.5 A schematic view of a ring construction process. | 16 |
| Figure 3.6 Node degree of a sample configuration of nodes. | 17 |
| Figure 3.7 Creation of strings or rings of interconnected first neighbors around a central node. | 19 |
| Figure 3.8 Connection of the sets of strings or rings to the central node. | 20 |
| Figure 3.9 A sample topology of 7 nodes. | 21 |
| Figure 3.10 A sample topology of 31 nodes. | 22 |
| Figure 3.11 7-node sample topology network connections with degree threshold = 3. | 23 |
| Figure 3.12 7-node sample topology network connections with degree threshold = 4. | 23 |
| Figure 3.13 7-node sample topology network connections without topology control. | 24 |
| Figure 3.14 31-node sample topology network connections with degree threshold = 3. | 24 |
| Figure 3.15 31-node sample topology network connections with degree threshold = 4. | 25 |
| Figure 3.16 31-node sample topology network connections without topology control. | 25 |
| Figure 3.17 The flow chart of the software implementation. | 27 |

| | |
|--|----|
| Figure 3.18 A schematic view of a connectivity matrix. | 28 |
| Figure 3.19 Asynchronous Transmission in timeslot. | 28 |
| Figure 4.1 Delay at Each Node, 7 Nodes, Threshold 3, Ring scheme. | 34 |
| Figure 4.2 Delay at Each Node, 7 Nodes, Threshold 4, Ring scheme. | 34 |
| Figure 4.3 Delay at Each Node, 7 Nodes, Plain scheme. | 34 |
| Figure 4.4 End-to-End Delays, 7 Nodes, Threshold 3, Ring scheme. | 36 |
| Figure 4.5 End-to-End Delays, 7 Nodes, Threshold 4, Ring scheme. | 36 |
| Figure 4.6 End-to-End Delays, 7 Nodes, Plain scheme. | 36 |
| Figure 4.7 Average Delay per Hop for Different Hop Number, 7 Nodes, Threshold 3, Ring scheme. | 38 |
| Figure 4.8 Average Delay per Hop for Different Hop Number, 7 Nodes, Threshold 4, Ring scheme. | 38 |
| Figure 4.9 Average Delay per Hop for Different Hop Number, 7 Nodes, Plain scheme. | 38 |
| Figure 4.10 Edge Throughput, 7 Nodes, Threshold 3, Ring scheme. | 40 |
| Figure 4.11 Edge Throughput, 7 Nodes, Threshold 4, Ring scheme. | 40 |
| Figure 4.12 Edge Throughput, 7 Nodes, Plain scheme. | 40 |
| Figure 4.13 Average Throughput per Hop for Different Hop Number, 7 Nodes, Threshold 3, Ring scheme. | 41 |
| Figure 4.14 Average Throughput per Hop for Different Hop Number, 7 Nodes, Threshold 4, Ring scheme. | 41 |
| Figure 4.15 Average Throughput per Hop for Different Hop Number, 7 Nodes, Plain scheme. | 41 |
| Figure 4.16 Frequency of Edge Overlapping, 7 Nodes, Threshold 3, Ring scheme. | 42 |
| Figure 4.17 Frequency of Edge Overlapping, 7 Nodes, Threshold 4, Ring scheme. | 42 |
| Figure 4.18 Frequency of Edge Overlapping, 7 Nodes, Plain scheme. | 42 |
| Figure 4.19 Delay at Each Node, 31 Nodes, Threshold 3, Ring scheme. | 46 |
| Figure 4.20 Delay at Each Node, 31 Nodes, Threshold 4, Ring scheme. | 46 |
| Figure 4.21 Delay at Each Node, 31 Nodes, Plain scheme. | 46 |
| Figure 4.22 Average Delay per Hop for Different Hop Number, 31 Nodes, Threshold 3, Ring scheme. | 48 |
| Figure 4.23 Average Delay per Hop for Different Hop Number, 31 Nodes, Threshold 4, Ring scheme. | 48 |

| | |
|---|----|
| Figure 4.24 Average Delay per Hop for Different Hop Number, 31 Nodes, Plain scheme. | 48 |
| Figure 4.25 Edge Throughput, 31 Nodes, Threshold 3, Ring scheme. | 50 |
| Figure 4.26 Edge Throughput, 31 Nodes, Threshold 4, Ring scheme. | 50 |
| Figure 4.27 Edge Throughput, 31 Nodes, Plain scheme. | 50 |
| Figure 4.28 Average Throughput per Hop for Different Hop Number, 31 Nodes, Threshold 3, Ring scheme. | 51 |
| Figure 4.29 Average Throughput per Hop for Different Hop Number, 31 Nodes, Threshold 4, Ring scheme. | 51 |
| Figure 4.30 Average Throughput per Hop for Different Hop Number, 31 Nodes, Plain scheme. | 51 |
| Figure 4.31 Frequency of Edge Overlapping, 31 Nodes, Threshold 3, Ring scheme. | 53 |
| Figure 4.32 Frequency of Edge Overlapping, 31 Nodes, Threshold 4, Ring scheme. | 53 |
| Figure 4.33 Frequency of Edge Overlapping, 31 Nodes, Plain scheme | 53 |
| Figure 5.1 The Average Delay per Hop for Different Hop Number of the Sample 31-node Topology. | 55 |
| Figure 5.2 The Average Throughput per Hop for Different Hop Number of the Sample 31-node Topology. | 57 |
| Figure 5.3 Average Delay per Hop Number for Different Lambda Value. | 58 |

ACRONYMS

| | |
|-------|---|
| 2D | Two-dimensional |
| 3D | Three-dimensional |
| CN | Central Node |
| F | Frequency |
| FIFO | First-In First-Out |
| IEEE | Institute of Electrical and Electronics Engineers |
| kHz | Kilo Hertz |
| MANET | Mobile Ad-hoc Network |
| nm | Nano Meter |
| PE | Processing Equipment |
| QoS | Quality of Service |



CHAPTER ONE

INTRODUCTION

Water domain takes up 70% of the earth space. Unlike terrestrial communication where numerous communication means are accessible, underwater communications to date are still left far behind. Collaboration of underwater sensor nodes enables the various applications of oceanographic data collections, pollution monitoring, offshore exploration, disaster prevention and tactical surveillance [1]. To make uses of these applications in underwater domains, there comes the need of underwater communication among these devices.

Wireless underwater communication is a challenging task. Most commonly used communication means which are well established for communications in air cannot be applied in water. Despite the greater success of radio transmission in terrestrial domains, radio waves do not propagate in sea water. At present, acoustic communication systems are the main underwater communication means due to the exceptional performance of sound propagation in water [2]. Acoustic systems are capable of handling communication range up to 100 meters; however, its maximum data transmission rate cannot exceed 10 kilobits per second (kbps).

Despite the fact that underwater (u/w) wireless optical communication systems can have even shorter communication ranges as a result of greater attenuation of light propagating through water, these wireless optical communication systems may provide higher bandwidth (up to several hundred kbps) communications as well as

covertness. One way to deal with the severe spectral attenuation in water is to use the blue-green LED [3]. As seen in Figure 1-1, the blue-green region of the visible spectrum provides the least attenuation.

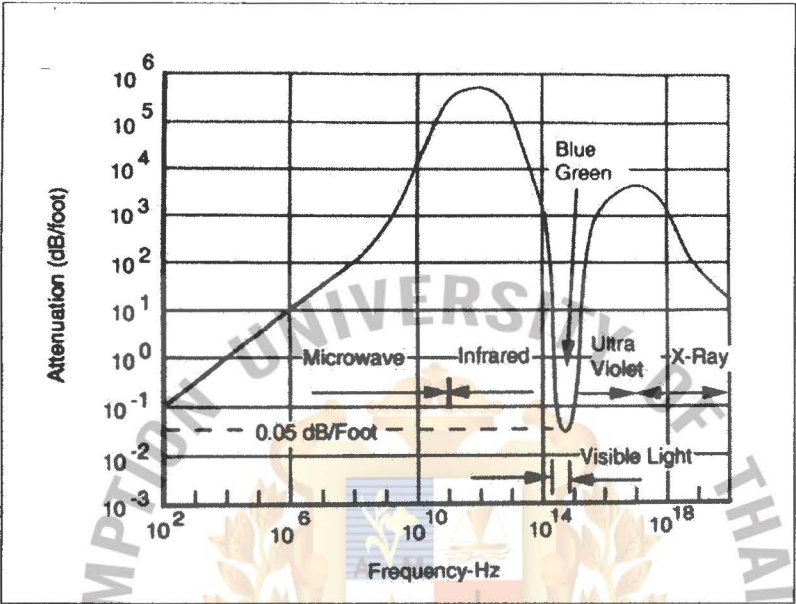


Figure 1.1 Spectral attenuation coefficient of ocean waters [4].

The focus of this work is on the performance of the underwater optical wireless communications. Our particular focus is on the construction of topology control technique to preserve a full connectivity network and at the same time to reduce the congestion at any particular nodes. Instead of using the standard hop count scheme, emphasis is placed on the use of a reasonable node degree for path selection in order to avoid hot spots in any particular fraction of the network.

Due to the unavailability of the statistical regularity of randomly distributed underwater networks, basic simulation studies do not always reveal certain characteristics of the influence of the topology control in the routing process. Therefore, the simplified assumptions are used for the initiation of the computational

experiments with the emphasis on the topology control and its influences in the routing process. A semi-analytical model and its implementation in software for testing arbitrary network configurations is developed and included as part of the thesis.



CHAPTER TWO

OVERVIEW AND BACKGROUND

2.1 LITERATURE REVIEW

2.1.1 Wireless Sensor Network (WSN)

Sensors nowadays can be obtained at a very low price. These sensors are tiny and in some cases are taking the form of dust. However, they may be capable to provide storage and even communication capabilities.

Sensors are being applied in many circumstances. In the health-care industry, sensors allow continuous monitoring of life-critical information. In the food industry, biosensor technology applied to quality control can help prevent rejected products from being shipped out; therefore enhancing consumer satisfaction levels. In agriculture settings, sensors can help determine the conditions of soil and moisture level; they can also detect other bio-related components. Sensors are also widely used for environmental and weather information gathering. They enable us to make preparations in times of bad weather and natural disaster. In the underwater world, sensors allow continuous monitoring of temperature, salinity, acidity and specific chemicals.

A wireless sensor network is one form of an ad hoc wireless network. Sensors are wirelessly connected and they, at appropriate times, relay information back to

some selected nodes. These selected nodes then perform computations based on the collected data (a process commonly known as *data fusion*) to derive an ultimate statistic (that reflects an assessment of the environment and tactical conditions) to allow critical decisions to be made [5].

In [5], several unique features can be identified with wireless sensor networks (WSN):

- (1) **Inherent distribution:** The sensors are widely distributed all over a physical space, and are primarily connected wirelessly.
- (2) **Dynamic availability of data source:** The set of available sensors changes over time as a result of mobility, addition or loss of sensors.
- (3) **Constrained application quality of service demands:** Minimum quality of service (QoS) is required for sensor network; however, this level must be maintained over an extended period of time. There may be many ways to achieve the QoS. For instance, various sensors may monitor over the data or services that meet the applications' QoS requirements. However, the set of QoS requirements can change over time, as the state of the given applications and/or the set of available sensors change.
- (4) **Resource limitations:** Both network bandwidth and sensor energy are constrained. This is especially true when considering battery-powered sensors and wireless networks.
- (5) **Cooperative applications:** Sensor network applications share available resources (such as sensor energy and channel bandwidth) and either cooperate to achieve a single goal, or, at the very least, do not compete for these limited resources.

2.1.2 Free Space Optics (FSO)

Free Space Optics (FSO) communications refer to the transmission of modulated visible or infrared (IR) beams through the atmosphere to obtain optical communications [6]. FSO is also referred as Free Space Photonics (FSP) or Optical Wireless Communication. Similar to Fiber Optical Communication, Free Space Optics (FSO) uses lasers or light emitting diodes, LEDs, to transmit data instead of enclosing the data stream in a glass fiber; transmission is carried via the free space.

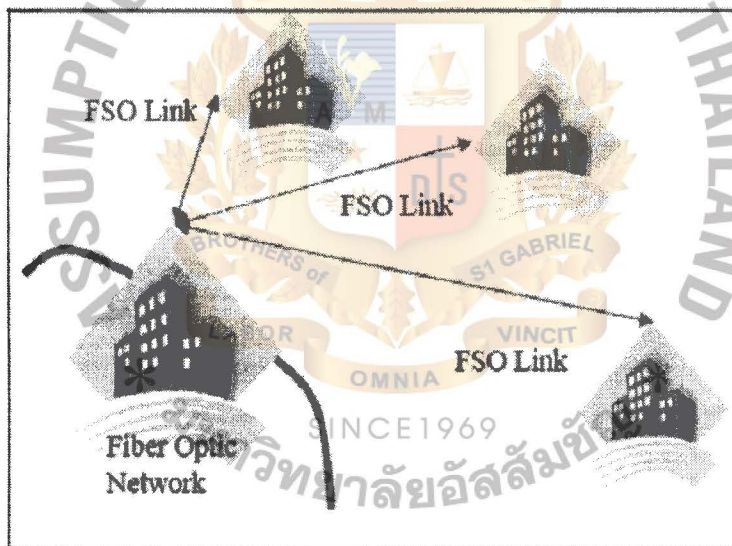


Figure 2.1 FSO links when no direct link fiber optic network is possible [7].

FSO provides several unique advantages. FSO avoids interferences with existing RF communication infrastructures [8]. FSO can be cheaply deployed since government licensing is not required. FSO is not susceptible to “jamming” attacks, and provides a convenient bridge between the wireless sensor network and the optical fiber network. Most importantly, FSO networks enable high bandwidth

transmissions, which make it possible to support multimedia sensor networks [9].

The main limitation of FSO is the requirement of a direct line-of-sight path between the sender and the receiver. Another main disadvantage associated with FSO is the uncontrollable transmission medium. The effects of weather conditions, atmospheric distortions, scintillation, and attenuation can only be minimized or compensated by other sophisticated optical transmission equipments. Applying FOS above the water and under the water is very similar. The major difference is the wavelength of operation.

2.1.3 Optical Sources and Detectors for Underwater Communications

Newly developed LEDs emit substantial light and can be obtained inexpensively. LEDs now can emit up to several watts of power and the angle divergence of the light can increase up to several tens of degrees. If multiple of these LEDs are placed in an array, the resulting output power can be very substantial.

Advance in photodiodes allows pulses as narrow as several nanoseconds to be responded, such as avalanche photodiodes. Balance between speed and sensitivity is needed for photodiode. A typical arrangement is to use transimpedance amplifier in amplifying the current from the photodiode. Using this arrangement, the data transmission rates can be as high as several hundred kbps [2].

According to [10], it is not an easy task to find out the optimal wavelength for use in underwater communication systems since it can be affected by many factors. However, it is clear that light absorption in water increases towards the red and

infrared part of the spectrum. Minimal absorption occurs for wavelength between 400nm and 500nm in the blue-green portion of spectrum.

2.1.4 Underwater Sensor Network Communication Architecture

Two possible underwater sensor network communication architectures can be considered as follows:

(1) Two-dimensional underwater sensor network for ocean bottom monitoring.

Figure 2.2 shows the architecture for two-dimensional underwater networks. A group of sensor nodes are located on the bottom of the ocean for data sensing. By means of optical links, these underwater sensor nodes are interconnected and relaying data back to a surface station. However, considering the long distance between these ocean bottom sensor nodes and the surface station, it might not be practical having all the sensor nodes send the sensed data back to the surface station. One or more underwater sinks (uw-sinks) may be included taking the responsibility of collecting data from all sensor nodes and relaying the collected data from the ocean bottom network to a surface station. Sensor nodes can be connected to uw-sinks via direct links or through multi-hop paths. In the former case, each sensor directly sends the sensed data to the selected uw-sink. This is rather simple from the network point of view. However, this might not be energy efficient since the uw-sink may be located far away from the sensor nodes. In the case of multi-hop paths, sensor nodes relay sensed data to intermediate sensor nodes for relaying to the selected uw-sink. This increases the routing complexity in such a network. However, energy saving and increased network performance can be expected.

