

**Development of Energy Efficient Network Architecture and
Protocol Stack for Underwater Wireless Sensor Networks
(UWSNs)**

THESIS

Submitted in partial fulfillment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY

by

DHONGDI SARANG CHANDRASHEKHAR

Under the Supervision of

Prof. K. R. Anupama

and

Co-supervision of

Dr. Lucy J. Gudino




BITS Pilani
Pilani|Dubai|Goa|Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

2016

Certificate

This is to certify that the thesis entitled “**Development of Energy Efficient Network Architecture and Protocol Stack for Underwater Wireless Sensor Networks (UWSNs)**” and submitted by **Dhongdi Sarang Chandrashekar**, ID. No. **2010PHXF022G** for award of Ph.D. of the Institute embodies original work done by him under our supervision.

Signature of the Supervisor 

Name in capital letters : **Prof. K. R. Anupama**

Designation : **Professor**

Date : **12/10/2016**

Signature of the Co-supervisor 

Name in capital letters : **Dr. Lucy J. Gudino**

Designation : **Assistant Professor**

Date : **12/10/2016**

Abstract

Wireless Sensor Networks (WSNs) deployed underwater are known as Underwater Wireless Sensor Networks (UWSNs). If acoustic signals are used to form the wireless communication links between sensor nodes, then such networks are termed as Underwater Acoustic Sensor Networks (UASNs).

In this work, a problem of designing three-dimensional UASN has been considered. This three-dimensional UASN can be used for various applications in the field of oceanography such as pollution monitoring or ecosystem observation of ocean column.

For designing such UASN, a static and structured three-dimensional network architecture has been proposed. In the proposed architecture, network is arranged in the form of clusters of sensor nodes at varying depths below sea-surface. A single node is deployed at sea-surface using a buoy. This node serves as a Base-Station and has the RF link to communicate with an on-shore control station. Every cluster deployed at varying depths below sea-surface has one Cluster-Head node and certain number of cluster member nodes. During deployment, nodes are programmed with their functionalities and corresponding location. It is assumed that sensor nodes are equipped with oceanographic sensors, batteries, processing units, acoustic modems and a suitable mechanism to keep them afloat at desired depth in the ocean.

For a long operational life of the proposed three-dimensional UASN, energy efficiency is very critical design parameter. In this work, an energy efficient cross-layer protocol stack has been developed for the three-dimensional network. The proposed cross-layer protocol stack encompasses the important functionalities of physical layer, data link layer, network layer, transport layer as well as application layer. Also, various protocols of network management planes such as time synchronization, Cluster-Head selection and power level management has been incorporated. This cross-layer protocol stack has following features and functionalities:

- For transmission of information, optimum power levels have been chosen by the nodes in the network. These power levels have been calculated from the physical layer parameters such as distance of communication, modulation technique and Signal-to-Interference-plus-Noise Ratio (SINR). This provision of dynamic power level adjustment has ensured energy efficient communication at physical layer.

-
- An energy efficient TDMA based MAC protocol has been proposed and incorporated in the cross-layer protocol stack. This MAC protocol has been termed as “Dynamic Cluster-Based TDMA (DCB-TDMA)” MAC protocol. In this protocol, every node has certain number of pre-assigned time slots for transmission and reception. Nodes have been put to sleep mode for duration other than the assigned time slots to conserve energy. Moreover, network has a sleep-wake mechanism tuned to the requirement of an application. In this way, the DCB-TDMA MAC protocol has provided energy efficient operation by exploiting large durations of sleep period.
 - In the proposed protocol stack, a provision has been made for addition of new nodes in the network as a future expansion. Similarly, number of nodes can be dynamically removed from various clusters of the network. These provisions have demonstrated a feature of scalability of the network.
 - Bi-directional communication links have been established between Base-Station and every other node in the network via Cluster-Head nodes. A forward link from Base-Station to network nodes has been utilized for transmission of control information while reverse link has been utilized for data collection.
 - In the protocol, selection of Cluster-Head node has been performed on the basis of energy consumption of nodes. When a new Cluster-Head node is selected, the time slots of old and new Cluster-Head nodes are exchanged. A route between a Base-Station and network nodes is also re-established via newly selected Cluster-Head nodes. This feature of Cluster-Head selection has been implemented in the proposed protocol stack. It has been shown that the Cluster-Head selection algorithm ensures energy balancing among the network nodes and prevents possible failure of network due to energy depletion of Cluster-Head nodes.
 - An energy efficient multi-hop time synchronization protocol has been proposed and implemented for achieving a global time scale in the network. This protocol utilizes least number of message exchanges for achieving time synchronization. Hence, the protocol consumes lesser power compared to other regression based time synchronization protocols. It also reduces overall computational cost and processing time.

Various simulation tools have been explored for demonstrating the proposed protocol stack. Initially, analysis of proposed time synchronization protocol has been performed using MATLAB tool. DCB-TDMA MAC protocol has been implemented and demonstrated using SUNSET simulation platform. Finally, a complete cross-layer protocol stack has been implemented on UnetSim simulator. Results of multiple runs of simulation have been studied in terms of

parameters of network performance such as end-to-end delay, throughput, Packet Delivery Ratio (PDR), and channel utilization. Analysis of energy consumption of nodes with the implementation of proposed protocol stack have been provided in the thesis. Feature of energy efficiency of the proposed cross-layer protocol stack has been successfully demonstrated using the simulation tools ensuring a long life-time of the deployed network.

For the systematic verification and validation of UASN, the protocol stack design should be tested on hardware testbed before at-sea deployment. For possibility of such testing, an indoor laboratory based hardware testbed set-up has been developed in this work. A prototype of underwater acoustic sensor node has been built using Commercial Off-The Shelf (COTS) components such as TelosB mote, acoustic modem and power supplies. Number of such underwater acoustic sensor nodes have been used for deploying various topologies of UASN. The proposed protocol stack has been tested using this testbed. The results of deployment in terms of end-to-end delay, Packet Delivery Ratio (PDR) and time synchronization parameters have been analyzed in detail. Overall, the testbed has served as a unique, novel and cost-effective platform for validation of UASN protocols.

Acknowledgements

I take this opportunity to express my gratitude to my supervisor, Prof. K. R. Anupama, for her continuous guidance and motivation. She has extended all necessary facilities and help right from commencement to the conclusion of this research work. Without her constant support and inspiration, this thesis would not have been possible.

I express my gratitude to my co-supervisor, Dr. Lucy J. Gudino, for her guidance and support.

My sincere thank to members of Doctoral Advisory Committee, Prof. G. Raghurama and Dr. C. K. Ramesha for their valuable suggestions and comment.

I express my thanks to Prof. Souvik Bhattacharyya, Vice-Chancellor, BITS-Pilani, Prof. R. N. Saha, Acting Director, BITS-Pilani, K K Birla Goa Campus, Prof. Sasikumar Punnekkat, Former Director, BITS-Pilani, K K Birla Goa Campus, Prof. S. K. Verma, Dean, Academic Research Division, BITS-Pilani, Dr. P. K. Das, Associate Dean, Academic Research Division, BITS-Pilani K K Birla Goa Campus, Prof. M. K. Deshmukh, Head, Dept. of EEE, BITS-Pilani, K K Birla Goa Campus, and members of Doctoral Research Committee of Dept. of EEE, BITS-Pilani, K K Birla Goa Campus for providing administrative support and facilities to carry-out my research studies at the institute.

With great pleasure, I express my appreciation to my project students and co-authors Rohit Sant, Rohit Agrawal, Vishwesh Rege, Pritish Nahar, Rishabh Sethunathan, Mayank Joneja and Laksh Bhatia. I thank all of them for helping me conducting numerous experiments on hardware testbed in laboratory as well as on simulation platforms.

I express my thanks to my colleagues Dr. N. S. Manjarekar and Pravin Mane for their immense help, constructive comments, discussions and moral support. I thank Abhishek Joshi, Vikas Khairnar and Bhushan Kadam for providing their help at different levels in the work.

I thank all my teachers, colleagues, friends and relatives who have helped me either directly or indirectly at different stages of my research work.

I remain indebted to my parents, Chandrashekhar Dhongdi and Megha Dhongdi, for their blessings, moral support and motivation. I thank my grandmother, Maya Deshpande, for her constant inspiration and blessings to me throughout this work. I am thankful to my sister, Kirti Barde, for all the encouragement throughout this endeavor.

From the bottom of my heart, I acknowledge the co-operation, patience and understanding of my wife, Rasika, throughout the course of this work.

I am thankful to God Almighty for bestowing his blessings on me.

Sarang Dhongdi

Contents

Certificate	i
Declaration	ii
Abstract	iii
Acknowledgements	vi
Contents	ix
List of Figures	xiii
List of Tables	xv
Abbreviations	xviii
Symbols	xx
1 Introduction	1
1.1 Wireless Sensor Network (WSN)	3
1.2 Underwater Acoustic Sensor Network (UASN)	6
1.2.1 Introduction to UASN	6
1.2.2 Applications of UASN	8
1.2.2.1 Scientific applications	8
1.2.2.2 Industrial applications	9
1.2.2.3 Defense applications	10
1.2.3 Underwater acoustic communication	11
1.2.4 Major challenges of UASN	13
1.2.4.1 Challenges in design of protocol stack for UASN	14
1.2.5 Categories of UASN	19
1.3 Scope and objectives of the research work	21
1.4 Organization of the thesis	23
2 Literature Survey	25
2.1 Protocol stack development of Underwater Acoustic Sensor Network (UASN) . .	25

2.1.1	Physical layer	26
2.1.2	Data Link Layer	27
2.1.3	Network layer	32
2.1.4	Transport layer	37
2.1.5	Application layer	39
2.1.6	Management planes for UASN	39
2.1.6.1	Localization	39
2.1.6.2	Time synchronization	41
2.1.6.3	Clustering	43
2.2	Cross-layer protocol stack for UASN	44
2.3	Research platforms for sensor networks	46
2.3.1	Development in software for UASN	48
2.3.1.1	Aqua-Sim (UWSN Laboratory, University of Connecticut)	48
2.3.1.2	DESERT (Under project NAUTILUS)	49
2.3.1.3	SUNSET (UWSN group, SENSES Lab, Sapienza University)	49
2.3.1.4	WOSS (Patavina Technologies, NATO Undersea Research center)	50
2.3.1.5	UNET (NUS, Singapore)	50
2.3.2	Development of hardware tools for UASN	50
2.3.3	Testbeds and real-time deployments	51
2.4	Summary	52
3	Design of Network Architecture and Physical Layer for Three-Dimensional UASN	54
3.1	Introduction	54
3.2	Generic three-dimensional UASN architecture	55
3.3	Details of three-dimensional network architecture for UASN - a specific case	58
3.4	Deep water channel of underwater acoustic communication	61
3.5	Design and modeling of physical layer of UASN	72
3.6	Summary	76
4	Development of Testbed for UASN	78
4.1	Introduction	78
4.2	Components of testbed	80
4.2.1	The underwater acoustic node	80
4.3	Software set-up	83
4.4	Different topologies and protocols implemented on UASN testbed set-up	86
4.4.1	Multi-hop communication architecture	87
4.5	Characteristics of testbed	91
4.6	Summary	91
5	Time Synchronization for UASN	93
5.1	Introduction	93
5.2	Background and related work	95
5.2.1	Time in distributed systems	95
5.2.2	Challenges in time synchronization	95
5.3	Time synchronization methods for UASN	97

5.3.1	TSHL (Time Synchronization for High Latency) protocol	98
5.3.1.1	Analysis of TSHL protocol	100
5.3.2	Tri-message time synchronization protocol	102
5.3.2.1	Analysis of Tri-message time synchronization protocol	104
5.3.3	Comparison of TSHL and Tri-message time synchronization protocols	105
5.3.3.1	Resource utilization of TSHL and Tri-message time synchronization protocols	108
5.4	Extension of Tri-message time synchronization protocol for multi-hop topology	109
5.4.1	Analysis of multi-hop Tri-message time synchronization protocol	111
5.4.2	Simulation results of multi-hop Tri-message time synchronization	114
5.5	Implementation of multi-hop Tri-message time synchronization protocol on hardware testbed set-up	115
5.6	Summary	123
6	Dynamic Cluster-Based TDMA MAC Protocol for UASN	125
6.1	Introduction	125
6.2	Dynamic Cluster-Based TDMA MAC protocol for three-dimensional UASN	126
6.2.1	Various types of control information	130
6.3	Pseudo algorithm of DCB-TDMA MAC protocol for UASN	132
6.3.1	Additional features for improvement of network performance	136
6.4	Performance evaluation of DCB-TDMA MAC protocol	137
6.4.1	Maximum size of MAC payload	137
6.4.2	End-to-end delay	139
6.4.3	Throughput	141
6.4.4	Packet Delivery Ratio	142
6.4.5	Channel utilization	142
6.5	Simulation of DCB-TDMA MAC protocol	143
6.5.1	Setting the duration of time slot	144
6.5.2	Setting the power levels of nodes	145
6.5.3	Results and analysis of simulation of DCB-TDMA MAC protocol	146
6.5.4	Number of packets transmitted and received	149
6.5.5	Calculations of delay and energy	151
6.6	Comparison of DCB-TDMA MAC protocol with Basic TDMA MAC protocol	158
6.6.1	End-to-end delay	158
6.6.2	Throughput	160
6.6.3	Channel utilization	161
6.6.4	Energy consumption	162
6.7	Summary	164
7	Cross-layer Protocol Stack Development for Three-Dimensional UASN	166
7.1	Introduction	166
7.2	Description of proposed protocol stack for three-dimensional UASN	167
7.2.1	Network architecture of UASN	167
7.2.2	Physical layer	167