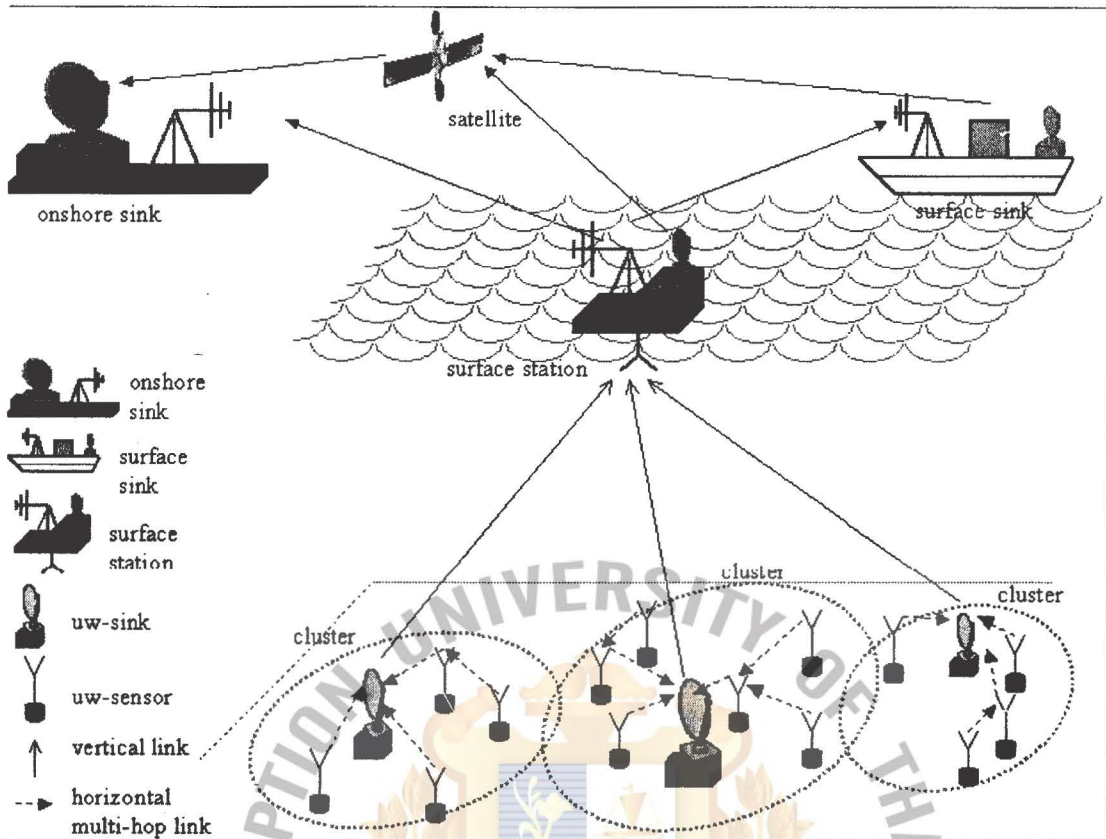


Figure 2.2 Architecture for 2D Underwater Sensor Networks

(2) *Three-dimensional underwater sensor network for ocean column monitoring.*

Figure 2.3 shows the architecture for three-mensional underwater networks. Three-dimensional underwater networks are used to observe and monitor phenomenon that cannot be adequately observed by means of ocean bottom sensor nodes such as collection of oceanographic data in different ocean depths. In three-dimensional underwater sensor networks, sensor nodes are float at different depths of the ocean in order to observe the given phenomenon from different depths. One possible way to control the depth of the sensor nodes is to wire the sensor nodes to a surface buoys. By adjusting the length of the wire, the sensor nodes attached to this wire can be placed at different depth of the ocean. However, this is not a practical way since multiple floating buoys may obstruct ships navigating on the ocean surface or in the military settings these sensor nodes can be easily detected by enemies. Another way

dealing with the depth of location is to anchor these sensor nodes to ocean bottom and equip these sensor nodes with a floating buoy that can be inflated by a pump. The buoys push these sensor nodes to the ocean surface. The depth of the sensor nodes can be adjusted using the wire that connects the sensor nodes to the anchor electronically by an engine that resides on these sensor nodes. This 3D structure needs to take special concerns in sensing coverage and communication coverage. In order to obtain samplings of the desired phenomenon at all depths, sensor nodes need collaboratively regulate their depths to achieve full column coverage according to their sensing ranges. Since there is no notion of uw-sinks in 3D underwater sensor networks, sensor nodes should be able to relay their sensed data back to the surface station by means of multi-hop paths. Therefore, it is important that the network topology is always connected. That is at least one path is always available from every sensor to the surface station.

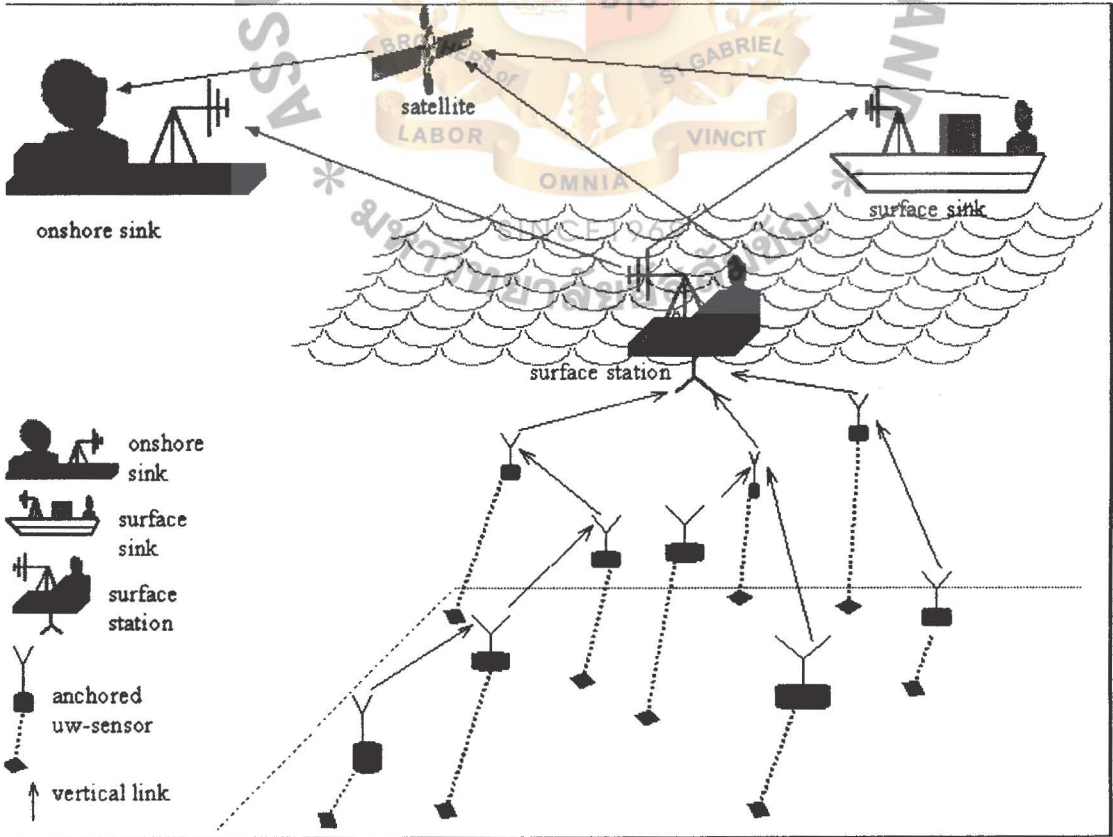


Figure 2.3 Architecture for 3D Underwater Sensor Networks.

2.1.5 Underwater Sensors

Figure 2.3 shows a typical internal structure of an underwater sensor to use with optical communication. It consists of CPU, which is the main controller in the unit. The CPU is interfaced with an oceanographic instrument or so called sensor through a sensor interface circuitry. The controller receives sensed data from the sensor, stores the received data in the onboard memory, processes the data and then relays the processed data via the optical transmitter to other network devices [11].

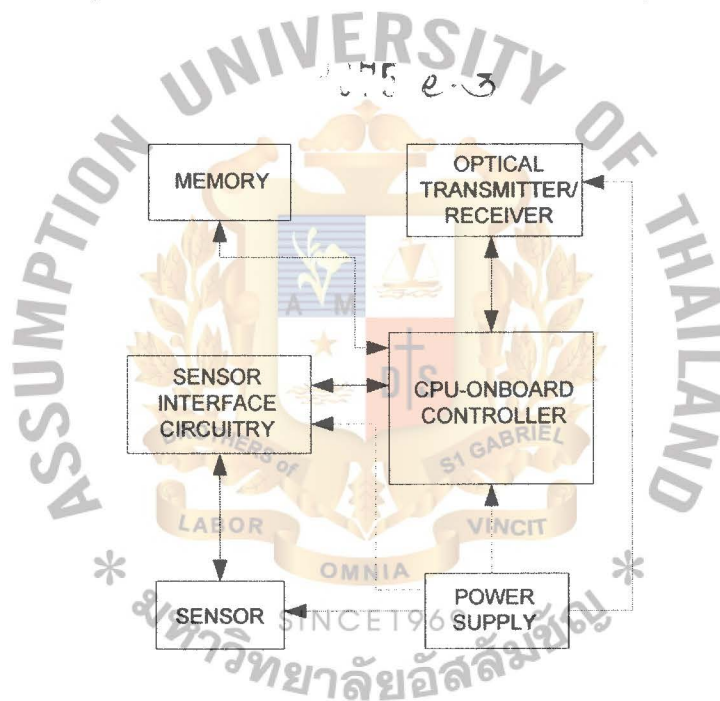


Figure 2.4 Internal architecture of an underwater sensor node.

CHAPTER THREE

METHODOLOGY

3.1 Network Platform

The underwater network in interest is a network consists of numbers of nodes that are connected by means of optical links. The blue-green semiconductor is assumed for the light source to be used in the optical transmissions.

3.2 Node Design

Due to the line-of-sight requirement and limited angle aperture of spectrum beam of wireless optical communications, nodes in the underwater networks are vital to have multiple optical transmitters/receivers in order to transmit and receive data from their first neighbor nodes simultaneously. In the computational experiments, every node contains 3 or 4 numbers of optical transmitter/receivers. This network structure is considered as a quasi-two dimensional network where a node is capable to cover different transmission directions in the same depth of the water. Due to the fact that the pressure of water increases towards the depth of water, it is difficult for a node to transmit the light towards the bottom of the ocean where the multi-path dispersion can be very severe. Figure 3.1 and 3.2 show the schematic block diagrams of the underwater networks. Figure 3.3 shows the schematic block diagram of the multi-section optical transmitters/receivers.

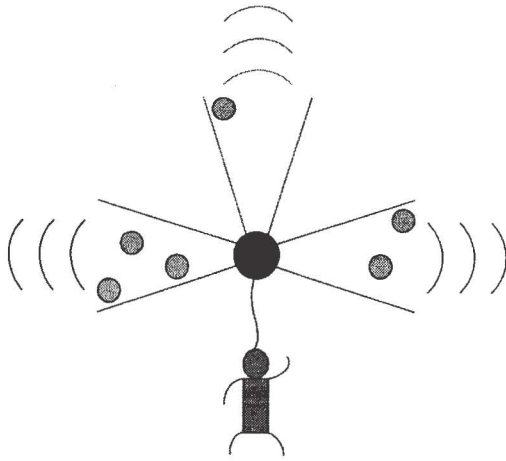


Figure 3.1 Schematic diagram of the underwater diver network with three optical transmitters and receivers with a buoyant.

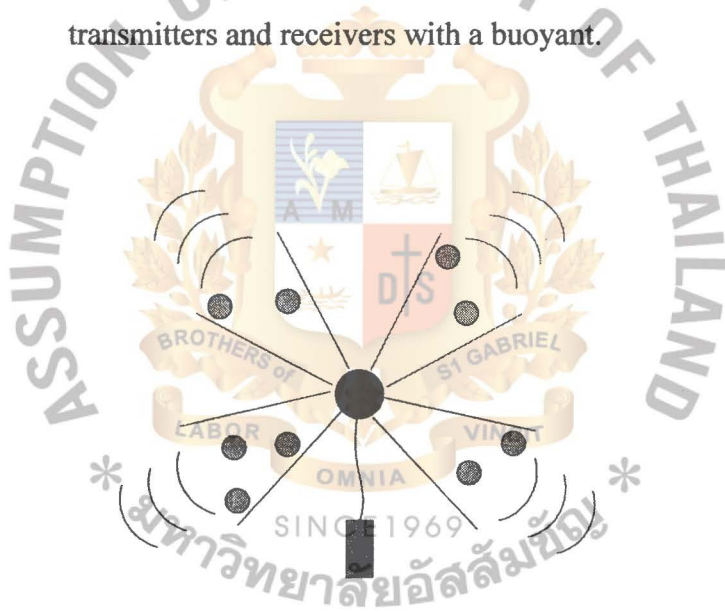


Figure 3.2 Schematic diagram of the underwater static network with four optical transmitters and receivers with a buoyant.

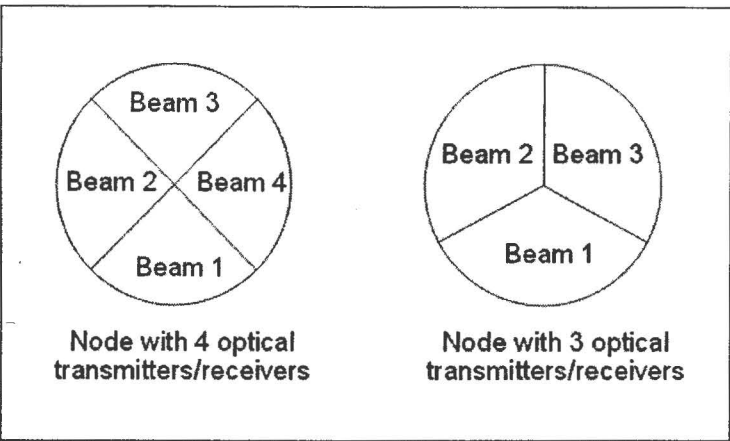


Figure 3.3 Schematic diagram of the multi-section transmitters/receivers.

The nodes in the underwater network must be capable of asynchronously receiving and transmitting information in the form of packets of variable length among multiple input/output optical ports. It is assumed in the computational experiments that only one output port is activated at a time. The data flow is controlled by a FIFO buffer, using the M/M/1 queuing model for the computational experiments. The total traffic arrival rate (λ_{Total}) per one active node is

$$\lambda_{\text{Total}} = \lambda_1 + \lambda_2 + \dots + \lambda_N \tag{3.1}$$

and the mean service Rate, μ , is

$$\mu = R_1 + R_2 + \dots + R_N \tag{3.2}$$

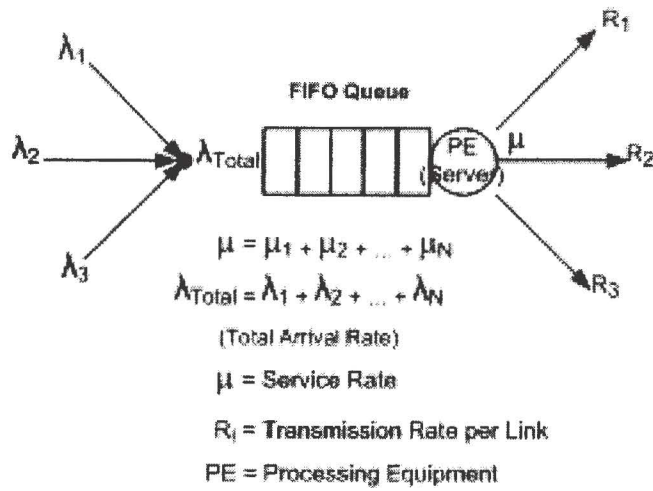


Figure 3.4 A basic optical node design.

3.3 Topology Control

Topology control is employed to dynamically reconfigure node connectivity by means of minimizing the number of input ports and at the same time minimizing the number of output ports. Topology control in this manner helps mitigate the effects of dispersion and attenuation in the underwater wireless optical communications. The set/ring construction algorithm is proposed to carry out the topology control scheme and implemented in the computational experiments. To evaluate the effects of topology control in the routing process, computational experiments are performed for network configurations with topology control and without topology control.

Due to the technological constraints in underwater optical communications, the underwater optical communication ranges are limited to a very short distance. As a result, such a communication system should include only a small number of nodes that are located in close proximity within the transmission range of each other. In

additional to this, all nodes in this underwater network are equipped with 3 or 4 optical sources that are aimed to different directions for transmission. Therefore, it is sufficient to have a flat topology for such an underwater network.

3.4 Set/Ring Construction Algorithm

From an algorithm point of view, the process of ring construction is a topology control technique. The connectivity matrix is an input to the algorithm and the ring connectivity matrix is an output of the algorithm. By means of topology control schemes being performed on the input connectivity matrix, the resultant ring connectivity matrix is the reconfigured node connectivity. Figure 3.5 shows the schematic view of a ring construction process.