7.2.3	Data Link Layer	168
7.2.4	Network layer	168
7.2.5	Transport layer	168
7.2.6	Application layer	169
7.2.7	Management planes of UASN	169
7.2.7.1	Clustering	169
7.2.7.2	Time synchronization	169
7.2.7.3	Power level management	170
7.3	Implementation of proposed cross-layer protocol stack of UASN on UnetSim simulator	170
7.3.1	UnetStack	170
7.3.2	Deployment of three-dimensional network	171
7.3.3	Physical layer parameters	173
7.3.4	Description of proposed protocol stack in terms of Master cycles	174
7.3.4.1	Setting the duration of time slot	176
7.3.5	Simulation results	177
7.3.5.1	Results of transmission delay	177
7.3.5.2	Results of time synchronization	178
7.3.5.3	Illustration of Data cycle phase	179
7.3.5.4	Results of energy consumption	179
7.3.5.5	Evaluating network performance	182
7.4	Hardware testbed implementation of proposed protocol stack	184
7.4.1	Network deployment	185
7.4.2	Physical layer parameters	186
7.4.3	Working of protocol stack	187
7.5	Summary	189
8	Conclusions and Future Scope	191
8.1	Conclusions	191
8.2	Future scope	196
A	Simple Acoustic Modem (SAM)	197
B	Results of Implementation of Cross-layer Protocol Stack	200
	Bibliography	212
	Publications Based on Present Work	235
	Brief Biography of the Candidate	237
	Brief Biography of the Supervisor	238
	Brief Biography of the Co-supervisor	239

List of Figures

1.1	Example of deployment of Wireless Sensor Network (WSN).	4
1.2	Example of deployment of Underwater Wireless Sensor Network (UWSN).	7
1.3	Protocol stack development of UASN.	14
1.4	Cross-layer solution.	19
3.1	A generic three-dimensional architecture of UASN.	56
3.2	Proposed three-dimensional architecture of UASN.	59
3.3	Transmission Loss (TL) for range of 500 m at water depth = 1000 m.	64
3.4	Transmission Loss (TL) for range of 10 m at water depth = 1000 m.	65
3.5	Transmission Loss (TL) for range of 500 m with operating frequency = 30 kHz.	65
3.6	Transmission Loss (TL) for range of 10 m with operating frequency = 30 kHz.	66
3.7	Transmission Loss (TL) for varying distance of communication at various depth in ocean.	67
3.8	Power spectral density of ambient noise in ocean.	68
3.9	A vertical profile of sound speed in seawater as the function of depth for varying surface temperature.	69
3.10	A vertical profile of sound speed in seawater as the function of depth for varying salinity.	70
4.1	Building an underwater acoustic node using Commercial Off-The-Shelf components (COTS).	81
4.2	Initial version of UASN testbed set-up.	83
4.3	Communication architecture deployed on UASN testbed.	84
4.4	Current version of UASN testbed set-up.	85
4.5	Possible real-time deployment of multi-hop network architecture of UASN testbed.	88
5.1	Characteristics of clock.	96
5.2	Round-trip time of message between Node A and Node B.	98
5.3	TSHL time synchronization protocol.	99
5.4	Phase 2 of TSHL time synchronization protocol with jitter.	100
5.5	Tri-message time synchronization protocol.	103
5.6	Effect of delay on time estimation error in time synchronization protocols.	107
5.7	Effect of jitter on time estimation error in time synchronization protocols.	107
5.8	Effect of time interval after last synchronization on time estimation error.	108
5.9	Extension of Tri-message time synchronization protocol for multi-hop scenario.	111
5.10	Time estimation error for two-hop topology.	115

5.11	Estimation of instant error for multi-hop topology.	116
5.12	Network architecture for testbed implementation of multi-hop Tri-message time synchronization protocol.	116
6.1	Structure of Master cycle of DCB-TDMA MAC protocol.	127
6.2	Simplified network architecture for delay calculation.	140
6.3	Overall energy consumption of nodes after cycle 1, 2 and 3 in DCB-TDMA MAC protocol.	155
6.4	Comparison of average end-to-end delay in DCB-TDMA and Basic TDMA MAC protocols.	160
6.5	Comparison of throughput in DCB-TDMA and Basic TDMA MAC protocols. . .	161
6.6	Comparison of channel utilization in DCB-TDMA and Basic TDMA MAC protocols.	162
6.7	Overall energy consumption of nodes after cycle 1, 2 and 3 in Basic TDMA MAC protocol.	164
7.1	The UnetStack architecture.	172
7.2	Overall energy consumption of BS and CH nodes after cycle 1, 2, 3 and 4.	182
7.3	Node deployment for exploring functionality of vertical link (case 1).	185
7.4	Node deployment for exploring functionality of horizontal link (case 2).	186

List of Tables

1.1	Typical bandwidths of underwater channel.	11
2.1	List of various modems for UASN.	51
2.2	Various testbed implementations and real-time deployments.	52
3.1	SNR calculations for various transmission power levels.	74
3.2	Probability of bit error for different modulation schemes.	75
4.1	Percentage Packet Delivery Ratio (PDR) of multi-hop communication architecture.	90
5.1	Resource utilization for TSHL and Tri-message time synchronization protocols.	109
5.2	Results of implementation of multi-hop Tri-message time synchronization of UASN testbed.	121
5.3	Clock skew and offset values of implementation of multi-hop Tri-message time synchronization on UASN testbed.	122
5.4	Results of multi-hop Tri-message time synchronization under varying sleep durations.	122
5.5	Clock skew and offset values of multi-hop Tri-message time synchronization on UASN testbed with varying sleep durations.	123
6.1	Structure of Control cycle phase of DCB-TDMA MAC protocol.	128
6.2	Structure of Data cycle phase of DCB-TDMA MAC protocol.	129
6.3	Maximum MAC payload in DCB-TDMA MAC protocol for various bit rates.	139
6.4	End-to-end delay on forward link in DCB-TDMA MAC protocol for various bit rates.	140
6.5	End-to-end delay on reverse link in DCB-TDMA MAC protocol for various bit rates.	141
6.6	Average end-to-end delay in DCB-TDMA MAC protocol for various bit rates.	141
6.7	Throughput in DCB-TDMA MAC protocol for various bit rates.	142
6.8	Channel utilization in DCB-TDMA MAC protocol for various bit rates.	143
6.9	Parameters used in simulation of DCB-TDMA MAC protocol on SUNSET simulation platform.	145
6.10	SNR calculations for simulation of DCB-TDMA MAC protocol in SUNSET simulation platform.	145
6.11	Actual time slots of Control cycle phase of second Master Cycle in simulation of DCB-TDMA MAC protocol. (Values of S.T. and R.T. are in seconds.)	147
6.12	Actual time slots of Data cycle phase of second Master Cycle in simulation of DCB-TDMA MAC protocol. (Values of S.T. and R.T. are in seconds.)	148

6.13	Results of power levels and SNR from the simulation of DCB-TDMA MAC protocol.	149
6.14	Energy consumption by BS and CH nodes in cycle 1 in DCB-TDMA MAC protocol.	152
6.15	Energy consumption by CN nodes in cycle 1.	153
6.16	Energy consumption by BS and CH nodes in cycle 2 in DCB-TDMA MAC protocol.	153
6.17	Energy consumption by BS and CH nodes in cycle 3 in DCB-TDMA MAC protocol.	153
6.18	Overall energy consumption of nodes after cycle 1, 2 and 3 in DCB-TDMA MAC protocol.	154
6.19	Control cycle phase of fourth Master cycle in simulation illustrating CH rotation.	156
6.20	Data cycle phase of fourth Master cycle in simulation illustrating CH rotation. . .	157
6.21	Energy consumption by BS and new CH nodes in cycle 4 in DCB-TDMA MAC protocol.	157
6.22	Energy consumption of BS, new CH nodes and old CH nodes at the end of fourth cycle in DCB-TDMA MAC protocol.	158
6.23	Average end-to-end delay in Basic TDMA MAC protocol for various bit rates. . .	159
6.24	Throughput in Basic TDMA MAC protocol for various bit rates.	161
6.25	Channel utilization in Basic TDMA MAC protocol for various bit rates.	162
6.26	Overall energy consumption of nodes after cycle 1, 2 and 3 in Basic TDMA MAC protocol.	163
7.1	Locations of nodes in deployment of three-dimensional UASN.	173
7.2	Parameters for simulation of proposed protocol stack on UnetSim.	173
7.3	Calculations of power levels and SNR values in proposed protocol stack.	174
7.4	Illustration of Control cycle phase of first Master cycle for simulation in UnetSim simulator.	175
7.5	Illustration of Data cycle phase of first Master cycle for simulation in UnetSim simulator.	176
7.6	Result of time synchronization for BS(100) and CH1(200) nodes.	178
7.7	Result of time synchronization for CH1(200) and CN(201) nodes.	179
7.8	Results of Data cycle phase of first Master cycle on UnetSim simulator. (Values of S.T. and R.T. are in seconds.)	180
7.9	Results of energy consumption of BS and CH nodes in first Master cycle of proposed protocol stack.	180
7.10	Results of energy consumption of BS and CH nodes in second Master cycle of proposed protocol stack.	181
7.11	Results of energy consumption of BS and CH nodes in third Master cycle of proposed protocol stack.	181
7.12	Results of total energy consumption of CN nodes in four cycles of proposed protocol stack.	181
7.13	Illustration of Control cycle phase of fifth Master cycle in the implementation of proposed protocol stack in UnetSim.	183
7.14	Illustration of Data cycle phase of fifth Master cycle in the implementation of proposed protocol stack in UnetSim.	183
7.15	Comparison of energy consumption of new CH and old CH nodes after CH rotation.	184
7.16	Results of average end-to-end delay in proposed protocol stack implementation on UnetSim.	184

7.17	Result of protocol stack implementation on the hardware testbed set-up (case 1). . .	188
7.18	Result of protocol stack implementation on the hardware testbed set-up (case 2). . .	189
B.1	Node numbers and addresses in UnetSim simulator.	200

Abbreviations

ACK	Acknowledgment
ARQ	Automatic Repeat reQuest
AUV	Autonomous Underwater Vehicle
BER	Bit Error Rate
BFSK	Binary Frequency Shift Keying
BPSK	Binary Phase Shift Keying
BS	Base-Station
CH	Cluster-Head
CN	Cluster member Nodes
COTS	Commercial Off-The Shelf
CTS	Clear To Send
dB re μ Pa	Decibels reference to 1 micro pascal
DCB-TDMA	Dynamic Cluster-Based Time Division Multiple Access
DIN	Deutsches Institut für Normung (DIN; in English, German Institute for Standardization)
DLL	Data Link Layer
FEC	Forward Error Correction
GPS	Global Positioning System
HDLC	High-Level Data Link Control
ISI	Inter-Symbol Interference
MAC	Medium Access Control
nesC	network embedded systems C language
NACK	No Acknowledgment

NS-2	Network Simulator 2
PDR	Packet Delivery Ratio
RF	Radio Frequency
RS232	Recommended Standard number 232
RTS	Request To Send
p.s.d.	power spectral density
PHY	Physical layer
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SAM	Simple Acoustic Modem
SONAR	SOund NAvigation and Ranging
SPI	Serial Peripheral Interface bus
SUNSET	Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing
TDMA	Time Division Multiple Access
TSHL	Time Synchronization for High Latency
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
UASN	Underwater Acoustic Sensor Network
UNESCO	United Nations Educational, Scientific and Cultural Organization
UnetSim	Underwater Network Simulator
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
UUV	Unmanned Underwater Vehicle
UWSN	Underwater Wireless Sensor Network
WSN	Wireless Sensor Network

Symbols

B	Bandwidth	Hz
B_N	Noise Bandwidth	Hz
c	Speed of sound in water	m/s
d_u	Vertical distance between the successive clusters	m
D	Depth	m
\hat{D}	Packet Delivery Time	s
Delay_{FL}	End-to-end delay on Forward Link	s
Delay_{RL}	End-to-end delay on Reverse Link	s
DI	Directivity index	dB
D_u	Depth of cylindrical column	m
E_b	Energy per bit	J
E_{RX}	Energy consumed in reception	J
ERR_{TSHL}	Time estimation error in TSHL protocol	s
ERR_{TRI}	Time estimation error in Tri-message protocol	s
$E_{TX(PH)}$	Energy consumed in transmission while using power level P_H	J
$E_{TX(PL)}$	Energy consumed in transmission while using power level P_L	J
f	Frequency of operation	Hz
f_d	Doppler shift frequency	Hz
FEC_{CR}	FEC code rate	
FEC_{in}	Input to the FEC block	byte
FEC_{out}	Output of the FEC block	byte

GT	Duration of guard time	s
I	Time interval from last synchronization in TSHL protocol	s
I_i	Power of interference signal from i th path	dB
INT	Time interval from last synchronization in Tri-message protocol	s
I_t	Transmission power intensity	W/m ²
k	Spreading factor	
L_T	Total number of levels in the network	
LXX	L denotes the level number and XX denotes the two-digit number of a node	
MH	Size of MAC header	byte
MP	Size of MAC payload	byte
\hat{M}	Medium access time	s
NL	Ambient Noise Level	dB
N_t	p.s.d. of Noise due to turbulence	dB re μ Pa per Hz
N_{sh}	p.s.d. of Noise due to shipping	dB re μ Pa per Hz
N_w	p.s.d. of Noise due to wind	dB re μ Pa per Hz
N_{th}	p.s.d. of Thermal noise	dB re μ Pa per Hz
N	p.s.d. of Ambient noise	dB re μ Pa per Hz
N_o	Noise power spectral density	dB
pH	Acidity	moles per liter
\hat{P}	Propagation time	s
PD	Propagation delay	s
P_H	Higher power level of transmission	dB
P_L	Lower power level of transmission	dB
P_S	Packet Size	byte
P_b	Probability of bit error	
P_t	Power of transmission	W
r	range (of communication)	m
R	Bit Rate	bps

\hat{R}	Reception time	s
R_u	Radius of cylindrical column	m
RL	Level of received signal	dB
sh	Shipping activity factor	
S	Salinity	ppt
\hat{S}	Software access time	s
S_d	Symbol duration	s
SE	Skew Error	s
SINR	Signal-to-Interference-plus-Noise Ratio	dB
SNR	Signal-to-Noise Ratio	dB
t	time	s
\hat{T}	Transmission time (of packet)	s
T	temperature of water	$^{\circ}C$
TL	Transmission Loss	dB
Tri_Ii	Processing time interval after i th message in Tri-message time synchronization protocol	s
T_S	Duration of one time slot	s
T_{MAX}	Maximum number of slots in the Data cycle including the slots reserved for future expansion.	
w	Wind speed	m/s
α	Clock offset	s
α_t	Absorption coefficient	dB/km
β	clock skew	
δ	jitter	s

Chapter 1

Introduction

Ocean covers nearly 71% of Earth's surface, and so far only a very small percentage of the ocean floor and the global ocean has been explored by researchers. In the past decade, there has been a burgeoning interest among research community to monitor and explore the ocean.

Ocean studies are important not just because the ocean holds various natural treasures, but also because of various factors as follows:

1. Ocean is the dominant physical feature on our planet, it plays an important role in shaping our weather and climate.
2. From tiny microbes to blue whales, the diversity of life in the ocean is astounding. It supports unimagined ecosystems and exotic communities of life.
3. Ocean is a powerful force on our planet, helping to shape the physical features of Earth.
4. Ocean and humans are inextricably interconnected. From providing us with food, energy and mineral resources, and recreation opportunities to holding archaeological clues to the past, ocean affects every aspect of human life. In turn, our actions, from use of resources to pollution or conservation, directly affect the ocean [1].
5. Ocean bed (rocks and sediments) is an archive of information that allows us to unravel Earth's geological processes and history.

Oceanographers and marine scientists wish to explore more of the inner space of the ocean by establishing observatories using modern technologies. Collecting and analyzing real time data continuously using new technologies will help researchers to have a better understanding of the ocean properties, life-processes and events. At the same time, analyzing the data of ocean processes will help us to predict the future climate changes and its impact on human life.

From the scientific and commercial point of view, a continuous real-time, effective and synoptic sampling of ocean has gained huge importance. From the necessity of various possible applications such as ocean environmental monitoring, undersea exploration, seismic monitoring, assisted navigation, tactical surveillance and so on, a new technology of Underwater Acoustic Sensor Network (UASN) has been developed [2–6]. UASN can be considered as a conglomeration of underwater acoustic wireless communication and Wireless Sensor Network (WSN) [7, 8]. WSN deployed underwater is termed as UWSN (Underwater Wireless Sensor Network), and if the acoustic signal is used to form wireless communication links between sensor nodes, then it is termed as UASN (Underwater Acoustic Sensor Network).

WSN deployed on the ground is termed as terrestrial WSN. Though numerous applications of terrestrial WSN have already been conceptualized and implemented successfully, venturing into the underwater domain is not a easy task. There are major challenges in the design and implementation of UASN, most of them because of water as a communication medium and application area. These challenges have gathered huge attention among the research community, and has resulted in various important contributions in the area of UASN. Many new protocols for networking, new effective techniques of communication, hardware designs, simulation platforms, testbeds and tools have been designed. In most of these developments, life-time of the network is a main focus, since the nodes of UASN are usually in-accessible during most of the time of actual deployment. From the network engineer point of view, the life-time of a node can be improved by developing an energy efficient protocol stack for UASN, resulting in optimized use of available battery power for networking and communication. If the deployment is for long duration underwater, then the energy efficiency of the protocol stack becomes a very critical challenge of the research and development.

UASN is a type of network that can be tailer-made to the application of interest. In UASN, depending on the requirement and constraints of application, a suitable architecture and protocol

stack can be developed. The network architectures can be classified as i) two-dimensional or three-dimensional, ii) cluster-based hierarchical or flat, iii) structured or random, iv) static or dynamic, v) fixed or mobile and so on. A proper choice of network architecture makes a huge impact on coverage, scalability, longevity as well as on protocol stack development.

In this chapter, a brief introduction to WSN has been provided in Section 1.1, wherein the important characteristics and applications of WSN have been highlighted. In Section 1.2, a detailed overview of UASN has been provided. Notable applications of UASN, along with major challenges in the design and implementation of UASN have been discussed in this section. In Section 1.3, Scope and objective of the thesis work has been provided. Organization of the thesis has been provided in Section 1.4.

1.1 Wireless Sensor Network (WSN)

The development of low-power, low-cost, small-size multi-functional sensor nodes is already achievable today because of many advances in wireless communication and electronics field. These sensor nodes have sensing capabilities integrated into them along with communication, data processing, computational, and co-operative networking capabilities. Sensor networks are built using a large number of densely deployed sensor nodes. These nodes can be positioned either inside the object or in close proximity to the phenomenon under observation. Ideally the positioning of node should not affect the actual characteristics of the object or phenomenon. The monitoring process should also be non-obtrusive in nature.

The deployment of sensor networks can be i) on the ground, ii) underground, iii) in air, iv) underwater, v) on human bodies, vi) in vehicles, or vii) inside building structures. Sensor nodes are usually scattered in the sensor field in an infrastructure-less ad hoc architecture. Nodes sense and collect data from the field of observation using sensors. They further route this data towards the sink node of the network. Data can be routed to a sink node by direct single hop or via multi-hop communication path. Collection and transmission of data may be i) periodic, ii) threshold based, iii) query-based, iv) event triggered, or v) combination of these techniques. Figure 1.1 shows one such example of WSN, where the sensor nodes are deployed in the sensor field. As shown in this

figure, information collected by the sensor node at the left bottom corner is sent to the sink node using multi-hop wireless communication path.

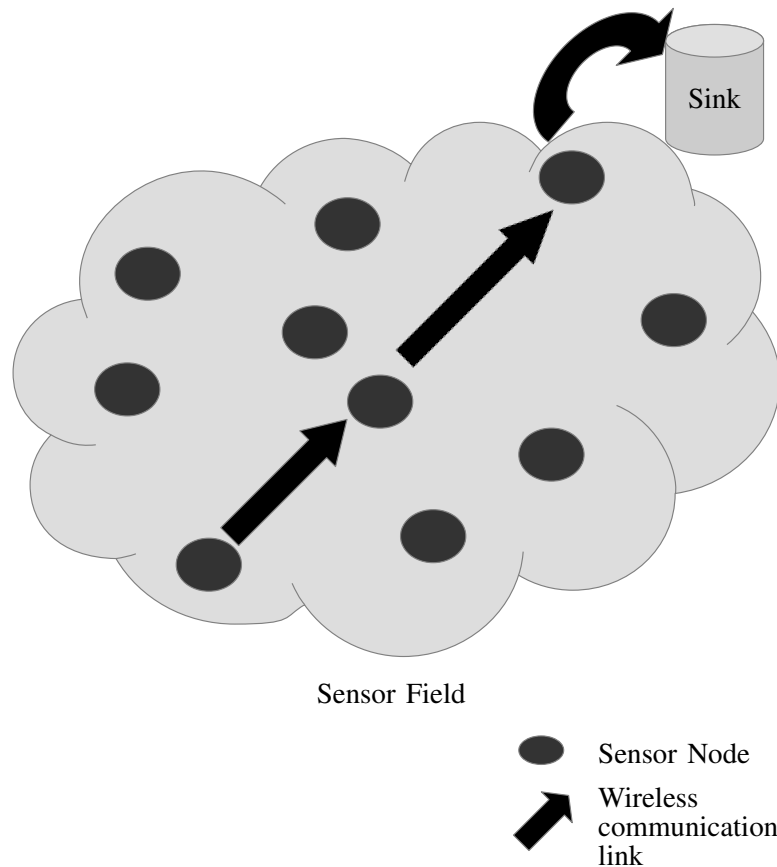


Figure 1.1. Example of deployment of Wireless Sensor Network (WSN).

The nodes of the sensor network can be characterized by their (i) power, (ii) robustness, (iii) algorithm's complexities, (iv) communication range, (v) timing mechanisms, (vi) size, (vii) cost, (viii) flexibilities and (ix) adaptiveness in their architecture [9]. These individual node characteristics need to be properly explored and optimally utilized to build a sensor network as a system. This system is evaluated by using the following parameters such as (i) lifetime, (ii) coverage, (iii) response time, (iv) temporal accuracy, (v) security, (vi) effective sampling rate, (vii) overall cost and (viii) ease of deployment. Factors such as (i) heterogeneity, (ii) distributed processing, (iii) low bandwidth communication, (iv) large scale coordination, (v) optimum utilization of sensors, (vi) real time computation, (vii) fault tolerance under normal and severe environmental conditions are some of the design parameters in sensor networks. Research in WSN aims to meet the above requirements by introducing new design concepts, creating or improving existing protocols, and developing new algorithms. Sensor networks

have limited resources in terms of power, memory (storage space), computational capabilities, and communication bandwidth. Along with these limited resources, environmental conditions and application's requirements pose many design and operational challenges. One of the most important operational challenges is "Energy Efficiency". The design of the sensor network should be done by keeping the energy efficiency as a main focus. Not just the network architecture but the entire protocol stack needs to be critically examined and modified for the energy optimization under the chosen area of implementation, which can substantially help in increasing the lifetime of deployed network.

Based on these characteristics, WSN find wide use in applications such as i) defense, ii) health, iii) environment, iv) home and other commercial areas [10, 11]. In defense applications, WSN can be used for i) monitoring friendly forces; equipment and ammunition, ii) battlefield surveillance, iii) reconnaissance of opposing forces and terrain, iv) targeting, v) battle damage assessment, vi) nuclear, biological and chemical attack detection [12]. Some of the health applications of WSN include i) body area networks [13], ii) tele-monitoring of human physiological data, iii) drug administration in hospitals and so on. Under the category of environmental monitoring, WSN are used for i) precision agriculture [14], ii) habitat monitoring [15], iii) pollution monitoring, iv) forest fire detection, v) flood detection and alert [16], and vi) meteorological or geophysical research. Some of the commercial applications are i) monitoring material fatigue, ii) managing inventory, iii) monitoring product quality, iv) constructing smart office spaces, v) environmental control in office buildings, vi) robot control and guidance in automatic manufacturing environments, vii) factory process control and automation, viii) monitoring disaster area, ix) monitoring structural or machine health, x) machine diagnosis, xi) transportation, xii) factory instrumentation, xiii) local control of actuators, xiv) vehicle tracking and detection, and xv) instrumentation of semiconductor processing chambers. With the ongoing development in WSN field such as development of new sensors, sensor nodes, batteries, network protocols, simulation tools and so on, many new applications are regularly being conceptualized and implemented. One such area of application is Underwater Acoustic Sensor Network (UASN), which is described in detail in the Section 1.2.

1.2 Underwater Acoustic Sensor Network (UASN)

Wireless Sensor Network technology has many important features such as distributed processing, real time computing and communication, large scale coordination and self-organization. This has motivated researchers to use this networking paradigm for underwater applications. Various tasks such as (i) oceanographic data collection, (ii) pollution and environmental monitoring, (iii) offshore oil exploration, (iv) disaster prevention, (v) assisted navigation, (vi) monitoring marine ecosystem, (vi) tactical surveillance applications, and (vii) ocean climate recording can be performed more efficiently using WSN [17, 18].

1.2.1 Introduction to UASN

Underwater Wireless Sensor Network (UWSN) consists of variable number of sensor nodes deployed underwater along with possible use of devices such as gliders, ships, research vessels, underwater instruments, Autonomous Underwater Vehicles (AUVs) (or Unmanned Underwater Vehicles (UUVs)). These sensor nodes and devices usually have two-way underwater wireless communication link to form an UWSN. Various types of sensors perform sensing operation and send the data to a sink node or Base-Station (BS) of the network. This BS, which is usually deployed on the surface of ocean, is further connected to a main control station using RF or satellite link. Such a configuration of UWSN provides a complete real-time interactive environment. A remote observer can monitor, extract and analyze the real time data from specific area of the ocean. It is also possible to re-tune or reconfigure the network by sending control messages from main control station to an individual or a group of sensor nodes of the network. Since data is transferred to the control station when it is available, any loss of data is prevented until a complete failure occurs [3].

An example of UWSN deployment is shown in Figure 1.2. In this UWSN scenario, number of wireless sensor nodes are deployed on the sea-bed and other nodes are deployed at certain depth using stationary mooring. AUVs are also shown as traversing the network area. These sensor nodes and AUVs collect data using available sensors and send this data towards the surface station (or buoy) using multi-hop links. This surface station acts as sink node or BS for the underwater network. The surface station relays the collected information to the on-shore station either directly

using RF link or indirectly via a satellite link. The on-shore station acts as a control station and can issue commands to all the nodes and AUVs of the UWSN and thereby control the deployment scheme, network topology and the data gathering pattern.