Figure 3.5 A schematic view of a ring construction process.

In the proposed ring construction algorithm, the network is first split into independent rings or sets of strings where the exchange of node-degree information among first neighbors allows one to activate a distributed algorithm for the selection of a central node. The topology control scheme being proposed consists of four stages as follows:

a) *Construction of a First-neighbor Routing Table*

The information related to a neighbor node's degree of connectivity is available among its first neighbors. The degree of connectivity equals to the number of first neighbors of a node. The degree of connectivity is also referred as node degree. The neighboring table is then created on the basis of the node degree information available. Figure 3.6 shows the node degree for the sample configuration of nodes.

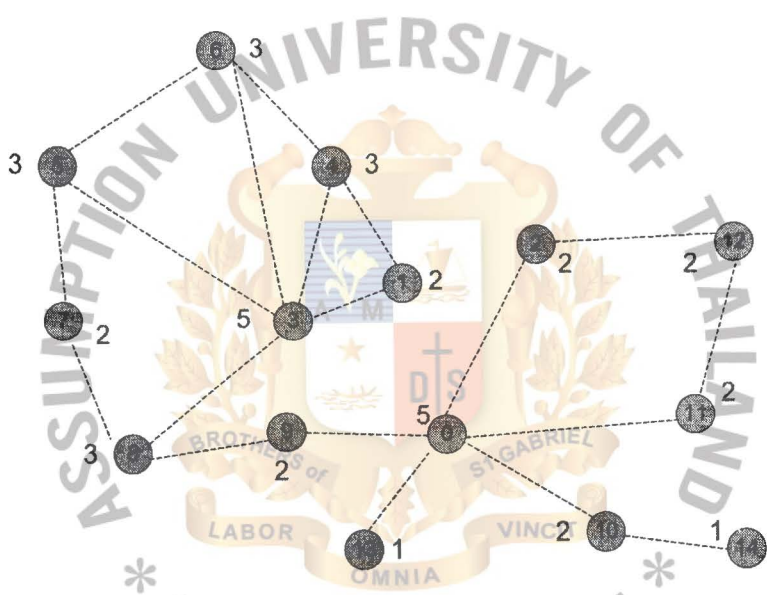


Figure 3.6 Node degree of a sample configuration of nodes.

| Node | Node Degree | Node | Neighbor Node |
|------|-------------|------|---------------|
| 0 | 5 | 0 | 2,9,10,11,13 |
| 1 | 2 | 1 | 3,4 |
| 2 | 2 | 2 | 0,12 |
| 3 | 5 | 3 | 1,4,5,6,8 |
| 4 | 3 | 4 | 1,6 |
| 5 | 3 | 5 | 3,6,7 |
| 6 | 3 | 6 | 3,4,5 |
| 7 | 2 | 7 | 5,8 |
| 8 | 3 | 8 | 3,7,9 |
| 9 | 2 | 9 | 0,8 |
| 10 | 2 | 10 | 0,14 |
| 11 | 2 | 11 | 0,12 |
| 12 | 2 | 12 | 2,11 |
| 13 | 1 | 13 | 0 |
| 14 | 1 | 14 | 10 |

Table 3.1 Sample numerical values of node degree.

b) Selection of a Central Node Based on the Highest Node Degree.

With the available information of the node degree of the first neighbor nodes, one central node is to be elected on the basis of its highest node degree. However, to avoid any overlapping rings or sets of strings, a set/ring member node that already belongs to any sets of strings or rings is not elected as a central node.

c) Creation of Sets of Strings or Rings of Interconnected First Neighbors around a Central Node

For the sets of string or rings around a central node, one arbitrary set/ring member node is initially selected and more interconnected set/ring member nodes are being chosen consecutively to the left (counterclockwise) and to the right (clockwise) until the string of the given set/ring cannot be expanded any further. If the right-most node is interconnected with the left-most node, then a *ring* is created. On the other hand, if the right-most node is not interconnected with the right-most node, then a *set* is created.

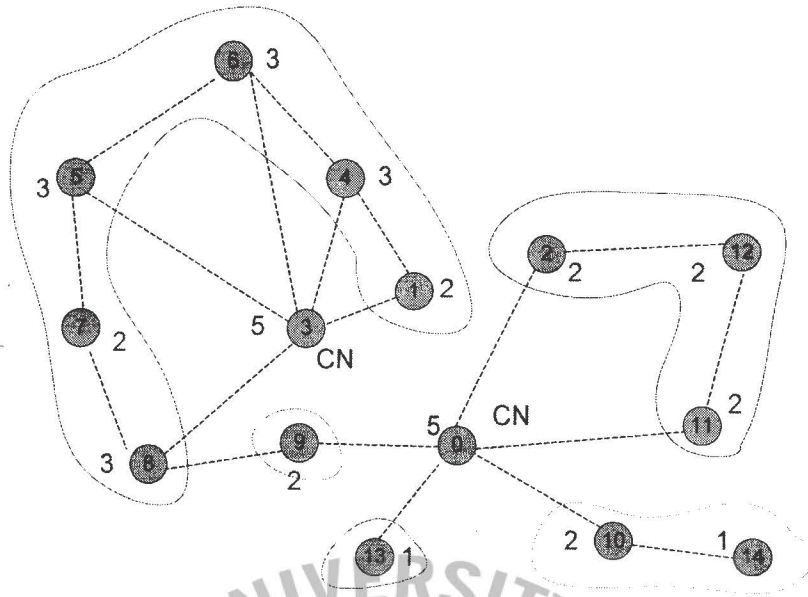


Figure 3.7 Creation of strings or rings of interconnected first neighbors around a central node.

d) Connect the Sets of Strings or Rings to the Central Node.

The central node connects to the sets of strings or rings on a uniformly distributed fashion. This is to avoid unwanted connections of the central node to the sets of strings or rings in close proximity to each other on any part of the sets of strings or rings. The number of connections to the sets of strings or rings from the central node is constrained to the optical sources available per node. For our computational experiments, the optical sources are constrained to node degree thresholds of 3 or 4 connections. Figure 3.8 shows a resulted sample of ring construction.

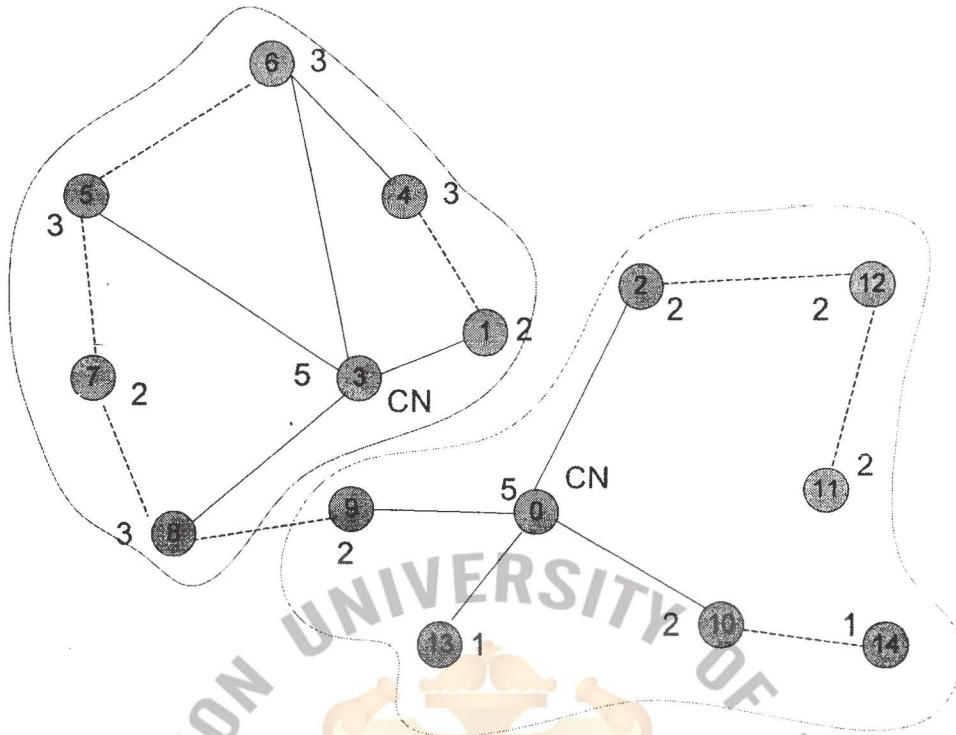


Figure 3.8 Connection of the sets of strings or rings to the central node.

3.5 Overview of the Computational Experiments for Set/Ring Construction Algorithm

To evaluate the performance of the set/ring construction algorithm, computational experiments with different size of networks and different configurations of connected nodes are considered. The evaluation is specifically chosen to study the performance of the sample topology that demonstrates the worst case. The worst case topology in concern is one that has only one node in the center while there are numerous nodes surround this node for relaying data using the shortest path.

Two sample topologies depicting the aforementioned worst case are considered. One is with 7 nodes, and another one is with 31 nodes. The 7-node network is referred

as a demo network while the 31-node network is an expansion of the 7-node demo network to demonstrate the underwater optical wireless network.

Figure 3.9 shows the configuration of the 7-node demo network. This sample topology is a network consisting of 7 nodes. There is only one node in the center with 6 other nodes surround this center node forming a hexagon-like topology.

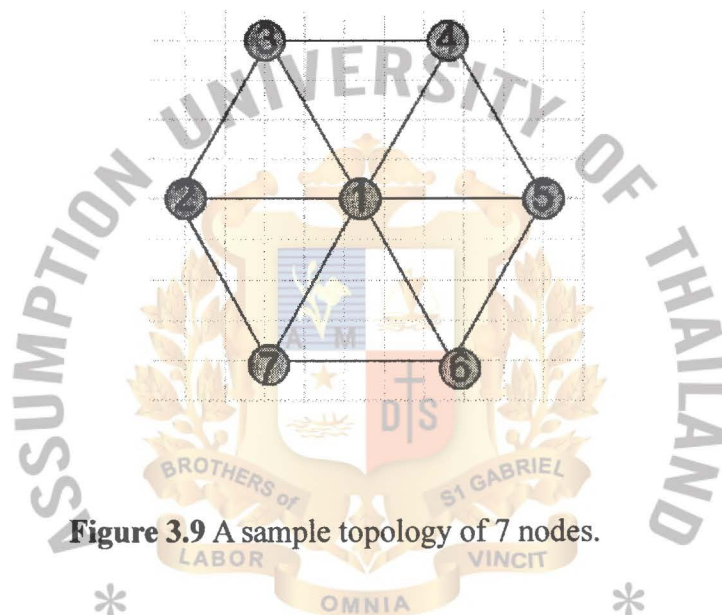


Figure 3.9 A sample topology of 7 nodes.

Figure 3.10 shows the configuration of the 31-node network. This sample topology is a network consisting of 31 nodes. This network is composed of 7 7-node demo network.

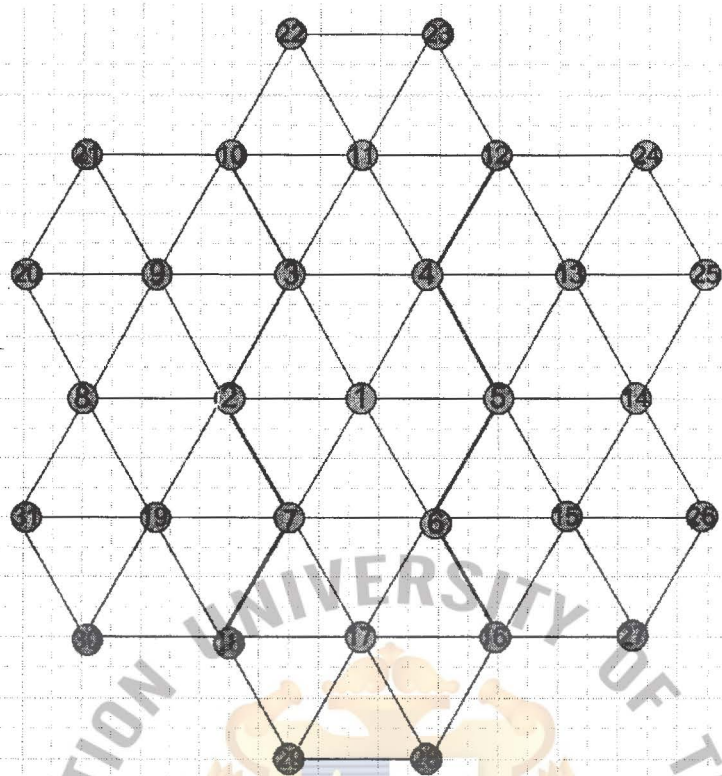


Figure 3.10 A sample topology of 31 nodes.

To evaluate the performance of the ring construction algorithm, 3 scenarios are studied for both sample topologies:

- Scenario 1: Ring Construction Topology Control with Degree Threshold = 3
- Scenario 2: Ring Construction Topology Control with Degree Threshold = 4
- Scenario 3: No topology control implemented

Figure 3.11 shows the possible resulting node connections of the 7-node sample topology by means of Scenario 1.

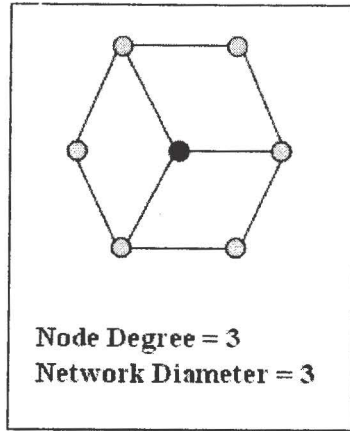


Figure 3.11 7-node sample topology network connections with degree threshold = 3.

Figure 3.12 shows the possible resulting node connections of the 7-node sample topology by means of Scenario 2.

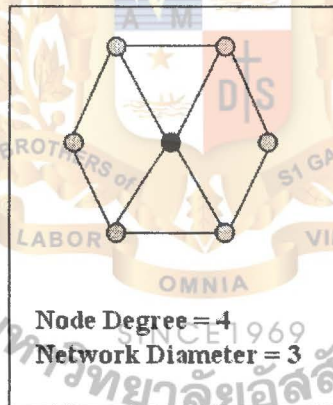


Figure 3.12 7-node sample topology network connections with degree threshold = 4.

Figure 3.13 shows the node connections of the 7-node sample topology by means of Scenario 3. This connectivity indeed represents the sample topology in plain scheme.

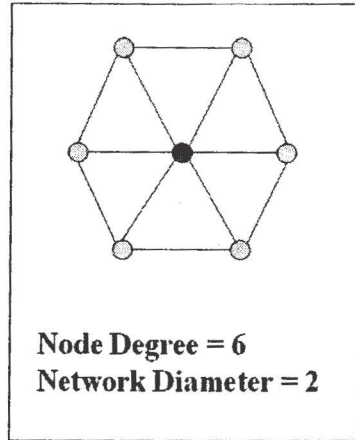


Figure 3.13 7-node sample topology network connections without topology control.

Figure 3.14 shows the possible resulting node connections of the 31-node sample topology by means of Scenario 1.

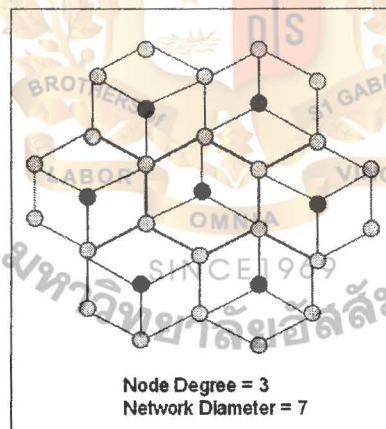


Figure 3.14 31-node sample topology network connections with degree threshold = 3.

Figure 3.15 shows the possible resulting node connections of the 31-node sample topology by means of Scenario 2.

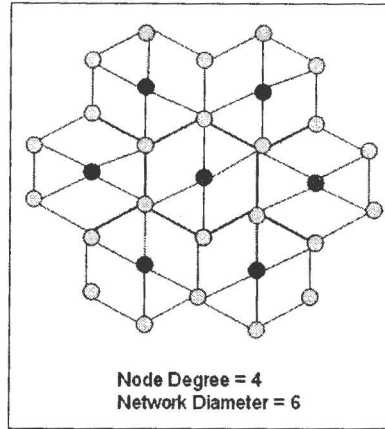


Figure 3.15 31-node sample topology network connections with degree threshold = 4.

Figure 3.16 shows the node connections of the 31-node sample topology by means of Scenario 3. This connectivity indeed represents the sample topology in plain scheme.