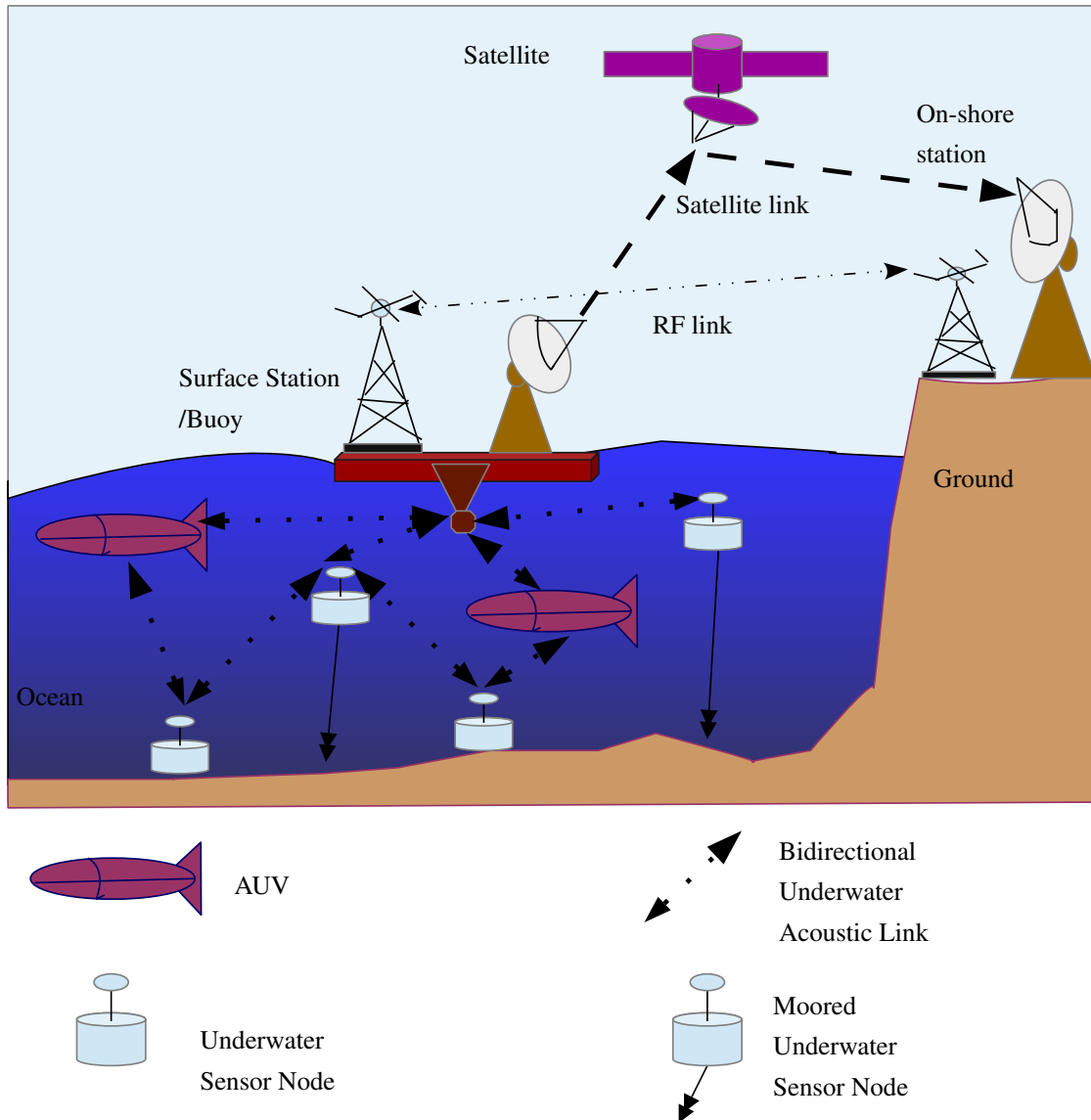


Figure 1.2. Example of deployment of Underwater Wireless Sensor Network (UWSN).

Efficient underwater wireless communication among devices or nodes in UWSN is one of the most fundamental and critical issue in the whole network system design. For wireless underwater communication, various choices of communication carrier such as sound, radio (EM) and optics have been studied in detail in literature [19]. Radio waves, though a preferred carrier for almost all terrestrial wireless communication, cannot be effectively used in underwater communication because they suffer from high attenuation, given the conducting nature of the medium. Optical waves, a carrier with high speed and high bandwidth, finds a limited scope (in short distance

underwater communication) because of absorption effect and scattering of waves due to suspended particles in water. Among the three different carriers, acoustic signal is the preferred carrier for underwater wireless communication systems. Acoustic waves suffer lesser attenuation, interference or scattering when compared to RF or optical waves in underwater. If the wireless communication carrier in UWSN are acoustic waves, then the network is commonly termed as Underwater Acoustic Sensor Network (UASN).

1.2.2 Applications of UASN

In [8, 20], UASN applications are categorized roughly into three classes: a) scientific, b) industrial, and c) defense.

1.2.2.1 Scientific applications

Scientific applications aim to observe ocean environment for various reasons. Some examples of scientific applications are i) geological oceanography, ii) marine biology, iii) ocean chemistry and so on. Geological oceanography is a study of Earth beneath the ocean. Oceanographers study various features such as rises and ridges, trenches, seamounts, abyssal hills and the oceanic crust to observe petrology and sedimentation (clastic, chemical, and biological) processes, erosional processes, volcanism, and seismicity. Since sediments on the ocean floor are relatively undisturbed, they can provide more accurate historical record and state of past climate. In marine biology, researchers study activities of marine animals such as fish, mammals, micro-organisms and so on. It provides better insight and knowledge of marine life, from unknown species and habitats, to migration routes and distribution patterns. The goal of marine biology is to record biodiversity and perform a census of marine life, that is, to obtain quantitative estimates of species distribution and density in the areas being studied. Ocean chemistry is the study of chemical properties of ocean. These properties determine support available for marine life. Pollution levels in atmosphere and ocean needs to be critically monitored since the rising pollution creates impact on the ocean chemistry and in-turn on marine life. For example, rising surface temperatures cause corals to expel symbiotic algae needed for their survival resulting in coral bleaching. These can

easily imbalance the ocean's chemistry and disrupt important biological processes such as food webs and wide-scale marine life productivity.

An example of implementation of UASN for scientific application is the Autonomous Ocean Sampling Network (AOSN) proposed in [21], which is used to observe dynamic ocean fields. Here, several AUVs are used along with distributed acoustic sensors and point sensors for sampling of the high gradients associated with the ocean fronts. AUVs are used as data mules which traverse the network and record temperature, salinity, current velocity and other data. The AUVs then relay key observations to the network nodes in real time and transfer a more detailed set of data after docking at the node. Each network node consists of a base buoy or mooring containing an acoustic beacon, an acoustic modem, point sensors, an energy source and a selectable number of AUV docks.

Another example of ocean monitoring is Coastal Ocean Monitoring and Prediction System (COMPS) [22]. Here, researchers have conducted a coordinated program of observations of the West Florida Continental Shelf (WFS). Interactions between continental shelf and the deep ocean gives rise to ocean circulation and resultant water properties. These ocean currents in combination with certain biological interactions result in evolution of harmful algal blooms. COMPS provides a supportive framework for red-tide prediction as well as for monitoring other coastal ocean phenomenon. For an accurate prediction, the system requires a broad mix of sensors and sensor delivery systems capable of modeling the three-dimensional structure of the velocity and density fields.

1.2.2.2 Industrial applications

Industrial applications involve exploration of minerals, petroleum, natural gas and oil in the deep ocean. These commercial applications require continuous monitoring and control of underwater installations, instruments and equipments such as pipelines. Researchers have used underwater networks for monitoring vibrations along the the Langed pipeline (World's longest underwater gas pipeline) installed at a depth of 800-1100 meters below the ocean surface, on an extremely rocky seabed and rough terrain [23]. In [24], authors provide a survey of various sensor network architectures including UASNs for monitoring underwater pipelines.

1.2.2.3 Defense applications

Defense applications include assisted navigation, monitoring and protection of port facilities, safeguarding ships, tactical surveillance, de-mining and so on. In [25], researchers have used a variety of sensors to detect and classify targets. In [26], researchers have used three-dimensional UASN to form three-dimensional barriers to detect intruding submarine. U.S. Navy Seaweb project is also an example of defense application of large-scale UASN comprising of AUVs, gliders, buoys, and ships [27].

From the requirement of various applications as discussed here, it can be inferred that the traditional approach to ocean observation may not be suitable for the newer applications that have evolved. In the traditional approach, oceanographic sensors are either mounted and towed by the ship or they are deployed for the duration of the monitoring mission to record data. These sensors are then recovered or removed from the ship after the mission. This approach has following disadvantages [3]:

1. **No real-time monitoring** : The recorded data cannot be accessed until the sensors or instruments are recovered. It creates long lags in receiving the recorded information.
2. **No failure detection** : If a failure occurs before recovery, all the data is lost. This can easily lead to the complete failure of mission.
3. **No real-time system configuration** : Adaptive tuning or reconfiguration of the system is not possible in an ad hoc fashion.
4. **Limited storage capacity** : Data to be collected might be limited since capacity of on-board storage devices may be limited.

These disadvantages can be overcome by using UASN which can provide synoptic volume coverage, adaptive sampling, flexible control, energy management and robustness to component failure.

1.2.3 Underwater acoustic communication

Study of underwater acoustic communication is fundamental to the development of UASN. Various important factors affecting underwater acoustic communication are a) Transmission Loss (TL), b) noise, c) multi-path fading, d) Doppler spread, and e) high and variable propagation delay [28]. This section provides a brief summary of these characteristics.

1. **Transmission Loss (TL):** Transmission Loss (TL) is a sonar parameter defined as the accumulated decrease in acoustic intensity when an acoustic pressure wave propagates outwards from a source [29]. The magnitude of acoustic intensity is estimated by adding the effects of geometrical spreading and absorption loss. There are two kinds of geometrical spreading, i) spherical, which characterizes deep water communication, and ii) cylindrical, which characterizes shallow water communication. (In ocean literature, shallow water refers to ocean depth less than 100 m, while deep water is the ocean depth more than 100 m [30]). Spreading loss increases with propagation distance and is not dependent on frequency. On the other hand, absorption loss increases with distance and frequency, thereby imposing a bandwidth limitation on underwater acoustic communication [31].

Table 1.1 shows the typical bandwidth of the underwater acoustic channel for different ranges [32].

Range	Range in km	Bandwidth (kHz)
Very long	1000	<1
Long	10-100	2-5
Medium	1-10	≈ 10
Short	0.1-1	20-50
Very Short	<0.1	>100

Table 1.1. Typical bandwidths of underwater channel.

2. **Noise :** Noise in the ocean can be categorized as man-made noise and ambient noise. Man-made noise is mainly caused by machinery noise, whereas ambient noise refers to hydrodynamics (movement of water including tides, current, storms, wind, and rain),

seismic disturbances, and biological phenomenon. The ambient noise in the ocean can be modeled using four sources: turbulence, shipping, waves, and thermal noise.

3. **High delay and delay variance :** Underwater acoustic propagation speed is a function of temperature, pressure (or depth), and salinity of sea water. The dependence on these functions results in temporal and spatial variability in the speed of sound. Acoustic speed ranges from 1450 m/s to 1550 m/s under varying operating conditions, and is usually centered around 1500 m/s. The relatively slow speed of propagation of sound through sea water ($c \approx 1500$ m/s) is also a factor that differentiates it from electromagnetic propagation ($c_e \approx 3 \times 10^8$ m/s). Because of this large propagation delay (0.67 s/km), UASNs are classified as high latency networks.
4. **Multipath :** In most environments and at the frequencies of interest for underwater communication, the ocean can be modeled as a waveguide with a reflecting surface and ocean floor. Using the ray propagation model, it can be observed that acoustic signals undergo reflections between the ocean surface and ocean floor. Also, acoustic signals suffer refractions because of spatially varying sound speed along the ocean column. These effects create multiple communication paths from source to destination. Multipath effect depends on link configuration, and is a function of depth and distance between the transmitter and receiver. Acoustic links can be categorized as vertical and horizontal links. This classification is according to the direction of sound ray with respect to the ocean floor. Horizontal channels have longer multipath spread as compared to vertical sound channels. Multipath phenomena causes large delay spread, resulting in Inter-Symbol Interference (ISI) due to waveform time-dispersion (or delay spread) [33].
5. **Doppler spread** – The Doppler frequency spread can be significant in underwater acoustic channels, causing degradation in the performance of digital communications. Doppler spread also limits the transmission data rate. Transmission at a higher data rate causes interference of adjacent symbols at the receiver.

Most of the factors stated above are dependent on chemical-physical properties of water medium such as temperature, salinity and density. Since, these properties undergo spatio-temporal variations; behavior of the acoustic channel becomes highly variable temporally as well as spatially.

1.2.4 Major challenges of UASN

Challenges of building UASN are fundamentally different than the challenges of terrestrial wireless sensor network. This is mainly because of the characteristics of underwater acoustic communication channel. Major challenges in UASN design and implementation are as follows [4, 30] :

1. Available acoustic bandwidth is severely limited.
2. Underwater channel is severely impaired, especially due to multi-path propagation and fading.
3. Propagation delay in underwater is significantly higher than in Radio Frequency (RF) terrestrial channels, and extremely variable.
4. High bit error rates and temporary losses of connectivity (shadow zones) can be experienced, due to the extreme characteristics of the underwater channel.
5. Channel impulse response of underwater acoustic channel is spatially as well as temporarily varied.
6. Batteries are energy constrained and cannot be recharged easily (solar energy cannot be exploited underwater).
7. Underwater sensors are prone to failures because of fouling and corrosion.
8. In a harsh underwater environment, it is anticipated that some nodes will be lost over time. Possible risks include fishing trawlers, underwater life, or failure of waterproofing.

Brady and Preisig stated that the acoustic signal is quite possibly nature's most unforgiving wireless communication medium [34]. This complex nature of the underwater acoustic channel and dynamic ocean environment poses many challenges to the network designers. The characteristics of the ocean environment influences medium access control as well as higher layer protocol design. This bears important implications on the design of network architectures and related protocols [35]. Different applications might need entirely different approaches to network design. A single network protocol stack and architecture may not satisfy all the needs.

Although there exists many recently developed network protocols for WSNs, the unique characteristics of the underwater acoustic communication channel require very efficient and reliable protocols to be developed specifically for UASN. Challenges in the design of protocol stack for UASN are described in the following subsection.

1.2.4.1 Challenges in design of protocol stack for UASN

The protocol stack for UASN consists of physical layer, data link layer, network layer, transport layer, and application layer. The protocol stack also contains various network management planes such as i) task management plane, ii) mobility management plane, and iii) power management plane as shown in Figure 1.3. The task management plane balances and schedules the sensing tasks assigned to the sensor field such that minimum required nodes are assigned to sensing tasks and the rest can utilize their energy on routing and data aggregation. Time synchronization protocol can be considered as a network service in task management plane. The power management plane is responsible for minimizing power consumption and it controls the network topology by controlling the sleep-wake pattern. The mobility management plane detects and registers movement of nodes. This plane provides a localization information which is necessary to maintain a data route to the sink. These management planes monitor the power, movement, and task distribution among sensor nodes.

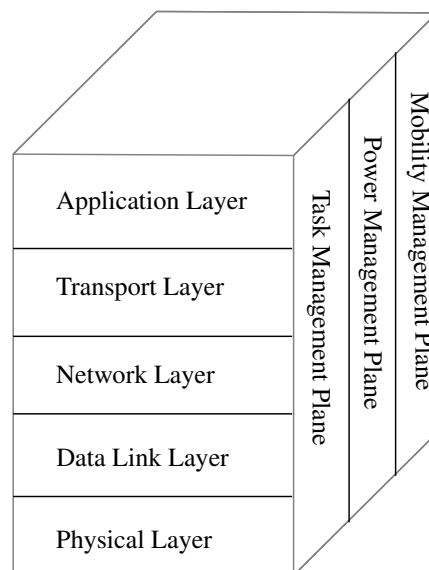


Figure 1.3. Protocol stack development of UASN.

In this subsection, the impact of underwater acoustic communication on the development of protocol stack of UASN has been summarized.

1. **Physical layer** : Transmission Loss (TL) in acoustic communication distinguishes it from the terrestrial wireless communication. TL in underwater acoustic communication depends not only on transmission distance, but also on the signal frequency. This imposes a limit on the useful acoustic bandwidth within the constraints of finite transmission power [31]. As a result of this, shorter communication links offer more bandwidth than a longer communication link. In [8], it is shown that, higher frequencies attenuates quickly over longer distances, prompting most kilometer-range modems to operate below several tens of kHz. This observation suggests that, for a given transmission range, an optimal frequency can be calculated.

Since larger bandwidth is available for smaller distances, it indicates that a greater information throughput is available if messages are relayed over multiple short hops instead of being transmitted over a long hop directly. A greater bandwidth also indicates that higher bit rate and shorter packets can be used. Use of shorter bits implies less energy per bit. Usage of shorter packets implies fewer chances of collision on links with different, non-negligible delays. Both facts have beneficial implications on the network performance (and lifetime), provided that the interference can be managed. Dividing a long link into a number of shorter hops also helps in power reduction similar to terrestrial WSN [36].

Underwater acoustic communication channel shows spatial and temporal variations. It is also severely impaired because of multipath and fading effects. There is a need to develop transmission technique with high-spectral efficiency. Providing energy efficient and robust communication system in this highly dispersive ocean environment is a challenge in the physical layer design. Along with this, there is a pressing need of developing low-cost, low-power, small size acoustic modems.

Design of physical layer along with network architecture has a huge impact on the development of higher layer protocols in UASN.

2. **Data Link Layer** : Data Link Layer (DLL) consists of two sublayers - MAC and LLC. In this work, MAC sublayer of DLL has been studied in detail. DLL is loosely referred to as MAC in the literature. In the design of MAC layer protocols, characteristics of

underwater acoustic channel such as large and variable delay, limited bandwidth, temporary losses of connectivity, high BER, spatial and temporal variations should be considered. In the terrestrial WSN, it can be assumed that receivers can receive the signals almost instantaneously and simultaneously since the speed of communication is high. Even though all receivers are not equidistant from transmitter, the error introduced because of the above assumption is negligible. Such is not the case in underwater networks, with a large transmission time for packets and propagation latency. This leads to uncertainty in obtaining correct channel state (whether channel is busy or idle), termed as space-time uncertainty [37]. Incorrect channel status can lead to poor performance of MAC protocols either by under-utilization or by increase in collisions and (thus) retransmissions.

Challenges in design of UASN MAC protocols are a) estimation of propagation delay to improve the network performance or to achieve synchronization between nodes, b) obtaining a correct channel status under spatio-temporal variations of acoustic channel, and c) scheduling an effective sleep-wake pattern for improving energy efficiency without affecting the working of the protocol.

3. **Network layer** : The function of the network layer is to compute optimal path from source to destination in a multi-hop UASN topology. In a highly dispersive underwater acoustic channel, the link quality is unpredictable. Moreover the disconnections can happen due to issues such as i) node failure, ii) depletion of battery of node, iii) unforeseen mobility due to ocean currents, iv) obstacles, and v) shadow zones. One of the main challenges in network layer design is to provide robustness against channel conditions. It should be an energy efficient design and should also provide latency bounds based on requirements of application.
4. **Transport layer** : In a WSN environment, hop-by-hop reliability of data transfer is a preferable approach than the conventional end-to-end reliability [38]. In UASN, the characteristics of underwater acoustic channel such as high BER, long propagation delay, limited bandwidth makes the design of transport layer a critical issue. Main challenge in the design of transport layer is to correctly infer the cause of packet drops. An energy efficient error recovery scheme needs to be designed for improving channel throughput.

5. **Application layer** : WSN or UASN technologies are designed to realize a particular application. Goal of the application layer is to provide an interface between the user and the network protocol stack to make effective use of underlying resources.
6. **Management planes of UASN** : Different management planes of UASN such as task management, mobility management and power management planes provide various network services such as localization, time synchronization, clustering and so on.

Localization refers to an estimation of geographic locations of sensor nodes. In UASN, nodes do not have any access to effective self-positioning tool such as GPS. A protocol needs to be developed to estimate the co-ordinates of sensor nodes in UASN. It is essential in sensor network to report the sensor data that is geographically meaningful [39]. Localization is additionally helpful in estimating the propagation delay and also to establish a route from source to destination.

Time synchronization refers to achieving a global scale of timing among sensor nodes of network [40]. Time synchronization algorithm is necessary to synchronize all sensor nodes in the network to a common global time, which can help to achieve an effective sleep-wake schedule. Time synchronization also helps in providing accurate timing information for the sensed data. In the design of scheduled based MAC protocols, time synchronization is essential.

Clustering is essentially grouping of network nodes into manageable sets [41]. A clustering protocol can help exploit the advantages of topology such as multi-hop connectivity. Clustering protocol helps in improving energy efficiency and hence can prolong the life-time of a network. It can additionally help in developing an effective data aggregation policy, achieving low latency, balancing traffic load and in improving system capacity by enabling bandwidth reuse.

Challenges in the design of these protocols are a) robustness, b) energy efficiency, c) complexity, d) scalability and e) obtaining desired precision in a complex underwater acoustic environment.

Traditionally, networking solutions have used simple and modular layered architecture for the development of network protocol stack. The modular approach is generally used in Internet

applications. It has helped developers to design and implement different protocols at various possible layers (application, transport, network, data link and physical layer). Use of standardized interfaces between layers ensures the compatibility of various protocols.

Sensor networks have severe resource constraints when compared to Internet based applications (which uses traditional layered approach). Hence, protocol stack that have large memory footprints are not desirable. Similarly, interfaces that require a large amount of processing between layers cannot be supported [42]. Application aware communication protocols that maximize the overall network performance and minimize the energy consumption are the primary requirement of WSNs. Considering these peculiarities such as scarce resources, wireless channel conditions, low-power transceivers, and specific requirements of application, researchers have proposed the joint optimization and design of networking layers, termed as cross-layer protocol stack. In a cross-layer protocol stack implementation, functionalities of two or more layers can be combined to form a single coherent framework. Recent studies on WSNs reveal that cross-layer integration and design techniques result in significant improvement in terms of energy conservation and stands as most promising alternative to inefficient traditional layered protocol architectures.

Specifically in the harsh environment like underwater networks, it is advocated to integrate highly specialized communication functionalities [30] to improve network performance and to avoid duplication of functions by means of cross-layer design. This also allows improving and upgrading particular functionalities without a requirement to redesign the entire communication system. Cross-layer implementation in UASN can help to achieve better adaptability to the characteristics of an underwater channel and lead to flexible Quality of Service (QoS).

Figure 1.4 shows one of the possible conceptual implementation of cross-layer solution along with management planes for WSN/UASN. In this module, several networking functionalities are unified in single functional module. Higher layer functionalities such as routing, medium access, congestion control can utilize the information from the physical layer (such as current channel condition) to make their operations channel-aware.

Design of protocol stack is closely related to the network architecture. Network architecture of UASN can be categorized into various classes depending on type of deployment, communication architecture and topologies. Various categories of UASN are briefly described in Subsection 1.2.5.

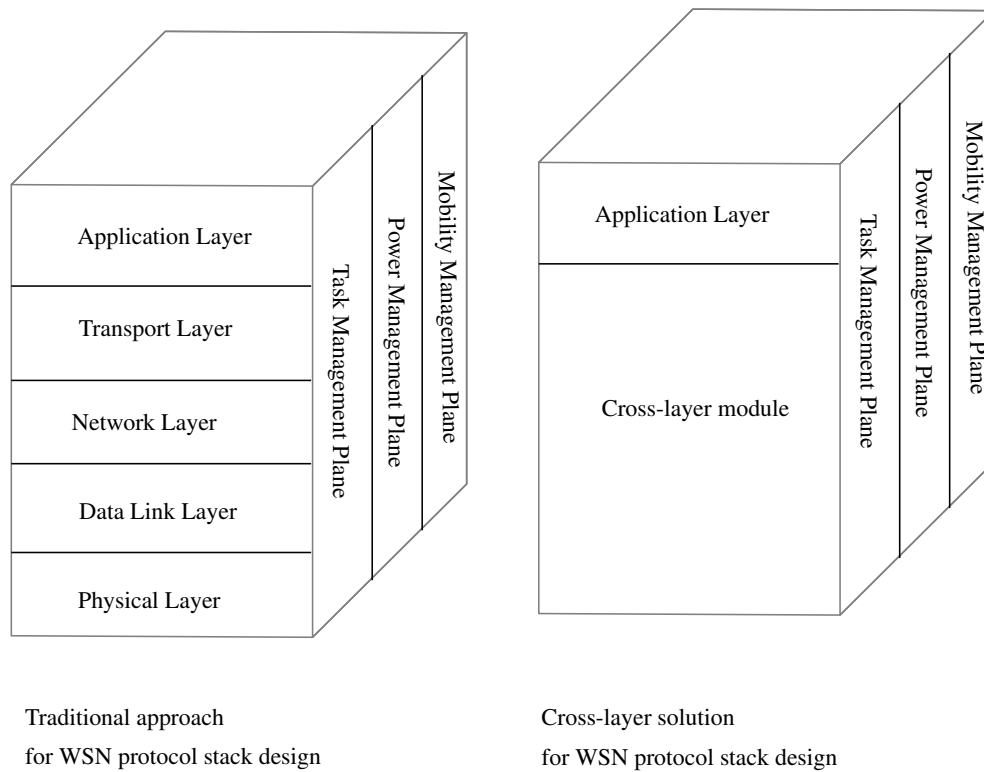


Figure 1.4. Cross-layer solution.

1.2.5 Categories of UASN

UASN Network architecture can be classified based on various parameters such as:

1. Deployment pattern :- Based on the pattern of deployment, UASN can be classified as either two-dimensional or three-dimensional. In the two-dimensional architecture, sensors are anchored to the bottom of the ocean. In the three-dimensional architecture, sensors float at different ocean depths covering the three-dimensional region. Three-dimensional architectures are useful for detecting and observing phenomena in the three-dimensional ocean column space [43].
2. Depth of deployment :- Based on the depth of deployment, UASN can be classified as shallow water UASN and deep water UASN. Networks deployed in water depth less than 100 m are termed as shallow water UASN, whereas networks deployed in deep water (beyond 100 m depth) are termed as deep water UASN.

3. **Mobility** :-Based on mobility, UASN can be classified as static or mobile networks. In a static network, sensor nodes are anchored to sea floor. Some of the sensor nodes are placed on floating buoys on ocean surface. These nodes are usually static after initial deployment. If the node drift because of the ocean current, it induces a passive mobility in the network. On the other hand, in mobile networks, mobility is induced by using AUVs or UUVs equipped with sensors. Depth adjustment of nodes can also be done using certain mechanism introducing mobility in the network.
4. **Topology** :- UASN can have either random or structured topology. In random topology, network nodes are deployed in random fashion. In structured topology, nodes are deployed based on a pre-determined pattern. In structured topology, sensor nodes are usually aware of their locations and neighbors.
5. **Hierarchy** :- UASN can have flat network architecture or a multi-tiered cluster-based network architecture.
6. **Hop-count** :- UASN can be single-hop (where every node can communicate directly with the sink) or multi-hop network (where data is forwarded via a multi-hop path to the sink).
7. **Duration of deployment** :- Based on the duration of deployment, UASN can be classified as short term or long term network. A short term network is usually deployed for few days or a week. Long term networks may be deployed for periods of several months to years.
8. **Range** :- Based on the range of communication and coverage, UASN can be categorized as large and small networks.
9. **Node density** :- Based on number of nodes per cubic meter, UASN can be classified as dense or sparse networks.
10. **Types of sensor nodes** :- In UASN, if all the sensor nodes have identical hardware structure, energy sources and network capabilities, then the network is termed as homogeneous network. If the nodes have different functionalities, then the network is termed as a heterogeneous network.