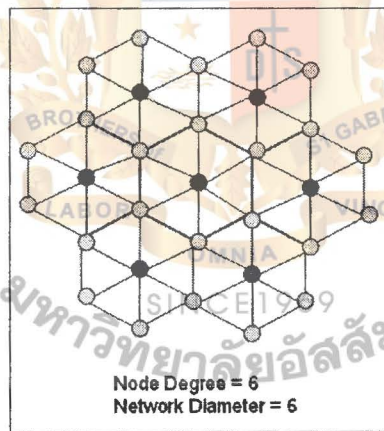


Figure 3.16 31-node sample topology network connections without topology control.

Therefore, total 6 cases of computational experiments are carried out.

Case 1: 7 Nodes, Set/Ring Construction Topology Control with Degree

Threshold = 3

Case 2: 7 Nodes, Set/Ring Construction Topology Control with Degree

Threshold = 4

Case 3: 7 Nodes, No Topology Control

Case 4: 31 Nodes, Set/Ring Construction Topology Control with
Degree Threshold =3

Case 5: 31 Nodes, Set/Ring Construction Topology Control with
Degree Threshold =4

Case 6: 31 Nodes, No Topology Control

3.6 Software Development

The software that is used for the computational experiments is written in Visual C++. The source code of the computation of the Set/Ring construction algorithm is included in Appendix A.1. The source code of the connectivity matrix for both 7-node and 31-node sample topologies is included in Appendix A.2. The source code of the sorting of the nodes according to their node degree to select the central nodes is included in Appendix A.3. Figure 3.17 shows the flow chart of the software implementation. When the software is executed, connectivity matrix for the input sample topology is first initialized. The software then checks if topology control is required. If topology control is implemented, the set/ring construction algorithm is run. The reconfigured connectivity matrix is then formed. In the case that topology control is not implemented, the connectivity matrix remains unchanged. Afterwards, an arbitrary shortest path for unicast routing is selected for transmission. Once transmission completed, the QoS parameters of latency and throughput are calculated for the transmissions and the results are stored in the Hard Disk.

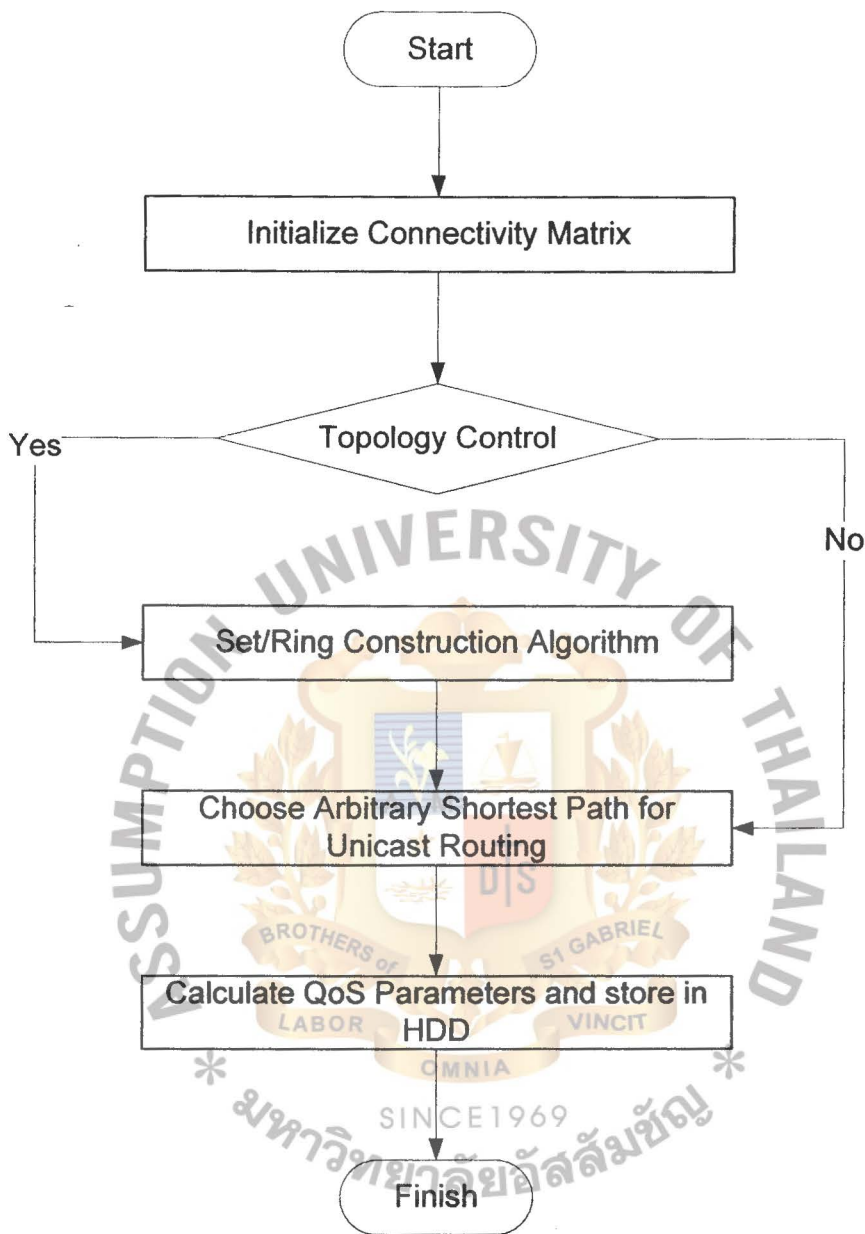


Figure 3.17 The flow chart of the software implementation.

3.7 Connectivity Matrix

The creation of connectivity matrix in the computational experiment represents the full connectivity among all nodes. The initial connectivity matrix characterizes the connectivity of a plain scheme with no topology control. The rows i of the connectivity matrix, cm , represent arbitrary source nodes for one-hop transmissions

among first neighbors and the columns j are the corresponding adjacent destination nodes as shown in Figure 3.18

$$cm[i][j] = \begin{matrix} & \text{Columns } j \\ \text{Row } i & \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{pmatrix} \end{matrix}$$

Figure 3.18 A schematic view of a connectivity matrix.

3.8 Unicast Routing

Unicast routing mode is assumed for the computational experiments. That means one source node is connected to one destination node at a time. In the computational experiments where a node is composed of 3 or 4 optical sources, only one optical source (of 3 or 4) is activated at a time and like that there is only one effective queue with its average service rate. Transmission takes term in each time slot. In each time slot, there is only one packet being sent out. Figure 3.19 shows the transmission in timeslot.

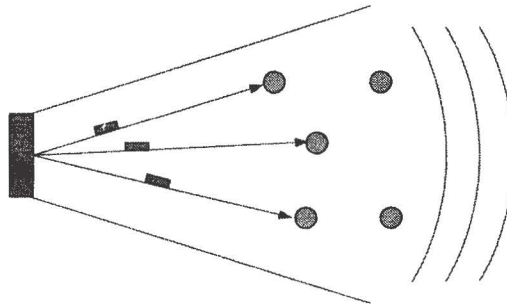


Figure 3.19 Asynchronous Transmission in timeslot.

This allows one to study the worst case of traffic flow that may occur at an edge between arbitrary nodes i and j . Assuming that there is only one single path available between arbitrary source nodes S and destination nodes D , consider the number of source nodes (denoted by N_s) that utilize the path $i \rightarrow j$ for routing, and the number of possible destination nodes (denoted by N_d) after traversing the path $i \rightarrow j$. The worse case of traffic flow that may occur is when the maximum number of N_s and N_d route traverse through the said edge $i \rightarrow j$ simultaneously.

3.9 Delay Calculation

For simplicity, the delay performance in this context is calculated using the basic M/M/1 queuing model[12] with the mean service rate of the single server, μ , and $\lambda_{\text{TOTAL}} = \lambda_1 + \lambda_2 + \dots$ is the total arrival rate at the input of a node based on the sum of multiple input flows from adjacent nodes entering into a FIFO queue. The mean delay can be expressed as a ratio with respect to a reference flow. $\lambda_{\text{REFERENCE}}$ related to the flow generated by a single source and is given in an arbitrary unit, a.u. To compute the *end-to-end delay* for different hop number, the mean *end-to-end delay per hop* is calculated with the assumption that the packet servicing capabilities of all the nodes are the same.

CHAPTER FOUR

NUMERICAL RESULTS

The parameters used for the computational experiments are fixed to certain acceptable normalized values with respect to a normalized channel rate, $R = 1$, as follows:

- Mean source rate of a single active node, $\lambda = [0,1]$;
- $\lambda_{\text{reference}} = 1/N$;
- Mean service rate of a server, $\mu = 1$.

Therefore, a single node will use a small fraction of the available channel bandwidth per port. It is to be expected that the increase of the number of nodes follows to a rapid congestion for a fixed value of λ .

The number of nodes is chosen to 7 and 31 nodes. Nevertheless, the computational experiments can be performed for arbitrary number of nodes within the range 0-100 where the upper limit arises due to the limited both operational memory and processing speed of the computer configurations in use. The shortest path algorithm is used for route selection in all computational experiments.

The computational experiments are based on the following sequence of tasks:

Topology Related Computations:

- Generation of a connectivity matrix for a sample topology;
- Set/Ring construction algorithm;

- Choose arbitrary shortest paths for unicast routing;

Quality of Service (QoS) Calculations:

- Calculate the frequencies of path overlapping on all the edges in the network;
- Calculate the throughput in each edge;
- Calculate the delay per node;
- Calculate the end-to-end delay for all active source-destination pairs;
- Calculate the end-to-end throughput for all active source-destination pairs;
- Calculate the mean end-to-end delay for different hop number;
- Calculate the mean end-to-end throughput for different hop number;

4.1 Performance of the Sample Topology of 7 nodes

4.1.1 Connectivity Matrix

Table 4.1 shows the connectivity matrix of the 7-node sample topology where “1” indicates the connectivity of the node i,j and “0” indicates the non-connectivity of the node i,j . For computational experiment case 3 (no topology control implemented), the reconfigured connectivity matrix will remain unchanged as in Table 4.1.

| | Neighbor Node 0 | Neighbor Node 1 | Neighbor Node 2 | Neighbor Node 3 | Neighbor Node 4 | Neighbor Node 5 | Neighbor Node 6 |
|--------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Node 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Node 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| Node 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| Node 3 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| Node 4 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| Node 5 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Node 6 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

Table 4.1 Connectivity Matrix, $cm[i][j]$, of 7-node Sample Topology (plain scheme).

However, for case 2 and case 3 where ring topology control is implemented, the initialized connectivity matrix is reconfigured by the set or ring construction algorithm. Table 4.2 shows the reconfigured connectivity matrix of the computational experiment case 1 (7 nodes, set/ring construction matrix, degree threshold = 3).

| | Neighbor Node 0 | Neighbor Node 1 | Neighbor Node 2 | Neighbor Node 3 | Neighbor Node 4 | Neighbor Node 5 | Neighbor Node 6 |
|--------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Node 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| Node 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Node 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| Node 3 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Node 4 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| Node 5 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Node 6 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

Table 4.2 Ring Connectivity Matrix, ring $[i][j]$, of 7-node Sample Topology.

Table 4.3 shows the available physical paths of pairs of nodes where they are not located adjacent to each other.

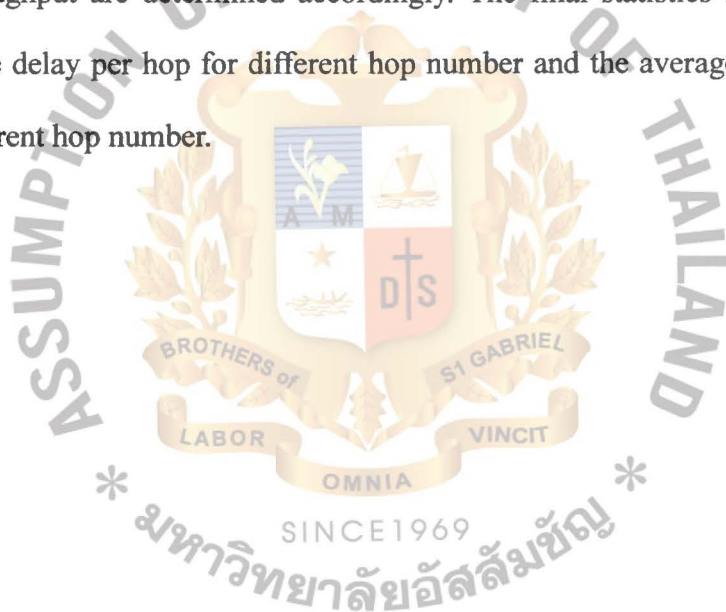
| |
|-----------------|
| 0 → 1 : 0 2 1 1 |
| 0 → 3 : 0 4 3 3 |
| 0 → 5 : 0 4 5 5 |
| 1 → 0 : 1 2 0 0 |
| 1 → 3 : 1 2 3 3 |
| 1 → 4 : 1 2 0 4 |
| 1 → 5 : 1 6 5 5 |
| 2 → 4 : 2 0 4 4 |
| 2 → 5 : 2 0 4 5 |
| 2 → 6 : 2 1 6 6 |
| 3 → 0 : 3 2 0 0 |
| 3 → 1 : 3 2 1 1 |
| 3 → 5 : 3 4 5 5 |
| 3 → 6 : 3 2 1 6 |
| 4 → 1 : 4 0 2 1 |
| 4 → 2 : 4 0 2 2 |
| 4 → 6 : 4 5 6 6 |
| 5 → 0 : 5 4 0 0 |
| 5 → 1 : 5 6 1 1 |
| 5 → 2 : 5 4 0 2 |
| 5 → 3 : 5 4 3 3 |
| 6 → 2 : 6 0 2 2 |
| 6 → 3 : 6 0 4 3 |
| 6 → 4 : 6 5 4 4 |

Table 4.3 Sample Available Physical Paths.

By means of the sequence of computational tasks, delay at each node and edge

throughput can be obtained. The edge throughput allows one to uncover the bottleneck edges, where the path overlapping results in low throughput. The delay at each node points out directly where the congested nodes are in the network. Based on delays at each node and edge throughput, the source to destination end-to-end delay and source to destination end-to-end throughput can be determined accordingly.

With the sequence of computational tasks, edge throughput, and delay at each node are obtained. On the basis of this information, the end-to-end delay and end-to-end throughput are determined accordingly. The final statistics is concerned with the average delay per hop for different hop number and the average throughput per hop for different hop number.



4.1.2 Delays in Sample Topology of 7 nodes

4.1.2.1 Delay at Each Node

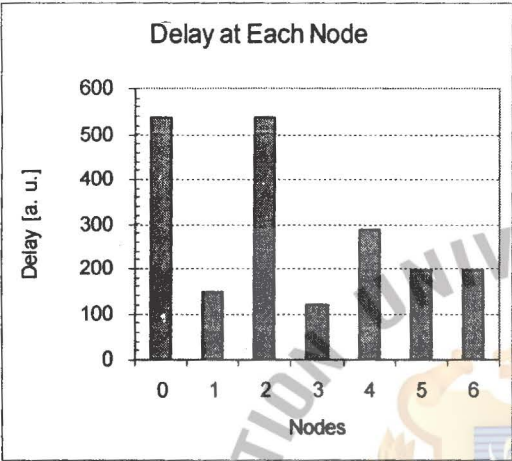
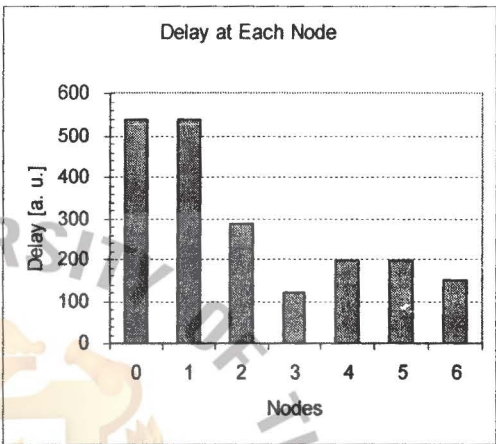
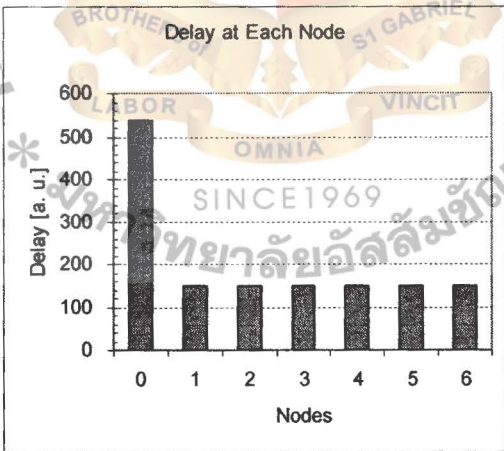
| - Delay at Each Node | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|---------------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|-------|---------------|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|
|  <table border="1"><caption>Data for Figure 4.1: 7 Nodes, Threshold 3, Ring scheme</caption><thead><tr><th>Nodes</th><th>Delay [a. u.]</th></tr></thead><tbody><tr><td>0</td><td>540</td></tr><tr><td>1</td><td>150</td></tr><tr><td>2</td><td>540</td></tr><tr><td>3</td><td>120</td></tr><tr><td>4</td><td>280</td></tr><tr><td>5</td><td>190</td></tr><tr><td>6</td><td>190</td></tr></tbody></table> | Nodes | Delay [a. u.] | 0 | 540 | 1 | 150 | 2 | 540 | 3 | 120 | 4 | 280 | 5 | 190 | 6 | 190 |  <table border="1"><caption>Data for Figure 4.2: 7 Nodes, Threshold 4, Ring scheme</caption><thead><tr><th>Nodes</th><th>Delay [a. u.]</th></tr></thead><tbody><tr><td>0</td><td>540</td></tr><tr><td>1</td><td>540</td></tr><tr><td>2</td><td>280</td></tr><tr><td>3</td><td>120</td></tr><tr><td>4</td><td>190</td></tr><tr><td>5</td><td>190</td></tr><tr><td>6</td><td>150</td></tr></tbody></table> | Nodes | Delay [a. u.] | 0 | 540 | 1 | 540 | 2 | 280 | 3 | 120 | 4 | 190 | 5 | 190 | 6 | 150 |
| Nodes | Delay [a. u.] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 540 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 540 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 120 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 280 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 190 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 190 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nodes | Delay [a. u.] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 540 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 540 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 280 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 120 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 190 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 190 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Figure 4.1 7 Nodes, Threshold 3, Ring scheme. | Figure 4.2 7 Nodes, Threshold 4, Ring scheme. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  <table border="1"><caption>Data for Figure 4.3: 7 Nodes, Plain scheme</caption><thead><tr><th>Nodes</th><th>Delay [a. u.]</th></tr></thead><tbody><tr><td>0</td><td>540</td></tr><tr><td>1</td><td>150</td></tr><tr><td>2</td><td>150</td></tr><tr><td>3</td><td>150</td></tr><tr><td>4</td><td>150</td></tr><tr><td>5</td><td>150</td></tr><tr><td>6</td><td>150</td></tr></tbody></table> | | Nodes | Delay [a. u.] | 0 | 540 | 1 | 150 | 2 | 150 | 3 | 150 | 4 | 150 | 5 | 150 | 6 | 150 | | | | | | | | | | | | | | | | |
| Nodes | Delay [a. u.] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 540 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 150 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Figure 4.3 7 Nodes, Plain scheme. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 4.1, 4.2, and Figure 4.3 show the delays at each node for the sample topology of 7 nodes in 3 scenarios.

Comparing the delays at each node for case 1 (7 nodes, Set/Ring construction

algorithm, degree threshold = 3) and case 2 (7 nodes, Set/Ring construction algorithm, degree threshold = 4), the overall delays are more or less the same. The highest delay is at 540 a.u. while the lowest delay is at 120 a.u. However, the set of nodes that are causing more delays are of slight difference. For instance, in case 1, the set of nodes that are causing more delays are node 0, 2, and 4 while in case 2, the set of nodes that are causing more delays are node 0, 1, and 2.

Comparing the delays at each node between the network with topology control and the network without topology control, both networks of degree threshold = 3 and degree threshold = 4 topology control show slightly higher delays than the network without topology control in terms of number of nodes that have higher delays. However, the delays at some nodes in the networks with topology control are occasionally lower, and the delays in the networks with topology control are comparable to the delays in the network without topology control.

4.1.2.2. End-to-End Delay

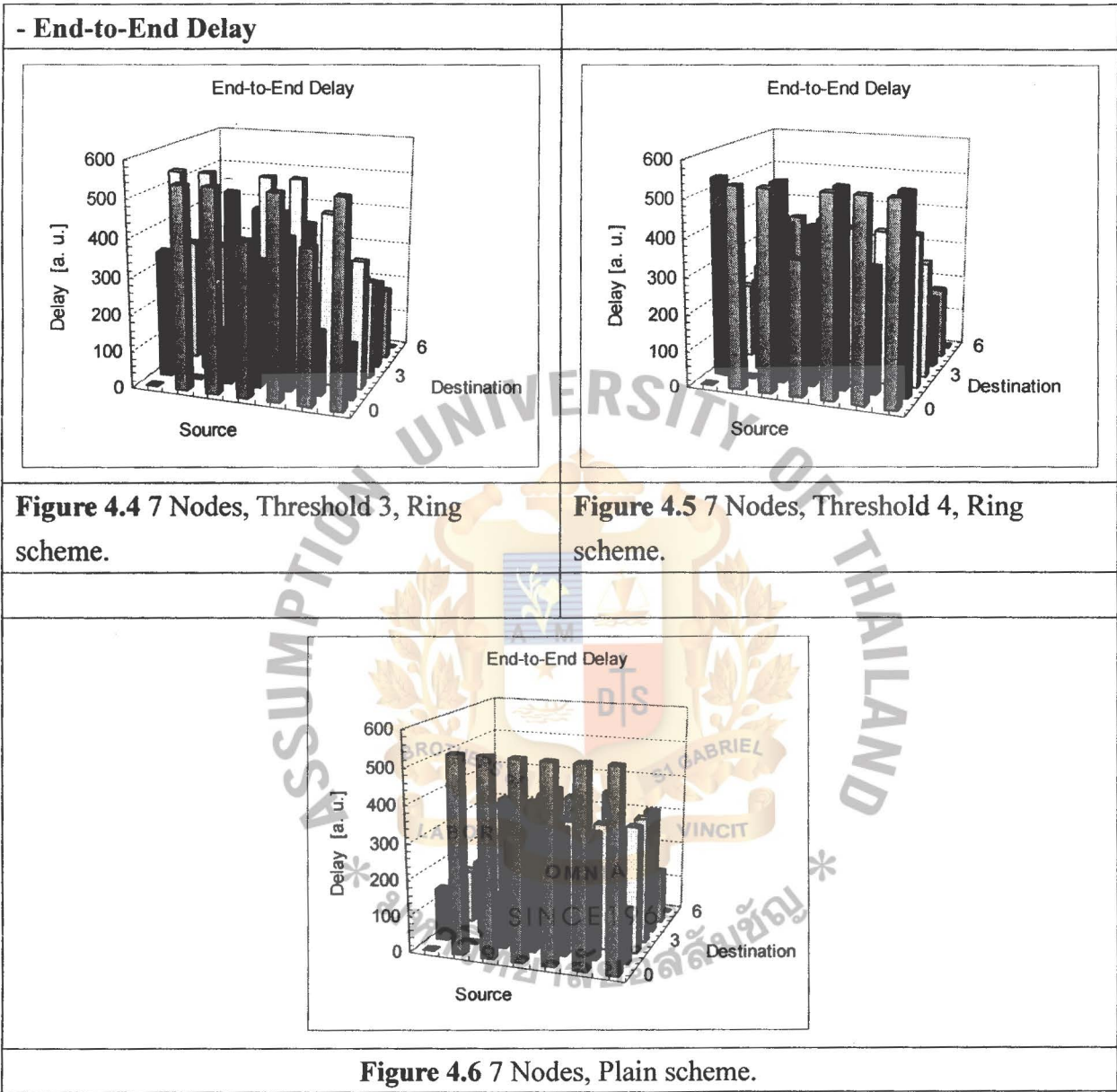
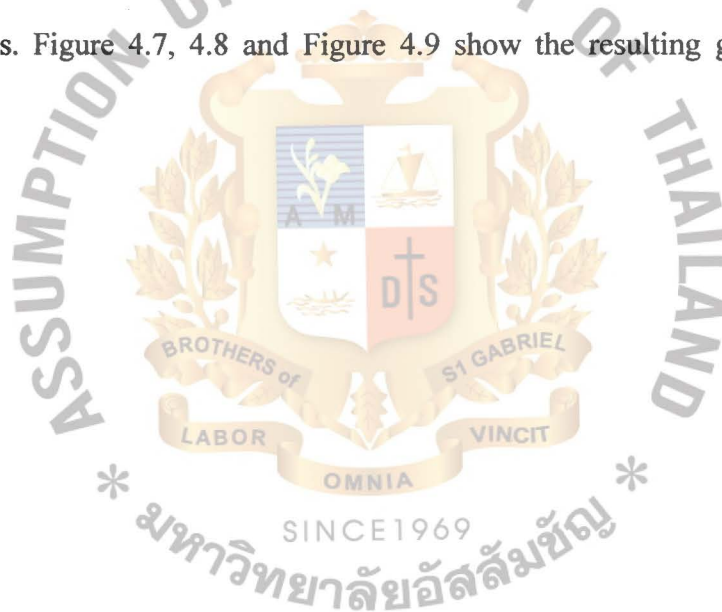


Figure 4.4, 4.5 and Figure 4.6 show the source to destination end-to-end delay for the sample network of 7 nodes in 3 scenarios.

Comparing the source to destination end-to-end delay between case 1 (7 nodes, Set/Ring construction algorithm, degree threshold = 3) and case 2 (7 nodes, Set/Ring construction algorithm, degree threshold = 4), the overall delays are about the same.

Comparing the source to destination end-to-end delay in networks with topology control and network without topology control, networks with topology control show slightly higher delays than the network without topology control. However, the delays at some source to destination links are occasionally lower, and the delays in the networks with topology control are believed to be comparable to the delays in the network without topology control.

To quantify the overall delays of the network with topology control and without topology control, the average delay per hop for different hop number is calculated for both 3 scenarios. Figure 4.7, 4.8 and Figure 4.9 show the resulting graphs of the calculation.



4.1.2.3 Average Delay per Hop for Different Hop Number

- Average Delay per Hop for Different Hop Number

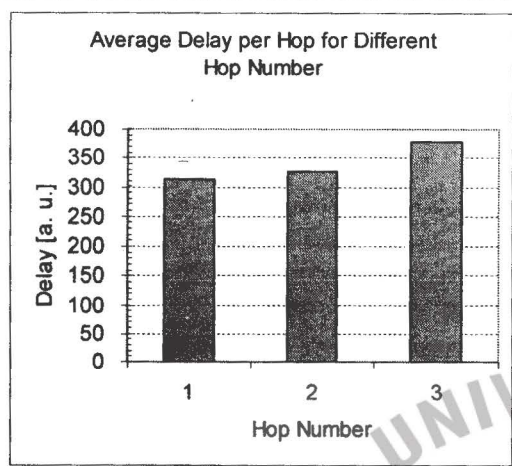


Figure 4.7 7 Nodes, Threshold 3, Ring scheme.

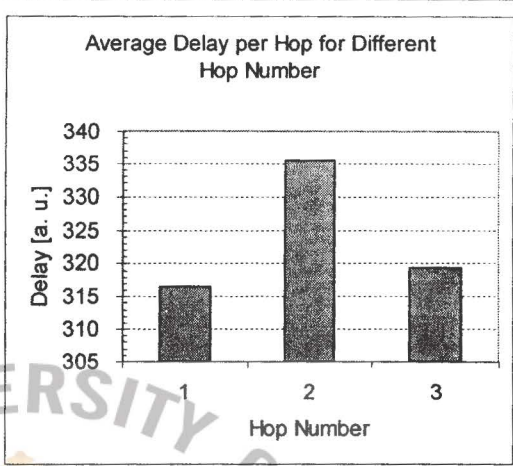


Figure 4.8 7 Nodes, Threshold 4, Ring scheme.

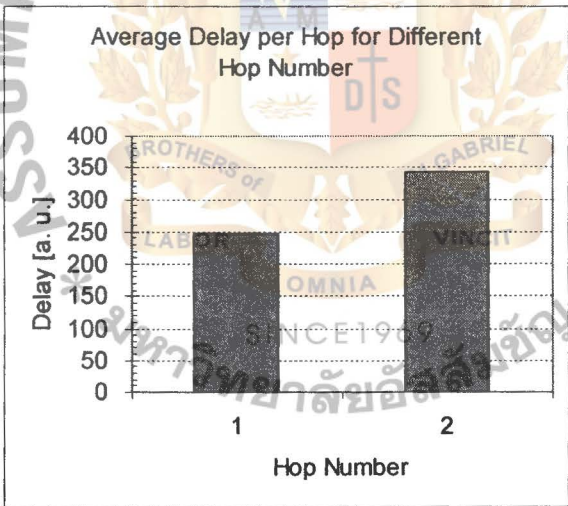


Figure 4.9 7 Nodes, Plain scheme.

Figure 4.7, 4.8 and Figure 4.9 show the average delay per hop for different hop number of the 3 scenarios in concern. Figure 4.9 shows that the network configuration without topology control is composed of and 1-hop and 2-hop paths while the networks with degree threshold = 3 and degree threshold = 4 are both composed of maximum 3-hop paths. Comparing the average delays among the 3 scenarios, the

network with topology control degree threshold =3 shows the highest average delays. The network with topology control degree threshold = 4 outperforms the other two network configurations. Although the average delay for 1-hop path in the network configuration of degree threshold = 4 shows a slightly higher delays than the average delay for 1-hop path in the network without topology control, the average delay for 2-hop path in the network configuration of degree threshold = 4 shows a slightly lower delays than the average delay for 2-hop path in the network without topology control. Moreover, the average delay of 3-hop path shows an even lower delay than the 2-hop path in the network configuration of degree threshold = 4.



4.1.3 Throughputs in Sample Topology of 7 nodes

4.1.3.1 Edge Throughput

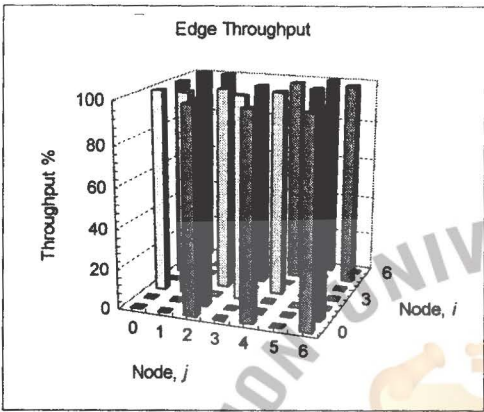
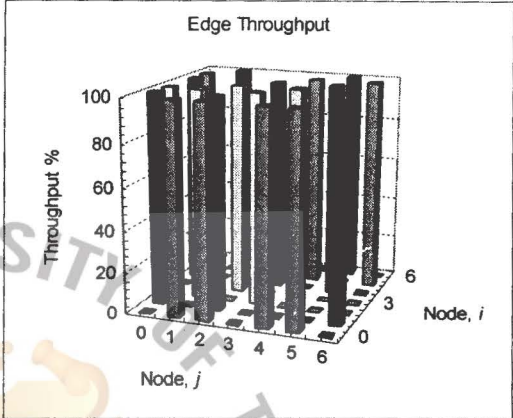
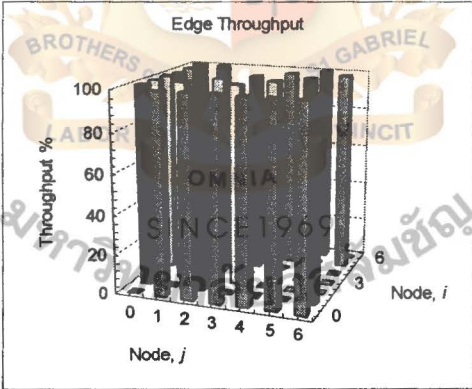
| - Edge Throughput | |
|---|--|
|  |  |
| Figure 4.10 7 Nodes, Threshold 3, Ring scheme. | Figure 4.11 7 Nodes, Threshold 4, Ring scheme.. |
|  | |
| Figure 4.12 7 Nodes, Plain scheme. | |

Figure 4.10, 4.11 and Figure 4.12 reveal that all 3 scenarios have the same level of edge throughput. In all three scenarios, all the edges of the networks show 100% edge throughput. The graphs are not the same because each scenario results in different number of links in the network. For case 1, the degree threshold is limited to 3 and therefore resulting in less number of edges. For case 3, there is no limitation on

the number of degree threshold. This network configuration results in more number of edges.