In this work, a problem of designing three-dimensional UASN has been considered. This three-dimensional UASN can be used for various applications in the field of oceanography such as

pollution monitoring or ecosystem observation of ocean column. The proposed three-dimensional network architecture is static, structured, sparse, homogeneous, multi-tiered and multi-hop in nature. The deployment field is assumed to be deep water ocean environment.

Underwater acoustic sensor nodes have limited battery which can not be easily replaced or recharged. For longer lifetime of the network, an energy efficient protocol stack has to be developed for the three-dimensional UASN. Considering a long-term ocean column monitoring application, an energy efficient cross-layer protocol stack has been proposed and developed in this work.

In the following section, the scope and objectives of the proposed research work have been provided.

1.3 Scope and objectives of the research work

Objective 1 - To study the characteristics of underwater acoustic channel.

In order to design and develop an UASN, it is important to understand the communication model of underwater acoustic channel. This particular topic has been extensively studied in the past, mostly with an aim to establish point-to-point communication. These studies and experiences have helped researchers to consider effective network design for UASN. Extensive literature survey has to be conducted on the topic of underwater acoustic communication to understand its impact on the network design.

Objective 2 - To develop a network architecture for ocean column monitoring.

Three-dimensional UASN can be designed for ocean column monitoring which is useful for various applications in oceanography such as pollution monitoring or marine ecosystem observation. A static and structured three-dimensional network architecture is to be conceptually developed to meet the requirements of application. Simplicity and scalability are important parameters for the design.

Objective 3 - To develop the testbed set-up for implementation of UASN.

Development of an indoor laboratory based testbed set-up for implementation of autonomous Underwater Acoustic Sensor network is one of the objective in the research work. For this testbed, an underwater acoustic node has to be developed using components such as underwater acoustic modems, TelosB motes and power supplies. Various UASN protocols and topologies can then be implemented on this hardware testbed set-up. It can also be used for validation of results obtained using analytical and simulation approach.

Objective 4 - To develop and implement a lightweight time synchronization protocol for UASN.

Sensor networks have to perform many collaborative tasks by effective co-ordination among all the nodes. For this, a common time-base across the network is required. The process of achieving and maintaining a common time-base is termed as time synchronization. The objective here is to develop and implement a simple, modular, and energy efficient (lightweight) multi-hop time synchronization.

Objective 5 - To develop an energy efficient protocol stack on three-dimensional UASN.

One of the main objective of the research work is to develop a complete protocol stack for three-dimensional UASN consisting of all five layers of protocol stack along with network management protocols such as time synchronization, Cluster-Head selection, and power management protocol. Energy efficiency is the main focus of the cross-layer protocol stack design for the longevity of the network.

Objective 6 - Implementation of three-dimensional UASN protocol stack on simulation platform and hardware testbed set-up.

Energy efficient protocol stack developed theoretically can be implemented and tested on hardware testbed set-up. In order to verify the scalability of the protocol stack, a larger topology can be implemented on a simulation platform. Working of the protocol stack can be analyzed in detail using these available tools.

Organization of the thesis has been described in the Section 1.4.

1.4 Organization of the thesis

The remainder of the thesis has been organized as follows:

Extensive literature survey on UASN has been covered in Chapter 2. Brief information of various protocols proposed and developed by researchers for different layers as well as for network management planes of protocol stack of UASN has been provided. Recent developments of cross-layer protocol stack for UASN has been explored. Development of hardware and software simulation tools, laboratory based testbeds have been covered in this survey. Efforts of laboratories around the world in practical deployment of underwater networks have also been described in this chapter.

In Chapter 3, a three-dimensional UASN architecture has been proposed for the ocean column monitoring application. Motivation and assumptions of this network architecture have been stated clearly. The deep water acoustic channel model has significant impact on the design of protocol stack for this network architecture. Hence, the characteristics of deep water acoustic communication have been discussed in this chapter. Based on the study of this underwater acoustic channel, the design and modeling of physical layer for the network architecture has been proposed.

In Chapter 4, details of development of laboratory based testbed set-up have been provided. Various network topologies and protocols tested on this testbed has been listed in this chapter. Details of testing of multi-hop communication architecture has been detailed.

Requirement of achieving and maintaining time synchronization in sensor networks has been discussed in Chapter 5. The challenges of achieving time synchronization in underwater network have been provided. A multi-hop time synchronization protocol has been proposed in this work for UASN. Performance of the proposed protocol has been evaluated and compared with other prominent protocol using MATLAB tool. This protocol has also been tested on the hardware testbed set-up. Detailed results of simulation and hardware testbed implementation have been provided in this chapter.

In Chapter 6, a new variant of TDMA based MAC protocol termed as DCB-TDMA protocol has been proposed for the three-dimensional UASN. This protocol involves provision of Cluster-Head selection, dynamic routing and power level management. Number of nodes can be dynamically

added or removed from the network providing a feature of scalability. The evaluation of network performance parameters such as Packet Delivery Ratio (PDR), end-to-end delay, throughput, channel utilization and energy consumption has been performed. The proposed protocol has been implemented using SUNSET, an open source simulation platform developed by SENSES lab, Sapienza University, Rome. The details of this protocol, results of simulation runs, and analysis of the results have been provided in this chapter.

A complete cross-layer protocol stack has been proposed for the three-dimensional UASN in Chapter 7. This protocol stack includes functionalities of all five layers (that is, physical layer, data link layer, network layer, transport layer and application layer) along with protocols for network management such as Cluster-Head selection, time synchronization, and power level management. The proposed protocol stack has been implemented on the UnetSim, an open source simulation platform developed by ARL laboratory at National University of Singapore, Singapore. The results of implementation has been analyzed from the perspective of functionality of various layers. It has been demonstrated that the proposed cross-layer protocol stack is energy efficient and hence suitable for a long-term deployment. This protocol stack has also been implemented on the hardware testbed set-up. Results of this hardware testbed implementation have been described in detail in this chapter.

In Chapter 8, brief summary of work and future scope of the research has been discussed.

Chapter 2

Literature Survey

Underwater Acoustic Sensor Networks are becoming increasingly important, with numerous applications emerging from vast areas such as commercial, scientific and defense. In this chapter, a comprehensive review of the current research in UASN has been provided.

The organization of chapter is as follows: In Section 2.1, a review of research and development of protocols at various layers of networking modules, namely physical, data link, network, transport and application layer has been given. Protocols developed for implementing network management planes of UASN has been reviewed briefly in this section. In Section 2.2, research and development of cross-layer protocol stack approach for UASN has been summarized. Development of various hardware tools, software simulation platforms and testbeds developed by prominent universities and research organizations has been described in Section 2.3. Brief summary of the chapter has been provided in Section 2.4.

2.1 Protocol stack development of Underwater Acoustic Sensor Network (UASN)

Network designer has to choose an appropriate type of network architecture based on the requirement of applications and availability of resources at hand. A still larger challenge is in terms of protocol stack development. Ideally, the network protocol stack is expected to be channel-aware,

application-aware leading to optimal network performance with efficient use of resources. The survey of various protocol design and development at different layers of UASN protocol stack is briefly described in the following sections [44].

2.1.1 Physical layer

While setting up the underwater acoustic network, a robust and spectrally efficient communication technique is essential. Given the unique characteristics of underwater acoustic channel, research in the area is very active. Choosing optimum parameters of the physical layer such as carrier frequency, modulation technique, bandwidth is very important, since these parameters has a huge impact on the design of higher level protocols of UASN [35, 45].

The underwater acoustic communication channel is highly dispersive. It is very difficult to achieve phase tracking. Because of this reason, initial efforts in underwater acoustic research (WHOI micro-modem [46]) or modem development (commercial, Teledyne-Benthos [47] has been concentrated on robust communication using FSK (Frequency Shift Keying) and non-coherent detection [8]. These non-coherent schemes have algorithmic simplicity and power efficiency. FSK based modulation schemes can suppress multi-path effects with the use of time guards between successive pulses. Disadvantage of these schemes are that they are not bandwidth efficient and are restricted to low bit rate applications.

Long range, high throughput systems were later developed with the help of advances in digital signal processing and availability of coherent modulation schemes such as PSK (Phase Shift Keying) and QAM (Quadrature Amplitude Modulation) in acoustic communication. In [48], the performance of coherent schemes is studied with use of adaptive Decision Feedback Equalizers (DFE). Results indicate the superior quality of reception with the use of joint equalizers. Partially coherent techniques, such as DPSK (Differential Phase Shift Keying) are also available, which alleviate the need of carrier phase tracking, but at higher probability of error as compared to PSK.

One of the promising solution for underwater communications is Orthogonal Frequency Division Multiplexing (OFDM) spread spectrum technique, which is particularly efficient when noise is spread over a large portion of the available bandwidth [49]. It proves to be an effective technique for combating the multipath delay spread without the need for complex time-domain equalizers.

In [50], authors provide the overview of signal processing techniques for underwater acoustic communication, including advances in single-carrier, multi-carrier modulation/detection system, iterative equalization, and space-time coding techniques. Both types of systems (single-carrier and multi-carrier) are being extended to multi-input multi-output (MIMO) configurations. These configurations can provide spatial multiplexing, that is, the ability to send parallel data streams from multiple transmitters. It has been demonstrated experimentally that, bit rates of several tens of kbps can be achieved. In [6, 51] authors have presented an acoustic MIMO-OFDM system as a low-complexity solution for spectrally efficient communications over bandwidth-limited frequency-selective underwater channels.

Various commercial modems have been developed with the use spread spectrum technique of DSSS (Direct Sequence Spread Spectrum) [52–54]. Due to the spreading operation, the data rates are often in the order of hundreds of bps while using bandwidth in the range of several kHz.

2.1.2 Data Link Layer

Medium Access Control (MAC) protocol of the data link layer is essential for controlling and managing the shared communication medium among all the nodes in the network. MAC protocol used in wireless sensor network provide self-organizing capabilities to the network by providing necessary infrastructure for hop-by-hop wireless communication [42]. Important attributes of MAC layer are energy efficiency, scalability, network throughput, fairness, latency, and bandwidth utilization. An efficient MAC protocol can ensure energy saving (by possibly providing sleep-wake patterns, collision avoidance mechanism and so on.), high network throughput and low channel access delay.

There are various MAC protocols developed for the wireless ad hoc and sensor networks. Authors Jurdak et.al., Kumar et.al., have presented an exhaustive survey of MAC protocols for wireless ad hoc networks [55, 56] whereas authors Demirkol et.al., Kredo et. al. have published a survey of MAC protocols for sensor networks [57, 58]. These protocols work well for the terrestrial RF wireless networks but cannot be utilized for underwater networks because of the typical characteristics of underwater medium, prominently the problem of limited bandwidth, long and variable propagation delay, multipath effect, and channel asymmetry [7].

Researchers have developed MAC protocols for UASN, which can be classified into two main categories as follows:

1. Contention free protocols (FDMA, TDMA or CDMA based protocols)
2. Contention-based protocols. Contention based protocols can be further categorized into protocols based on random access method and those based on collision avoidance methods.

Among contention free (or schedule based) protocols, FDMA based MAC protocols are not suitable for UASN since the available bandwidth in UASN is very limited. FDMA needs to further divide the band into narrower sub-bands for allocation to individual node. The narrow bandwidth channel of sub-bands would be more vulnerable to fading and multipath phenomenon.

Researchers have developed many TDMA centric MAC protocols (STUMP [59], DSSS-TDMA [60], C-MAC [61], USS-TDMA [62], I-TDMA [63], Super-TDMA [64]).

In STUMP (Staggered TDMA Underwater MAC protocol [59]), authors have designed a two-dimensional network to gather data and forward it to a remote user through a single gateway node. STUMP uses propagation information to schedule overlapping transmissions that do not collide at the receiver. Allowing overlaps in communication is the key for improved performance. By segmenting the area surrounding a node into logical rings, finer scheduling is achieved using a distributed algorithm. STUMP is well suited for periodic, high data rate application but the trade-off for improved performance is the overhead incurred in schedule generation.

DSSS-TDMA [60] is another TDMA-based protocol wherein concurrent transmissions are allowed to increase channel utilization. A Dynamic Slot Scheduling Strategy (DSSS) is used to schedule each transmission pair. In this protocol, slot size changes with varying number of nodes. More the nodes, smaller is the slot size. This would lead to a significant decrease in channel utilization, as slot size may not be sufficient to complete transmission in an overloaded network.

Another TDMA based MAC protocol, which is derived from cellular network concept is C-MAC (Cellular-MAC) [61]. In this protocol, the network is divided into logical hexagonal cells. Channel utilization is improved with the help of collision-free slot assignment and avoiding the need of control packets. But this paper does not explore the possibility of channel-reuse though it is basic

foundation of cellular networks. It also suffers from the large latency needed due to the decision making regarding time slots assignment in the case of large network.

In USS-TDMA (Underwater Self-Stabilizing TDMA) [62], a cube grid topology is assumed with each node being aware of its location. A simple routine based on diffusion computation is utilized to validate the slots assigned to nodes in the network. The Base-Station periodically sends a diffusion message in its time slot. Receiving node recalculates time slot based on an algorithm and forwards the message in the recalculated time slot. Energy efficiency is achieved by using collision-free schedule assignment and effective sleep-wake pattern. This protocol currently does not provide a solution to the probable failure of initiator node (of diffusion computation).