4.1.3.2 Average Throughput per Hop for Different Hop Number

- Average Throughput per Hop for Different Hop Number

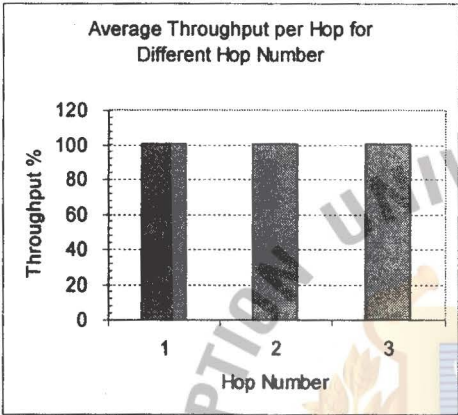


Figure 4.13 7 Nodes, Threshold 3, Ring scheme.

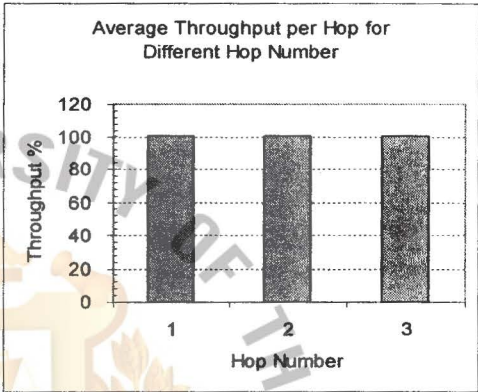


Figure 4.14 7 Nodes, Threshold 4, Ring scheme.

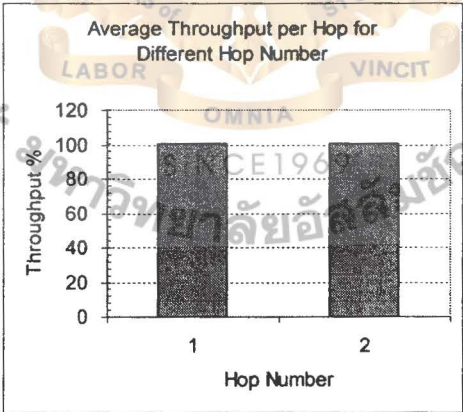


Figure 4.15 7 Nodes, Plain scheme.

Figure 4.13, 4.14 and 4.15 reveal that all 3 scenarios have the same level of average throughput per hop (100%) for paths with different hop number. However the difference between network with topology control and without topology control is that all paths in the network without topology control are 1-hop and 2-hop length and

the paths in the network with topology control have maximum 3-hop length. And these 3-hop paths also prove to provide 100% throughput.

4.1.4 Frequency of Edge Overlapping

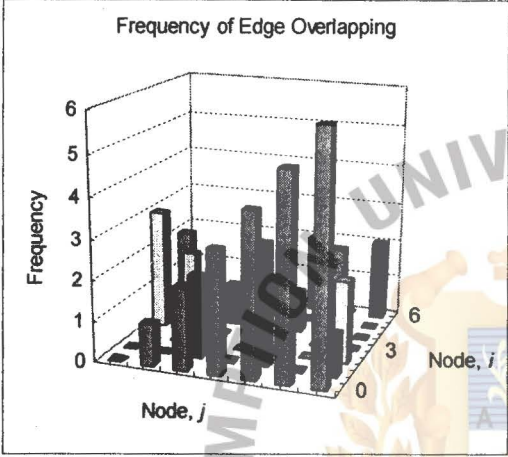
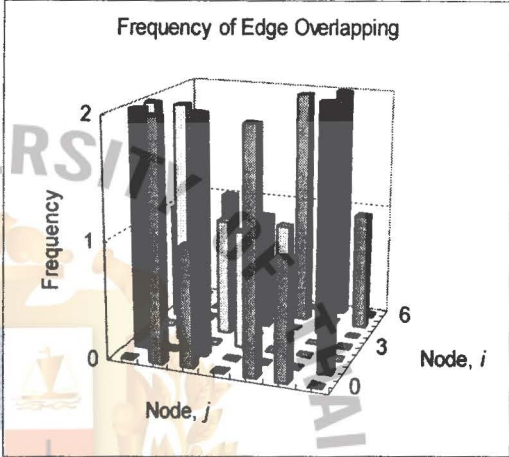
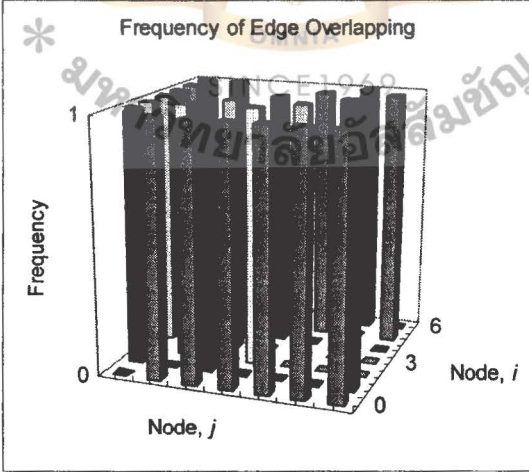
| - Frequency of Edge Overlapping | |
|---|---|
|  |  |
| Figure 4.16 7 Nodes, Threshold 3, Ring scheme. | Figure 4.17 7 Nodes, Threshold 4, Ring scheme. |
|  | |
| Figure 4.18 7 Nodes, Plain scheme. | |

Figure 4.16, 4.17 and Figure 4.18 show the frequency of the overlapping paths. In the network with topology control, frequency of the overlapping paths is higher due

to the fact that there is control in the number of connectivity. In the case without topology control, frequency of the overlapping paths equal to 1 for all edges because the center node has connectivity with all surrounding nodes. It can be seen that for any source to destination links that traverse these higher frequently used paths will result in higher end-to-end delays.

Considering all the figures shown above one can conclude that the topology control implemented in the network with 7 nodes does not have big effects on the performance in terms of overall delays and throughputs. This is probably explained by the very small size of network. Although the networks with topology control reduce the number of connectivity, the resulting delays are only slightly higher and in some occasions are lower, and throughputs can still be maintained at the same level as in the network without topology control.

4.2 Performance of the Sample Topology of 31 nodes

Table 4.4 shows the connectivity matrix of the 31-node sample topology where “1” indicates the connectivity of the nodes i,j and “0” indicates the non-connectivity of the nodes i,j .

4.2.1. Delays in Sample Topology of 31 nodes

For the network configurations consisting of 31 nodes, only two most informative graphs of delays are shown. They are delay at each node and average delay per hop for different hop number.



4.2.1.1 Delays at Each Node

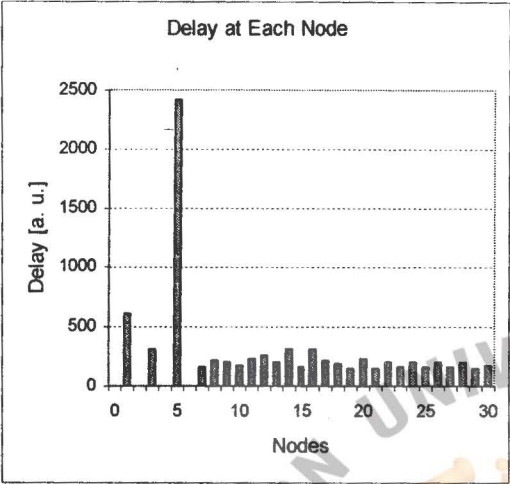
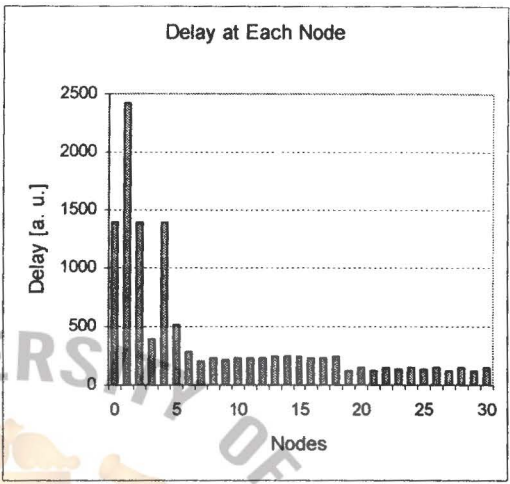
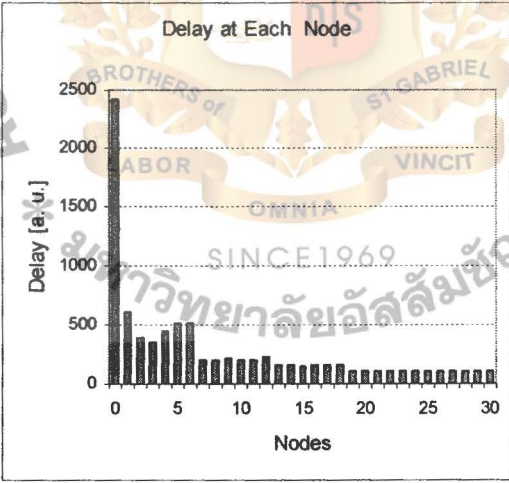
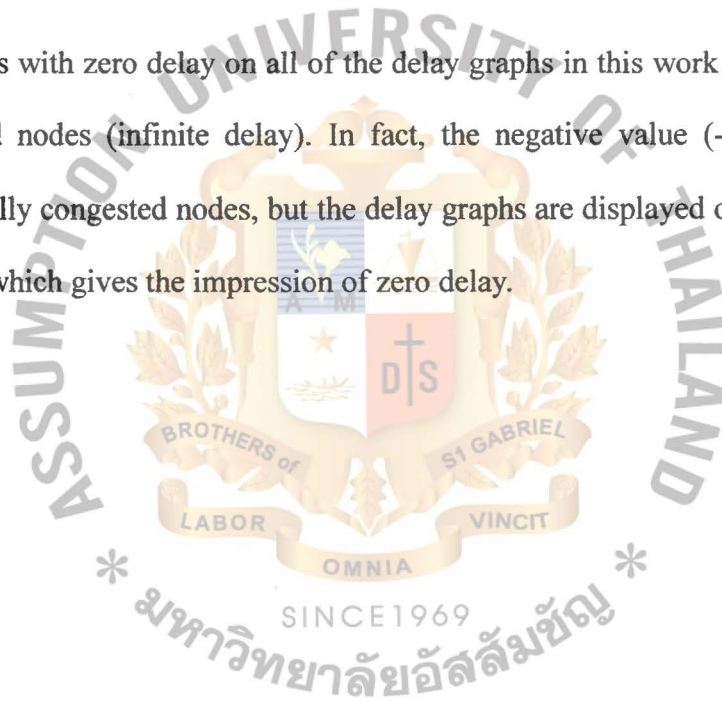
| | |
|---|--|
| - Delay at Each Node | |
|  |  |
| Figure 4.19 31 Nodes, Threshold 3, Ring scheme. | Figure 4.20 31 Nodes, Threshold 4, Ring scheme. |
|  | |
| Figure 4.21 31 Nodes, Plain scheme. | |

Figure 4.19, 4.20 and Figure 4.21 show the delays at each node for the 3 scenarios in concern. Figure 4.19 shows zero delays for a few number of the nodes in the network with topology control degree threshold = 3. These nodes are indeed fully congested with an infinite delay. However, the rest of the nodes in this network

configuration show a moderate delay at 300-400 a.u. range. Although there are no nodes showing infinite delays in the network with topology control degree threshold = 4, a few number of the nodes show a considerable delay at 400-600 a.u. and 1400 a.u. level, and the rest of the nodes in this network configuration show a moderate delay at 300-400 a.u. range. The network without topology control outperforms the other two networks with topology control since the delays associated with majority of the nodes are in the 100-300 a.u. range.

Note: The cases with zero delay on all of the delay graphs in this work correspond to fully congested nodes (infinite delay). In fact, the negative value (-1) is used to represent the fully congested nodes, but the delay graphs are displayed on the positive half-line only, which gives the impression of zero delay.



4.2.1.2 Average Delay per Hop for Different Hop Number

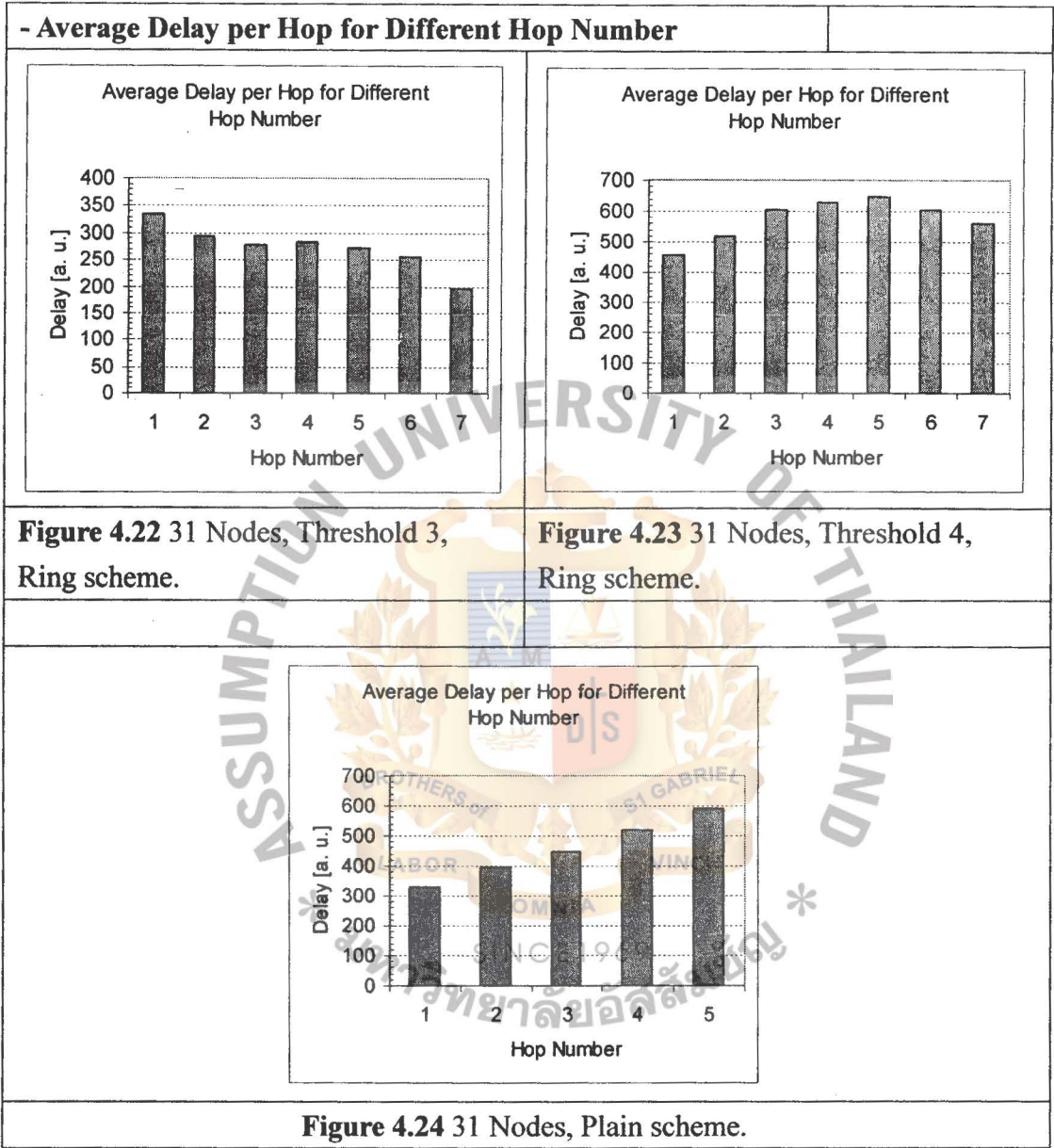


Figure 4.22, 4.23 and Figure 4.24 show the average delay per hop for different hop number of the 3 scenarios in concern. Figure 4.21 reveals that the average delay in the network with topology control degree threshold = 3 decreases as the hop number increases while the network without topology control is in reverse that the delays increase as the hop number increases. It is interesting to find that the average delay in the network with topology control degree threshold = 4 increases as the hop

number increases initially for paths with 1-hop to 5-hop, and the delays decreases gradually for paths with 6-hop and 7-hop. This should be explained by the effect of the topology control that reduces the number of connectivity of the nodes that have higher risk to become hot spot in the network. As a result of constrained number of connectivity, transmission routes become longer; however, the delays caused by the prospective congested nodes can be reduced.