Authors Zhong et.al. have proposed I-TDMA (Interleaved TDMA) [63], for Underwater Acoustic Sensor Network. In this protocol, channel utilization and network performance is improved by use of interleaving data packets in empty time axis by taking advantage of large propagation delay. Limitation of this protocol is that it is only suitable to equal distance three node network.

Authors Anjangi et.al. have presented Super-TDMA MAC protocol [64], in which the practical constraints of the acoustic modem has been included in the design. This protocol focuses on the computation of optimal packet and time slot length which helps in minimizing the guard times and maximizing utilization of time slot. Experimental results demonstrating the Super-TDMA protocol has been shown recently using three node network in Singapore waters [65].

TDMA based MAC protocols have several advantages in terms of simplicity, fairness and energy efficiency. Collisions, idle listening and over-hearing can be avoided in these protocols. The hidden node problem is easily solved without using extra message overhead because neighboring nodes transmit at different time slots. There are also certain disadvantages associated with TDMA based MAC protocols such as poor channel utilization, scalability on dynamic basis and the requirement of time synchronization.

Most researchers prefer CDMA based approach for designing MAC protocols such as [66, 67]. Large bandwidth channels of CDMA are quite robust to frequency selective fading caused by multi-paths in underwater networks. CDMA provides better performance with the possibility of coherently combining the multipath arrivals by employing Rake filters at the receiver. According to [2], CDMA and spread-spectrum signaling appear to be a promising multiple access technique for

shallow water acoustic networks. Disadvantage of CDMA is that the system capacity is restricted by the presence of multiple access interference (MAI) in the system. CDMA also suffers from the issue of Near-Far effect.

The first MAC protocol leveraging the properties of CDMA is proposed in [68]. Here, authors have proposed a distributed MAC protocol termed as UW-MAC, based on CDMA scheme. In this protocol, the code and transmit power are distributively selected by each sender, making it a fully distributed solution.

Authors Tan et.al., [66] have proposed PLAN, (Protocol for Long-latency Access Networks), a distributed MAC protocol which utilizes CDMA as the underlying multiple access technique. In this protocol, CDMA is used to minimize multipath and Doppler effects which are inherent in underwater physical channels. The proposed MAC protocol also involves a three-way handshake (RTS-CTS-Data), which collects the RTS from multiple neighboring nodes before sending a single CTS.

Among contention based techniques, various MAC protocols for underwater networks have been developed with aim of reducing or avoiding packet collisions. MAC protocols such as PCAP (Propagation-Delay-Tolerant Collision Avoidance Protocol) [69], APCAP (Adaptive Propagation-Delay-Tolerant Collision Avoidance Protocol)[70], DACAP (Distance Aware Collision Avoidance Protocol) [71] are based on handshaking mechanism with the requirement of RTS/CTS exchanges. Slotted-FAMA [72], is also another handshaking based MAC protocol wherein the RTS/CTS can only be transmitted at the beginning of a time slot. Various MAC protocols are also based on Multiple Access Collision Avoidance (MACA) scheme such as MACA-MN (MACA-Multiple Neighbors) [73], MACA-U (MACA for Underwater) [74], MACA-DT (MACA-Delay Tolerant) [75].

In recent years, authors Liao et. al have proposed Spatially Fair MAC Protocol (SF-MAC) [76], where the receiver waits to collect RTS messages from all the contenders during the RTS contention period and decides the transmission order accordingly. It can avoid collision by postponing the CTS by a time frame equal to RTS contention period. Authors Ng et al., [77] have proposed an asynchronous, single-channel handshaking based MAC protocol using Reverse Opportunistic Packet Appending (ROPA). It uses a novel approach that exploits the idle waiting time during a two-way handshake to set up concurrent transmissions from multiple nodes.

Contention based techniques which rely on handshaking mechanisms such as RTS/CTS are inefficient in underwater networks [4], because of the large and variable propagation delay. In situations where the distance is long and the propagation delay is large, it is possible for a node to sense the channel as idle inadvertently, even though the communication is ongoing, since the signal may not have reached the receiver yet. This can itself lead to more collisions and a drop in network efficiency. On the other hand, protocols which prefer time slot allocation, requires the slot to be larger than the maximum propagation latency of the deployed network, which again reduces throughput. This led to lot of analysis and research centered around simple ALOHA based MAC protocols for underwater networks such as ALOHA-CA (ALOHA with Collision Avoidance), ALOHA-AN (ALOHA with Advanced Notification), UW-ALOHA (UnderWater ALOHA), SA-ALOHA (Synchronized Arrival Slotted-ALOHA), ISA-ALOHA (Improved Synchronized Arrival Slotted ALOHA) [37, 78–82]. To improve the throughput in Aloha, it is important to reduce number of collisions (ALOHA-CA) or to reduce the unproductive transmission (ALOHA-AN). It can be stated that ALOHA protocols are suitable for low rate and random traffic scenario, but suffer from poor energy efficiency due to packet collisions and have limited improvement in overall throughput.

There are other MAC protocols for UASN which are based on policy of reservation R-MAC [83], RCAMAC [84], HRMAC [85], GOAL [86], UWAN-MAC [87], T-Lohi MAC [88], RIPT [89].

In R-MAC (Reservation based MAC) [83], transmission of control packets and data packets is scheduled at both the sender and receiver to avoid data packet collision completely. It supports fair sharing of the channel among the competing senders.

RCAMAC (Reservation Channel Acoustic Medium Access Control) [84] seeks to improve channel utilization with the introduction of a separate (control) channel for RTS/CTS transmissions. By transmitting short RTS packets on an orthogonal low bandwidth control channel, utilization of the majority-bandwidth main channel is maximized by minimizing the probability of data packet collisions.

In HRMAC (Hybrid Reservation-based MAC protocol) [85], node uses a reservation scheme to reserve the channel and spectrum spreading technology to reduce the collision of the control packets.

GOAL (Geo-rOuting Aware MAC protocol) [86] is the geo-routing and reservation based MAC protocol, which uses self-adaptation based REQ/REP handshake and carrier sensing to perform combined channel reservation and next-hop selection.

In [87], a distributed MAC protocol termed as UWAN-MAC (Underwater Wireless Acoustic Network-MAC) has been proposed with an objective of reducing the number of collisions to achieve higher energy efficiency. Each node schedules the time for transmitting next packet and broadcasts it by augmenting the current packet with this information. By this method, all the nodes are made aware of the subsequent communication and can follow an effective sleep-listen cycle.

T-Lohi (Tone-Lohi) [88] is a tone-based contention protocol for single-hop underwater networks. In T-Lohi protocol, time is divided into frames consisting of the Reservation Period (RP) and the data period. RP is further partitioned into Contention Rounds (CRs). Each node with a data packet first transmits a tone in a CR. If no other tones are received from neighbors, the node is successful in channel reservation, and therefore it can transmit the packet in the following data period. By exploiting space-time uncertainty and high latency to detect collisions and count contenders, T-Lohi protocol achieves high throughput and reduces energy consumption.

RIPT (Receiver Initiated Packet Train) [89], a random access handshaking based protocol uses receiver initiated reservations in a multi-hop underwater acoustic network. It coordinates the arrival of multiple packets at receiver in packet-train fashion.

Large variety of available MAC schemes on UASN shows that the research in this area is very active. Goal of MAC protocol development is to achieve high throughput, low delay in channel access, low energy consumption and fairness among all network nodes.

2.1.3 Network layer

Sensor networks generally use multi-hop paths to the sink. It is the function of network layer to compute the most suitable route from a source to destination in this multi-hop communication network. Suitability of the route depends on metrics chosen for route optimization. Different possible metrics in wireless networks are number of hops, distance of the route, congestion on route, delay, or a combination of these metrics. Intensive study of routing protocols for ad hoc

networks [90] and for wireless sensor networks [91] shows that the existing routing protocols can be divided into three categories, namely proactive, reactive and geographical routing protocols.

- Proactive routing protocol – DSDV (Destination-Sequenced Distance-Vector) [92], OLSR (Optimized Link State Routing) [93] are routing protocols that use proactive routing schemes. The very aim of these protocols is to minimize delay in the process of ‘route discovery’. Though these protocols might be suitable for delay sensitive or real time applications, the trade off is the requirement of maintaining and dynamically updating the routing information. The routing table of each node maintains the information of every route available to every other node in the network, making the establishment of the path a very quick process. In densely deployed large scale WSN/UASN, there is no specific need of finding and maintaining a route from one node to all other nodes of the network. Communication bandwidth and storage capacity required for storing and updating the routing information is also not available in WSN or UASN networks.
- Reactive routing protocol – AODV (Ad hoc On Demand Distance Vector) [94], DSR (Dynamic Source Routing) [95] are the primary examples of reactive routing protocols. In contrast to proactive routing protocols, the ‘route discovery’ process starts in reactive routing protocol only when the node requires it. Route discovery is done using flooding initiated by source node. Once the route is established, it is maintained till it is needed by the source node. Since these protocols rely on flooding and consume a lot of energy and bandwidth, they are not suitable for UASN.
- Geographical routing protocol – GFG (Greedy-Face-Greedy) [96], PTKF (Partial Topology Knowledge Forwarding) [97] are among the geographical routing protocols wherein the route is established with the help of location information. It needs the position of its neighbors and of the destination node to be available in order to establish a route from source to destination. Location information can be obtained using localization protocol which further depends on fine grained time synchronization protocol. In case of UASN, maintaining localization and time synchronization is very challenging task because of typical characteristics of the underwater channel and the non-availability of GPS.

For the network layer, many protocols have been developed by researchers [98–106]. Depending on the main feature of design of these protocols, they can be roughly categorized as i) geographical (or location based) routing protocols VBF [98], HH-VBF [99], FBR [100], and routing protocol for delay-sensitive and delay-insensitive applications [101], ii) flooding based protocols such as DFR [102], iii) diffusion based protocols such as UWD [103], and iv) cluster-based protocols such as DUCS [104, 105].

In [98], a Vector Based Forwarding (VBF) protocol has been proposed that attempts to provide a robust, scalable and energy efficient routing algorithm. VBF is a geographic routing protocol wherein routing and localization is performed at the same time. This routing protocol uses a routing vector to specify the forwarding path from sender to receiver. All the packet forwarders in the sensor network form a “routing pipe”. The sensor nodes in this pipe are eligible for packet forwarding, and those which are not close to the routing vector (i.e., the axis of the pipe) do not forward. VBF protocol can be used for networks with small or medium node mobility.

On the similar lines of location based routing, authors in [99] have developed adaptive routing protocol called Hop-by-Hop Vector Based Forwarding (HH-VBF). They suggest that better performance can be obtained with the use of routing vector for each individual forwarder rather than a single network-wide source-to-sink routing vector. HH-VBF is the enhanced version of VBF protocol, which can improve the packet delivery in the dense as well as sparse network overcoming the limitations of VBF protocol. Both the protocols, VBF and HH-VBF are robust to node mobility.

Another multi-hop routing protocol which is based on distributed algorithm and cross-layer scheme has been proposed in [100]. Termed as FBR (Focused Beam Routing protocol), this geographical routing protocol considers bit level energy consumption and end-to-end packet delay as the metrics for establishing route at each stage of the path on a dynamic basis.

In [101], two distributed routing algorithms are introduced for delay-insensitive and delay sensitive applications based on geographical routing protocols. The proposed solutions allow each node to select its next hop based on strategy of channel and application awareness, while aiming at energy saving. Routing algorithms for delay sensitive applications have stricter delay bounds for end-to-end packet delay.

Similarly, DFR (Directional Flooding based Routing) [102] is a hop-by-hop, geographical routing based protocol which relies on packet flooding technique. Participation of the nodes in the flooding or forwarding process depends on the link quality. This ensures network reliability.

In [103], diffusion based multi-hop ad hoc routing protocol termed as Underwater Diffusion (UWD) is proposed. UWD is an in-network processing protocol that has no proactive routing message exchange and negligible amount of on-demand floods.

In [104], author has provided the energy analysis of routing protocols in deep as well as shallow water. The protocols based on direct transmission, packet relaying and clustering have been analyzed in this paper. Author suggested the use of clustering based routing protocol to achieve better performance and energy efficiency. In [105], a distributed energy aware routing protocol (DUCS, Distributed Underwater Clustering Scheme) has been proposed. This GPS-free routing protocol uses data aggregation technique to reduce redundant information and also tries to reduce message exchanges.

Achieving distributed localization in UASN is very challenging and expensive because of high delay and variance of acoustic channel [107]. There are certain routing protocols which do not require the localization information for the routing. Examples of these protocols are PULRP [108], E-PULRP [109] and Hydrocast [110], and APCR [111].

Path Unaware Layered Routing Protocol (PULRP) is a routing protocol that has been proposed for a dense three-dimensional UASN [108]. This protocol does not require fixed routing tables, localization or time synchronization. Protocol is divided into two phases. In the first, (layering) phase, concentric spheres are structured around the sink, radius of these spheres depends on the probability of successful transmission of packets and the latency involved. In the second (communication) phase, intermediate sink nodes are chosen for 'on the fly' packet delivery from source to sink. Authors have added an energy optimization scheme in the protocol and have proposed E-PULRP, (Energy optimized Path Unaware Layered Routing Protocol) [109]. Here, the radius of sphere is decided based on the transmission energy of the node along with other criteria suggested in PULRP.