4.2.2 Throughputs in Sample Topology of 31 nodes

4.2.2.1 Edge Throughput

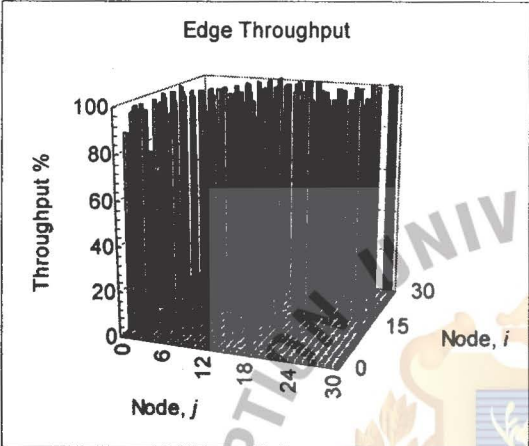
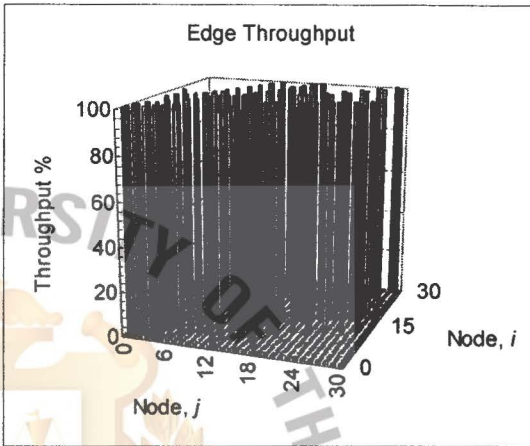
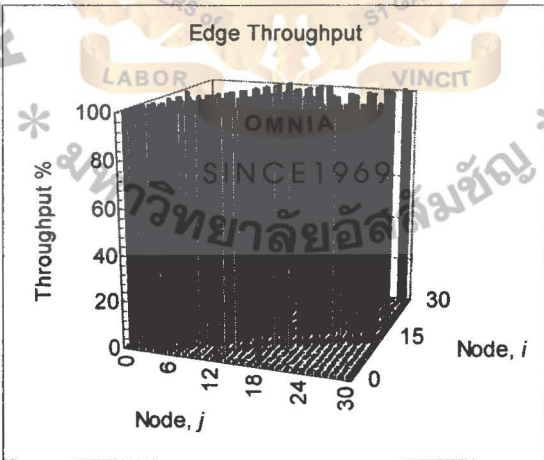
| | |
|---|--|
| - Edge Throughput | |
|  |  |
| Figure 4.25 31 Nodes, Threshold 3, Ring scheme. | Figure 4.26 31 Nodes, Threshold 4, Ring scheme. |
|  | |
| Figure 4.27 31 Nodes, Plain scheme. | |

Figure 4.25, 4.26, and Figure 4.27 show the edge throughput of the 3 scenarios in concern. The network with topology control degree threshold = 3 shows that the majority of paths have throughput level at the range between 80 – 100%. The network with topology control degree threshold = 4 and the network without topology

control show that all paths have throughput at 100% level.

4.2.2.2 Average Throughput per Hop for Different Hop Number

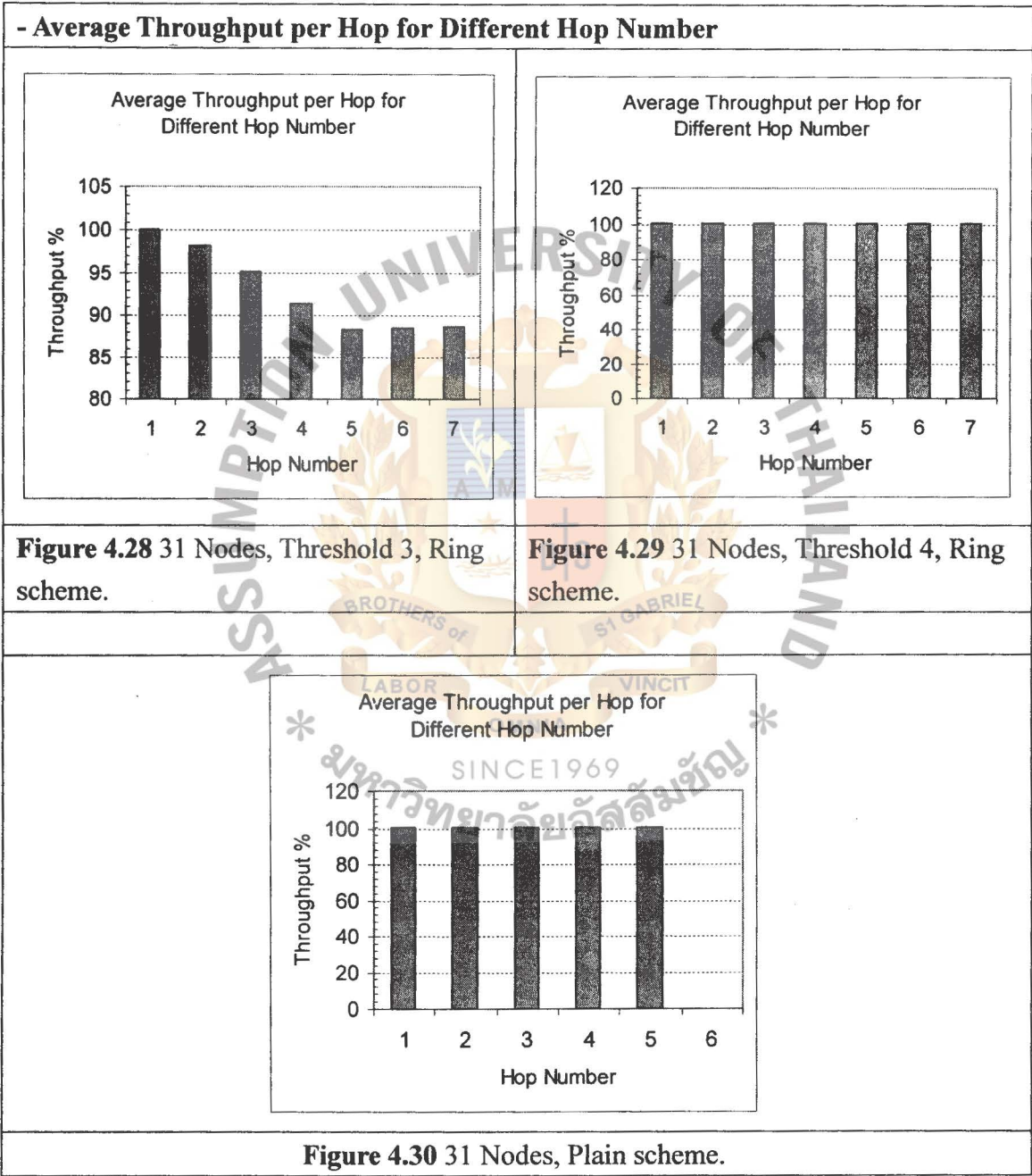
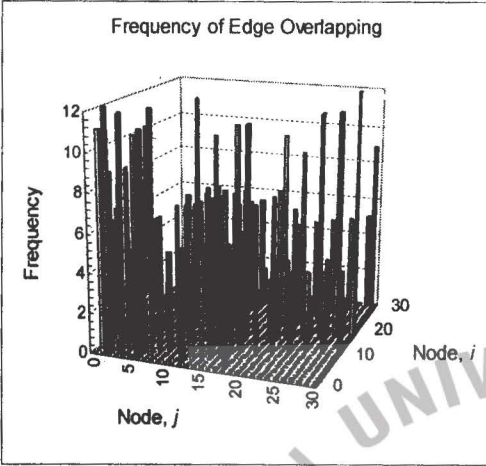
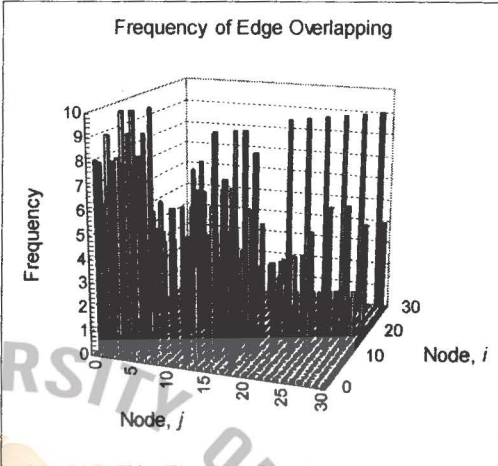
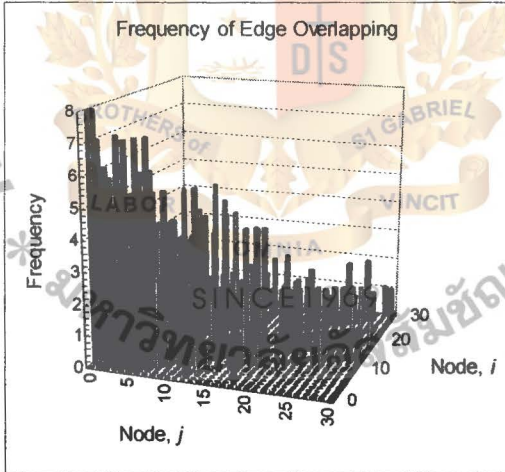


Figure 4.28, 4.29 and Figure 4.30 show the average throughput per hop for the 3 scenarios in concern. The network without topology control and the network with

topology control degree threshold = 4 show outstanding performance in terms of average throughput. Both the paths with low hop counts and the paths with high hop counts achieve the throughput at 100% level. Considering the average throughput per hop for the network with topology control degree threshold = 3, it is interesting to find that the average delay per hop decreases and the average throughput per hop increases as the hop number increases. This contradicting finding is explained by the calculations of the mean value of the delays where the fully congested paths are excluded in the calculation.



4.2.3 Frequency of Edge Overlapping

| | |
|---|--|
| - Frequency of Edge Overlapping | |
|  |  |
| Figure 4.31 31 Nodes, Threshold 3, Ring scheme. | Figure 4.32 31 Nodes, Threshold 4, Ring scheme. |
|  | |
| Figure 4.33 31 Nodes, Plain scheme. | |

From the above graphs of frequency of edge overlapping, one can see the frequency of edge overlapping is higher in the network with topology control. It can be concluded that the lower the degree threshold the higher the resultant frequency of edge overlapping. This is expectable because the degree threshold controls the number of connectivity. While there is reduced number of connectivity, overlapping

paths increased. While the frequency of overlapping paths is higher in the network with degree threshold = 3, it can be seen that the corresponded delays are also higher.

Considering all the figures related to the performance of the network with 31 nodes. One can see that with the increase of number of nodes, the congestion level in the network increases rapidly. The resulting delays in the networks without topology control are slightly higher but still comparable. However, with the topology control scheme implemented to reduce the number connectivity to threshold, the corresponded end-to-end throughput can still be maintained at the same 100% level.



CHAPTER FIVE

ANALYSIS AND DISCUSSION

The most important statistical information of delays and throughputs for comparisons among different network configuration is to study the mean values for different hop number. Figure 5.1 shows the average delay per hop for different hop number of the sample 31-node topology.

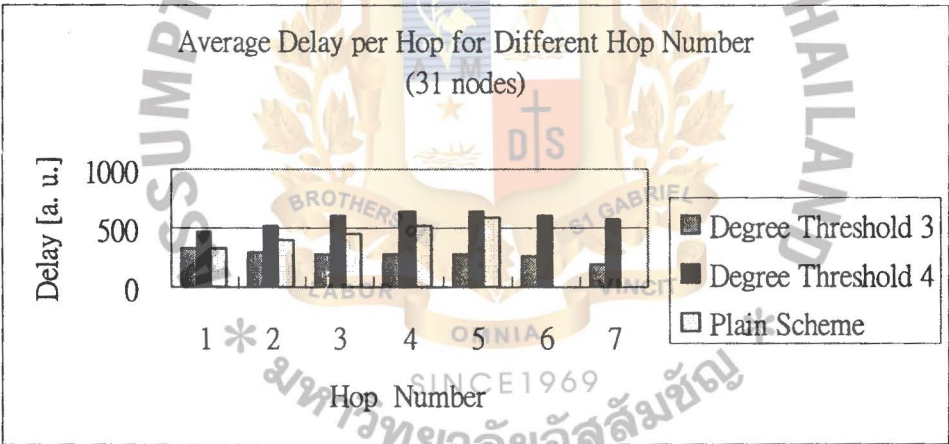


Figure 5.1 The Average Delay per Hop for Different Hop Number of the Sample 31-node Topology.

The network with topology control degree threshold = 3 seems to provide the least average delay per hop; however, it needs to be mentioned that the mean end-to-end delay is calculated by the averaging of non-congested paths only. Therefore, the effects of the fully congested nodes that appear in the network with topology control degree threshold = 3 are not concerned in calculating the average delay per hop. For this reason, it cannot be concluded that the network with topology

control degree threshold 3 outperforms the other two network configurations in terms of delays. In order to judge the performance among the 3 scenarios, it needs to take into concern of the average throughput as well.

The network with topology control degree threshold = 4 and the network without topology control have the same tendency of increased average delay per hop as the hop number increases. The average delay per hop in the network with topology control degree threshold = 4 is higher than that in the network without topology control; however, the difference is not much. They are rather comparable. One interesting point here is that the average delays per hop in the network with topology control degree threshold = 4 increases as the hop number increases and decreases gradually after 6-hop hop number. The average delay per hop for paths with 6 hops and 7 hops in the network with topology control degree threshold = 4 eventually have the same level of average delay as the paths with 5 hops in the network without topology control. This can be explained by the effect of the topology control scheme where the reduced connectivity helps alleviate the problem of hot spots causing delays in the network and therefore resulting in longer transmission paths.

Figure 5.2 shows the throughput per hop for different hop number of the sample 31-node topology. The throughput reduces steadily with the increase of the hop number for the network with topology control degree threshold = 3. However, the average throughput is still maintained in a relatively high level of 80-90%. The average throughput of the network with topology control degree threshold = 4 shows the same resultant throughput at 100% level for all paths.

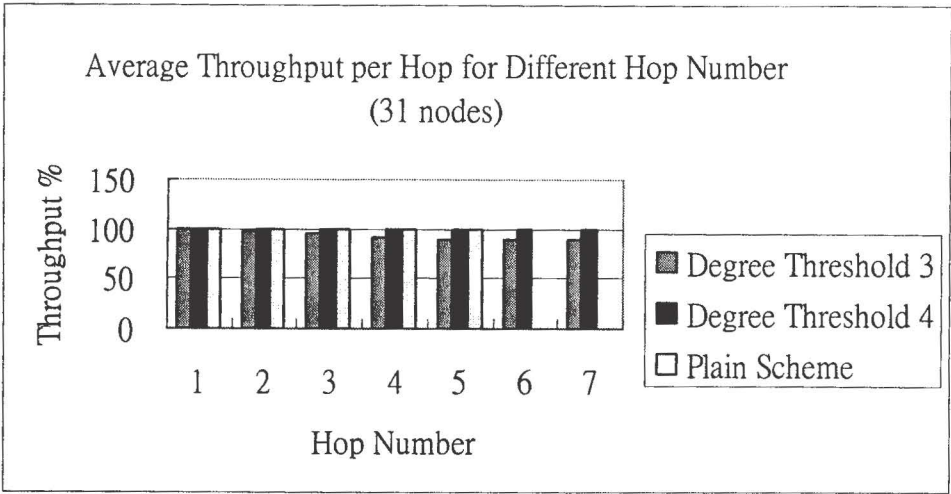


Figure 5.2 The Average Throughput per Hop for Different Hop Number of the Sample 31-node Topology.

Since the mean end-to-end delay is calculated by the averaging of non-congested paths only. The mean end-to-end throughput is calculated by the averaging of all the paths. In order to observe the peculiarities of hot spots one should view the individual edge throughput and node delays as it has been demonstrated in Chapter 4. In the three-dimensional graphs shown in Chapter 4, infinite end-to-end delays due to congestion are shown with a negative value of (-1). One can determine the percentage of fully congested paths by simply counting the negative values. From the calculations, the percentages of fully congested path for the network configuration without topology control, with degree threshold = 4 and with degree threshold = 3 are 0%, 0% and 56% respectively. This explains why the average delay per hop in the network with topology control degree threshold = 3 decreases as the hop number increases, and at the same time, the average throughput per hop in this network configuration decreases as the hop number increases.

From the experiments of different sizes of networks with and without topology

control, the statistical information of the delays and throughputs associated with each type of network reveal that delays is higher in the networks with topology control while the throughputs are almost the same in the networks with and without topology control. This is expected because the topology control here is to maintain the stability of the network with a very small number of interconnections.

However, it is found that the delays in the networks with topology control are occasionally lower than that of without topology control, especially when the network is at higher congestion level. This finding can be verified by changing the lambda parameter (which is the mean source rate per single source node) starts from lambda = 0.01 then gradually increases to 0.09 with a step of 0.02. When the congestion is well shown, the percentage of the fully congested nodes indicated by the negative value (-1) rapidly increase. The results are shown in Figure 5.3.

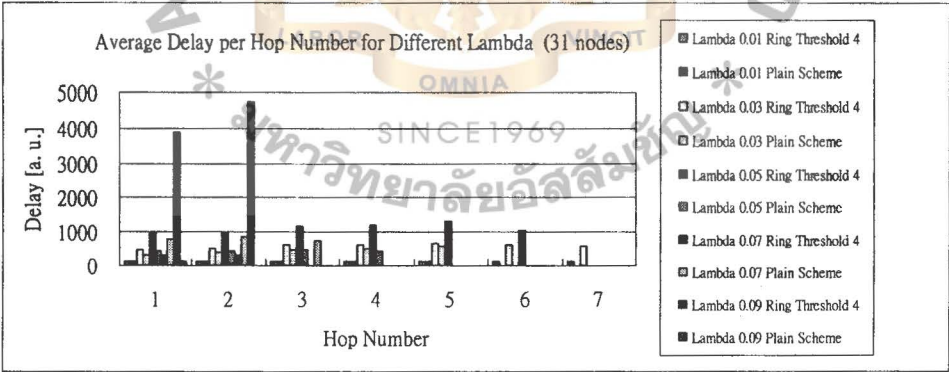


Figure 5.3 Average Delay per Hop Number for Different Lambda Value.

One can see that when the source rate is lower, the average delays in the network with topology control are higher than the average delays in the network without topology control. When the source rate is at 0.07 and 0.09, the delays of the network with topology control (threshold = 4) are occasionally lower than the delays of the

network without topology control and the delays are twice lower than that of the delays in the network without topology control. The possible explanation behinds this is that when a network is very congested, the reduction of connectivity to any specific node helps alleviate the problem of hot spots causing the increase loss of packets and consequent network failure.

The statistical information of the source to destination end-to-end delays and source to destination end-to-end throughput also reveals that for small size network with small number of nodes, whether having topology control or not does not provide any difference with respect to the throughputs of the network. However, in the cases of bigger size network (for instance, the 31-node network being created for the computational experiments), the proposed topology control proves to achieve the same level of throughput on the basis of reduced connectivity while maintaining full connectivity network. While full connectivity network can be maintained, the proposed topology; on the other hand helps alleviate the problem of no-connection and leaf nodes in some parts of the network.

CHAPTER SIX

CONCLUSION

The main contributions of this thesis can be summarized as follows:

1. A program for numerical evaluation of routing in underwater optical wireless network is developed. This program allows one to test the performance of arbitrary planar networks on the basis of computational experiments.
2. The Set/Ring Construction Algorithm is proposed and studied. The Set/Ring Construction Algorithm deals with the reductions of connections with any specific node in any given network in order to avoid the occurrence of hot spots in the network.
3. The statistical information obtained from the computational experiments of different network settings reveals that the proposed Set/Ring Construction Algorithm is capable to provide routing stability with a very limited number of connections that is required by the underwater optical wireless network.

Considering the resulting delays and throughputs of the 31-node sample topology with topology control degree threshold = 3 and degree threshold = 4, the optimal degree threshold for a network size of 31 nodes should be 4 nodes since the network with degree threshold = 4 outperforms the network with degree threshold = 3 in terms of both delays and throughputs. However, the optimal node degree threshold for any given size and configuration of networks are not included in this thesis. This study

can be considered as an initial attempt to provide computational information for further studies.



REFERENCES

- [1] Akyildiz I. F.; Pompili Dario; and Melodia T. "Challenges for Efficient Communication in Underwater Acoustic Sensor Networks" *ACM Sigbed Review*, Vol. 1, No. 2, July 2004.
- [2] Giles J. W.; and Bankman I. N., "Underwater Optical Communications Systems Part 2: Basic Design Considerations," *Military Communications Conference, 2005. MILCOM 2005*, Vol. 3, pp. 1700-1705, October, 2005
- [3] Schill F.; Zimmer W. R.; and Trumpf J., "Visible Spectrum Optical Communication and Distance Sensing for Underwater Applications," *Proceedings of the. 2004. Australasian Conference. on. Robotics. and. Automation.*, Canberra, Australia, December, 2004.,
- [4] Jagdishlal G. Y., "Underwater Free Space Optics," Master's Thesis, Master of Science in Electrical Engineering, North Carolina State University, N.C., U.S.A., December, 2006
- [5] Cheng C.; Wendi B. "Exploring Long lifetime Routing (LLR) in ad hoc networks" *MSWiM'04*, October 2004
- [6] <http://www.free-space-optics.org/> "Free Space optics (FSO): An Introduction"
- [7] Alan C. M. "Short Range Underwater Optical Communication Links," Master's Thesis, Master of Science in Electrical Engineering, North Carolina State University, N.C., U.S.A., June, 2005
- [8] D'iaz J.; Petit J.; and Serna M., "A random graph model for optical networks of sensors," *IEEE Transactions on Mobile Computing*, Vol. 2, No. 3, pp. 186-196, July-September 2003.
- [9] Llorca J.; Desai A.; Vishkin U.; Davis ;. and Milner S., "Reconfigurable optical

wireless sensor networks,” in *Proc. SPIE vol. 5237, Optics in Atmospheric Propagation and Adaptive Systems VI*, J. D. Gonglewski and K. Stein, Eds., Barcelona, Spain, February 2004, pp. 136–146.

[10] Risk W.; Gosnell T. ; and Nurmikko A., “The Need for Compact Blue-Green Lasers,” Cambridge University Press, *Text Book*, Published January 2003.

[11] Akyildiz I. F.; Pompili D.; and Melodia T., “Underwater Acoustic Sensor Networks: Research Challenges,” *Elsevier's Journal of Ad Hoc Networks*, Vol. 3(3), pp. 257--279, May 2005

[12] L. Kleinrock, *Queueing Systems*, Volumes 1 and 2, John Wiley & Sons, 1976.



APPENDICES

A.1 Source Code of Set/Ring Construction Algorithm

void **Ring_Construction** (int mode,int threshold,int N,short **cm, short **ring)

```
{    int i,j,k,l,m,p,q,count,cn,dummy,left,right,hopsold,hops,set_no;

    int *nodes,*node_degree,*sets,*seta,*ring_member;

    div_t div_result;

    count=0;dummy=0;left=0;right=0;hopsold=0;hops=0;

    nodes=ivector(0,N);node_degree=ivector(0,N);

    sets=ivector(0,N);seta=ivector(0,N);ring_member=ivector(0,N);

    //    Initialize the ring connectivity matrix

    i=0;while(i<N){j=0;while(j<N){if(cm[i][j]!=0){cm[i][j]=1;ring[i][j]=0;j++;}i++;}

    i=0;while(i<N){j=0;while(j<N){if(cm[i][j]>0){cm[j][i]=cm[i][j];}j++;}i++;}

    //    Sort the nodes according to their node degree to select the central node

    i=0;while(i<N){nodes[i]=i;node_degree[i]=0;ring_member[i]=0;i++;}

    i=0;while(i<N){j=0;while(j<N){if(cm[i][j]!=0){node_degree[i]++;}j++;}i++;}

    Shell3(1,N,2,node_degree-1,nodes-1,nodes-1);

    i=0;while(i<N){j=0;while(j<N){if(cm[i][j]!=0){node_degree[i]++;}j++;}i++;}

    i=0;while(i<N){cn=nodes[i];set_no=0;

    //    Create sets/ring of interconnected first neighbors around a central node with maximum

    node degree
```



```

if (node_degree[cn]>2){if(ring_member[cn]==0){j=0;while(j<N){sets[j]=0;seta[j]=(-1);
if((j!=cn)&&(cm[cn][j]!=0)&&(node_degree[j]>1)){sets[j]=(-1);j++;}j=0;while (j<N){
if ((j!=cn)&&(sets[j]==(-1))) {set_no++;sets[j]=set_no;ring_member[j]=1;left=j;right=j;
hopsold=0;hops=0;count=1;seta[count]=j;k=0;while (k<N){hopsold=hops;
if((k!=cn)&&(k!=left)&&(sets[k]==(-1))) {if(cm[k][left]>0){ring[k][left]=1;
ring_member[k]=1;left=k;sets[k]=set_no;hops++;count++;l=count;while(l>1){
seta[l]=seta[l-1];l--;}seta[l]=k;}}if(right!=left){if((k!=cn)&&(k!=right)&&(sets[k]==(-1))) {
if(cm[k][right]>0){ring[k][right]=1;ring_member[k]=1;right=k;sets[k]=set_no;hops++;
count++;seta[count]=k;}}if(hops>hopsold){k=0;k++;}
}
// Connect the ring to the central node
if (cm[left][right]>0){ring[left][right]=1;div_result=div(count,2);
m=div_result.quot+div_result.rem;if(m<1){m=1;};if(m>count){m=count;};
ring[cn][seta[m]]=1;div_result=div(count,threshold);if (div_result.quot>0){
p=div_result.rem;q=m;l=1;while(l<threshold){l++;q=q-div_result.quot;
if (q<1){q=count-div_result.quot+1;};ring[cn][seta[q]]=1;if(l<threshold){if(p>0){p--;l++;
q=q-div_result.quot-1;if (q<1){q=count-div_result.quot+1;};ring[cn][seta[q]]=1;}}}}
else{l=1;while(l<=count){ring[cn][seta[l]]=1;l++;}k=0;while (k<N){if(sets[k]==set_no)
{cout<<k<<" ";k++;}cout<<"<<cn<<" Ring!\n";}

// Connect the set to the central node
else{div_result=div(count,2);m=div_result.quot+div_result.rem;if(m<1){m=1;};if(m>count)
{m=count;};ring[cn][seta[m]]=1;}}j++;}}

// Connect leaf nodes (degree=1) and relay nodes (degree=2)
else{j=0;while (j<N){ring[cn][j]=cm[cn][j];j++;}i++;}

```

// **Undirected Links**

```
i=0;while(i<N){j=0;while(j<N){if(ring[i][j]>0){ring[j][i]=ring[i][j];}j++;}i++;}
```

```
free_ivector(ring_member,0,N);free_ivector(seta,0,N);free_ivector(sets,0,N);
```

```
free_ivector(node_degree,0,N);free_ivector(nodes,0,N);}
```

// **End Ring**



A.2 Source Code for Obtaining the Sample Topologies

```
void Connectivity_Matrix(int N,short **cm)

{int i,j,dummy;i=0;while(i<=N){j=0;while(j<=N){cm[i][j]=0;j++;}i++;}

//   Sample Topologies Consisting of 7 and 31 Nodes

switch (N){

case 7:

    cm[0][1]=1;cm[0][2]=1;cm[0][3]=1;cm[0][4]=1;cm[0][5]=1;cm[0][6]=1;

    cm[1][0]=1;cm[1][2]=1;cm[1][6]=1;

    cm[2][0]=1;cm[2][1]=1;cm[2][3]=1;

    cm[3][0]=1;cm[3][2]=1;cm[3][4]=1;

    cm[4][0]=1;cm[4][3]=1;cm[4][5]=1;

    cm[5][0]=1;cm[5][4]=1;cm[5][6]=1;

    cm[6][0]=1;cm[6][5]=1;cm[6][1]=1;

    break;

case 31:

    cm[0][1]=1;cm[0][2]=1;cm[0][3]=1;cm[0][4]=1;cm[0][5]=1;cm[0][6]=1;

    cm[1][0]=1;cm[1][2]=1;cm[1][6]=1;

    cm[2][0]=1;cm[2][1]=1;cm[2][3]=1;

    cm[3][0]=1;cm[3][2]=1;cm[3][4]=1;

    cm[4][0]=1;cm[4][3]=1;cm[4][5]=1;

    cm[5][0]=1;cm[5][4]=1;cm[5][6]=1;

    cm[6][0]=1;cm[6][5]=1;cm[6][1]=1;

    cm[7][1]=1;cm[7][2]=1;cm[7][13]=1;cm[7][18]=1;cm[7][19]=1;cm[7][20]=1;

    cm[1][7]=1;cm[1][2]=1;cm[1][18]=1;
```

$$cm[2][7]=1;cm[2][1]=1;cm[2][13]=1;$$

$$cm[13][7]=1;cm[13][2]=1;cm[13][20]=1;$$

$$cm[18][7]=1;cm[18][1]=1;cm[18][19]=1;$$

$$cm[19][7]=1;cm[19][18]=1;cm[19][20]=1;$$

$$cm[20][7]=1;cm[20][13]=1;cm[20][19]=1;$$

$$cm[8][2]=1;cm[8][3]=1;cm[8][13]=1;cm[8][14]=1;cm[8][21]=1;cm[8][22]=1;$$

$$cm[2][8]=1;cm[2][3]=1;cm[2][13]=1;$$

$$cm[3][8]=1;cm[3][2]=1;cm[3][14]=1;$$

$$cm[13][8]=1;cm[13][2]=1;cm[13][21]=1;$$

$$cm[14][8]=1;cm[14][3]=1;cm[14][22]=1;$$

$$cm[21][8]=1;cm[21][13]=1;cm[21][22]=1;$$

$$cm[22][8]=1;cm[22][14]=1;cm[22][21]=1;$$

$$cm[9][3]=1;cm[9][4]=1;cm[9][14]=1;cm[9][15]=1;cm[9][23]=1;cm[9][24]=1;$$

$$cm[3][9]=1;cm[3][4]=1;cm[3][14]=1;$$

$$cm[4][9]=1;cm[4][3]=1;cm[4][15]=1;$$

$$cm[14][9]=1;cm[14][3]=1;cm[14][23]=1;$$

$$cm[15][9]=1;cm[15][4]=1;cm[15][24]=1;$$

$$cm[23][9]=1;cm[23][14]=1;cm[23][24]=1;$$

$$cm[24][9]=1;cm[24][15]=1;cm[24][23]=1;$$

$$cm[10][4]=1;cm[10][5]=1;cm[10][15]=1;cm[10][16]=1;cm[10][25]=1;cm[10][26]=1;$$

$$cm[4][10]=1;cm[4][5]=1;cm[4][15]=1;$$

$$cm[5][10]=1;cm[5][4]=1;cm[5][16]=1;$$

$$cm[15][10]=1;cm[15][4]=1;cm[15][25]=1;$$

$cm[16][10]=1;cm[16][5]=1;cm[16][26]=1;$

$cm[25][10]=1;cm[25][15]=1;cm[25][26]=1;$

$cm[26][10]=1;cm[26][16]=1;cm[26][25]=1;$

$cm[11][5]=1;cm[11][6]=1;cm[11][16]=1;cm[11][17]=1;cm[11][27]=1;cm[11][28]=1;$

$cm[5][11]=1;cm[5][6]=1;cm[5][16]=1;$

$cm[6][11]=1;cm[6][5]=1;cm[6][17]=1;$

$cm[16][11]=1;cm[16][5]=1;cm[16][27]=1;$

$cm[17][11]=1;cm[17][6]=1;cm[17][28]=1;$

$cm[27][11]=1;cm[27][16]=1;cm[27][28]=1;$

$cm[28][11]=1;cm[28][17]=1;cm[28][27]=1;$

$cm[12][1]=1;cm[12][6]=1;cm[12][17]=1;cm[12][18]=1;cm[12][29]=1;cm[12][30]=1;$

$cm[1][12]=1;cm[1][6]=1;cm[1][18]=1;$

$cm[6][12]=1;cm[6][1]=1;cm[6][17]=1;$

$cm[17][12]=1;cm[17][6]=1;cm[17][29]=1;$

$cm[18][12]=1;cm[18][1]=1;cm[18][30]=1;$

$cm[29][12]=1;cm[29][17]=1;cm[29][30]=1;$

$cm[30][12]=1;cm[30][18]=1;cm[30][29]=1;$

break;

A.3 Source Code of Shell Sorting Algorithm

```
void Shell3(int MinN,int MaxN,int n,int *x,int *y,int *z)

{int gap,i,j,jg;int xx;

div_t div_result;

div_result=div(MaxN-(MinN-1),2);gap=div_result.quot;

while (gap>(MinN-1)){i=gap;while (i<MaxN){i++;j=i-gap;while (j>0){jg=j+gap;

if (x[j]>=x[jg]) j=0;else{xx=x[j];x[j]=x[jg];x[jg]=xx;switch (n){

case 2:

xx=y[j];y[j]=y[jg];y[jg]=xx;break;

case 3:

xx=y[j];y[j]=y[jg];y[jg]=xx;xx=z[j];z[j]=z[jg];z[jg]=xx;break;}}j=j-gap;}}

div_result=div(gap,2);gap=div_result.quot;}}
```