An alternative to geographic routing, Hydrocast is proposed by Lee et.al. in [110]. It is a hydraulic based anycast routing protocol, which uses pressure information for routing data. It uses a novel

opportunistic routing mechanism to maximize the “greediness” of packet forwarding process by selecting a subset of forwarders. It also provides a discovery method to provide efficient recovery route using two-dimensional surface flooding.

APCR (Adaptive Power Control Routing) is another energy-aware routing protocol proposed by Al-Bzoor et. al. in [111] that does not require any location information. In this protocol, nodes are assigned to concentric layers around the sink, based on their power levels. Routes are determined on the fly depending upon the layer number and the residual energy within the node. It is assumed that the node can increase transmission power in order to find a forwarding node.

In UASN, link disconnection might be observed because of i) sparse deployment, and ii) changes in topology. Changes in topology might happen because of ocean currents or unforeseen mobility of nodes. This creates a routing hole or void in the network topology. Routing protocols VAPR [112], and Mobicast [113] are developed to address this issue.

In [112], Void-Aware Pressure Routing (VAPR) is proposed. This protocol uses surface reachability information to select each node’s next-hop in the direction towards the surface, through which local opportunistic directional forwarding can always be used for data packet delivery even in the presence of voids. Hydrocast [110] is also a pressure routing protocol with route recovery mechanism. In [112], it is shown that VAPR protocol outperforms Hydrocast protocol with route recovery.

Mobicast routing protocol [113] addresses the mobility and topology-hole issue. This protocol assumes either a mobile sink or AUV having pre-defined trajectory to continuously collect data from nodes within a series of 3-D zone of reference (termed as 3-D ZOR). Before entering in the subsequent 3-D ZOR, nodes in that zone are notified to enter in the active/wake-up state in order to save energy.

In [114], authors have provided a comprehensive overview of current research in field of UASN focusing on lower layers of network protocol stack such as physical, data link and routing layer.

From the survey of routing protocols, it is evident that the following features such as a) self-configuration in case of failure or loss of connectivity, b) robustness, c) reliability, d) energy efficiency and e) achieving latency bounds based on application’s requirement are very important in the design.

2.1.4 Transport layer

Transport layer protocols are required for reliable data transmission, along with flow and congestion control. In the conventional transport layer protocols, two end nodes, that is, source and destination nodes cooperate with each other to ensure the end-to-end reliability. These end nodes may have multiple hops in-between. Intermediate nodes do not have to assess reliability. If the receiver has not received the packet, it can ask sender to retransmit it. This end-to-end approach is used in TCP (Transmission Control Protocol) implementation, which relies on accurate estimate of RTT (Round Trip Time). In underwater networks, such approaches are inefficient because of long and variable propagation delay.

For underwater networks, various ARQ based schemes are available which work on hop-by-hop basis [115–117]. These schemes are based on use of ACK, NACK for notifying the success/failure of transmission to the source node. ARQ-based solutions suffer from long propagation delay, congestion, and high energy consumption problems due to retransmissions.

FEC (Forward Error Correction) based schemes essentially exploit redundancy to ensure reliability. Additional energy is always used for reliable data transfer irrespective of the channel conditions. Hence, pure FEC based protocols are not energy efficient for UASN. Various data transport protocols such as ADELIN (ADaptive rELIable traNsport) [118], IPool-ADELIN (ADELIN protocol with IPool Node) [119], FRT (Fast and Reliable Transport protocol) [120], AR RTP (Adaptive Redundancy Reliable Transport Protocol) [121] have been developed based on use of FEC on per hop basis. In order to perform error control coding, these protocols use BCH (Bose, Chaudhuri, and Hocquenghem), fountain, erasure, Hamming, or Reed Solomon codes.

MPNC (Multiple Paths and Network Coding) [122] is a reliable data transport protocol that is based on network coding [123]. Authors claim that by employing coding at the nodes (referred to as network coding), bandwidth can in general be saved. Here, multiple paths (namely three disjoint paths) are first established between a source and sink node. Two groups of packets A and B, that use network coding, are transmitted over the two side paths, and another group of packets C ($C = A \oplus B$) are transmitted over the middle path. It is shown that, MPNC not only improves the throughput of network but also has a higher data delivery ratio and lower energy consumption without any reduction in the data transmission reliability.

Hybrid ARQ schemes, which combine coding with per hop retransmission-upon-failure, are promising schemes for error-prone underwater networks. Several protocols based on a hybrid approach have been proposed recently such as SDRT [124], FOCAR [125], UW-HARQ [126].

SDRT (Segmented Data Reliable Transfer) [124] is a hybrid approach that uses both FEC and ARQ. In this protocol, the encoded packets are transmitted block-by-block. Each block is forwarded hop-by-hop from source to destination. A stream of encoded packets in a block is transferred until a positive feedback is received from receiver. A window based control mechanism is used in SDRT, wherein the packets inside the window are transmitted quickly and remaining packets are sent at lower rate. A receiver continues receiving packets until it can reconstruct the original data packet. After reconstructing the original data packet it sends a positive feedback to sender and then forwards the packets on next hop.

Authors Zhou et. al. [125] have proposed a reliable data transfer protocol; termed as FOCAR (FOuntain Code based Adaptive multi-hop Reliable data transfer). FOCAR is hybrid ARQ scheme, combining per hop retransmission-upon-failure with the use of Fountain codes. Fountain codes have been selected due to the strong error correction capability and rateless property, which greatly simplifies the selective repeat procedure for reducing end-to-end delay and improving energy efficiency.

Recently, authors Mo et.al. have proposed a UW-HARQ scheme for end-to-end reliable data transfer in underwater network [126]. This hybrid ARQ scheme, termed as Underwater Hybrid ARQ (UW-HARQ) uses both FEC coding and ARQ techniques. For FEC, it uses simpler random binary linear code to suit the low computational capabilities and energy constraints of UASN. In ARQ, this scheme uses NACK packets to notify sender about missing packets at the receiver's side. Authors have also described the details of implementation and test-runs performed using acoustic modems.

An interesting biologically-inspired congestion control protocol has been proposed in [127]. It is based on the study of certain marine communities which form a self-organized biological system and are capable of bringing the system to stable state. These marine communities adapt to biological events (such as phytoplankton blooms) without any need for a central controller. This congestion control protocol utilizes a distributed algorithm to distinguish between packet losses due to congestion and those due to high link error rates. It detects and eliminates flow

starvation, and also provides event fairness. This protocol reduces packet losses due to congestion and improper channel conditions bringing the system to the stable state after a small convergence time.

2.1.5 Application layer

Application layer makes the hardware and software details of lower layers transparent to the network user. This layer is used by system administrator to interact with the sensor network for configuration, data collection, information dissemination and so on. Application layer should provide a mechanism for querying the whole sensor network or the part of the network. It should also be able to assign tasks to particular subset of the network. Dissemination of essential network management information (such as location update, time synchronization, reconfiguration, resource allocation, network reorganization) must be effectively handled by the application layer protocol for the network. Application layer protocol development for underwater acoustic sensor network is a relatively unexplored area [4, 128].

2.1.6 Management planes for UASN

Sensor networks require various management planes such as i) task management plane, ii) mobility management plane, and iii) power management plane. The implementation of these planes require different protocols such as localization, time synchronization, clustering and so on. A lot of research is directed in an effort to develop protocols suitable for underwater environment. In this section, literature review of these protocols is described briefly.

2.1.6.1 Localization

Many applications of WSN or UASN requires the data to be location-aware to make it more suitable for analysis. Along with the application's requirement, various protocols such as geographical routing, MAC, clustering, and time synchronization also require location information.

With regard to the mechanisms used for location estimation, localization algorithms can be divided into two major categories, range-based and range-free. In the range-based location algorithms, distance or angle estimation with respect to the neighbors are used for calculating node's location. In range-free location algorithms, the neighbor distance/angle information is assumed to be unavailable for positioning due to cost and hardware limitations.

There are several localization protocols developed for terrestrial WSNs, but since an acoustic channel is severely impaired and GPS signal cannot propagate far through water, the existing positioning schemes for terrestrial WSNs cannot be used directly for UWSNs. Various localization schemes have been proposed by researchers for UASN [129–131].

In [129], a new positioning scheme, Dive'N'Rise (DNR) is proposed. Here localization is performed with the use of mobile DNR beacons equipped with GPS. These DNR beacons move to water surface to learn their coordinates via GPS and then dive down to disseminate this information for localization of ordinary nodes of network. Positioning at nodes is performed either by bounded box or triangulation algorithms.

In localization scheme proposed in [130], three types of nodes are used: surface buoys, anchor nodes and ordinary sensor nodes. Localization is performed in hierarchical manner in two steps. First step is to localize anchor nodes by using direct communication with surface buoys equipped with GPS services. In step two, localization of ordinary nodes is performed by using a novel distributed method based on three-dimensional Euclidean distance estimation and a recursive location estimation.

LDB (Localization with Directional Beacons) [131], uses AUVs as a mobile beacon sender. This AUV is programmed with pre-defined trajectory to traverse the sparse three-dimensional UASN. It then sends the beacons with constant intervals to sensor nodes, so that the nodes can localize themselves.

A detailed survey of various localization schemes proposed for UASN is available in [107, 132].

2.1.6.2 Time synchronization

The process of achieving and maintaining a common time-base throughout the network is called as time synchronization. Sensor networks are distributed systems where operations of individual nodes are controlled by the timing information available from the local clock. Applications require collaborative information from multiple sensor nodes. Timing information associated with data of each sensor node needs to be correct. A time synchronization protocol can be used to achieve global time base for all nodes in the network. Many protocols in the protocol stack of sensor networks (for example, TDMA based MAC protocols, localization) cannot work efficiently without achieving time synchronization. In fact it can be stated that if “Time synchronization protocol works efficiently, the rest of network protocol stack will automatically converge.”

Many time synchronization protocols have been proposed by researchers for terrestrial WSNs [40, 133–135]. These protocols assume that propagation latency is negligible; a valid assumption since the RF propagation speed in air is 3×10^8 m/s. The speed of sound in water is approximately 1500 m/s, which leads to large propagation delays. Protocols developed for RF-based WSNs such as RBS (Reference-Broadcast Synchronization) [133] and FTSP (Flooding Time Synchronization Protocol) [134] cannot be applied to the UASN scenario, since these protocols assume nearly instantaneous and simultaneous reception, a phenomenon impossible for the underwater acoustic communication. Other protocols such as TPSN (Timing-sync Protocol for Sensor Networks) [40], LTS (Lightweight Time Synchronization) [135] ignore the clock drift during synchronization. Synchronization phase itself will take significant time in UASN due to high propagation delays per message transmission; therefore such protocols cannot provide desired accuracy.

Various protocols developed by researchers for achieving time synchronization in UASN are TSHL [136], Tri-message [137], MSUAN [138], UA-TSP [139] and TSLA [140].

In [136], authors have introduced a protocol titled ‘Time Synchronization for High Latency’ (TSHL) that compensates for high-latency communication. The key idea in TSHL is that it splits time synchronization into two phases. Phase 1 is used to compensate clock skew by using linear regression over multiple beacon values, whereas phase 2 corrects the clock offset. In phase 1, around 25 beacon messages are used for linear regression process to achieve better accuracy. In the energy constrained network such as UASN, this would result in faster drain of the node’s

energy. Also, since the propagation delays are high and transmission rates are typically lower for the acoustic modems, these message exchanges take significant amount of time. The difficulty of implementing TSHL on multi-hop hierarchical network would be quite pronounced.

In [137], ‘Tri-message time synchronization protocol’ is suggested for high latency networks, keeping the resource constrain as main focus of design. Authors Tian et. al., suggest the use of only three messages for achieving time synchronization, thus reducing the time and energy spent in the process. Protocol is designed assuming several factors such as (i) constant propagation delay over the duration of message exchange, (ii) time-stamping at lowest layer of protocol stack and (iii) short-term skew-stable clocks. Authors have proposed this protocol for time synchronization over single hop link, and have shown that the overall computational cost of this protocol is very less.

In paper [138], authors Sun et.al., have discussed the extension of Tri-message protocol for the multi-hop scenario, considering overhearing as a tool to achieve better accuracy. This protocol is termed as Multi-hop time Synchronization scheme for Underwater Acoustic Network (MSUAN). In sensor network which needs implementation of sleep-wake mechanism for energy conservation, overhearing of the information may not be feasible. Moreover overhearing and idle listening are major sources of energy wastage and hence not recommended in resource constrained sensor network.

In [139], time synchronization protocol termed as UA-TSP has been proposed. This protocol assumes the availability of precise local timer for synchronization. It implements a loop convergence scheme in order to reduce errors in multi-hop topology. A sink node is the controller of time synchronization and it is used to broadcast the local time in terms of synchronization commands. In the implementation of this protocol, effect of sleep-wake schedule and packet drops is not considered.

In [140], authors have presented a Time Synchronization and Localization Algorithm (TSLA) for underwater acoustic sensor networks. It uses number of anchor nodes for broadcasting one-way packets to synchronize and localize sensor nodes. It is stated that since the nodes only need to receive packets, energy consumption of the sensor nodes is reduced. This protocol requires spatial distribution of anchor nodes for achieving the synchronization. The process of synchronization

of anchor nodes as well as the energy consumption of these nodes has not been discussed in the paper.

Node mobility can have significant impact on the the time synchronization. Certain time synchronization protocols have been developed for mobile UASN scenario such as D-Sync (Doppler-based time Synchronization) [141], TSMU (Time Synchronization scheme for Mobile Underwater sensor Network) [142], Mobi-Sync [143], E2DTS (Energy Efficiency Distributed Time Synchronization) [144].

2.1.6.3 Clustering

Use of clusters and multi-hop communication for transmitting data to the Base -Station leverages the advantages of small transmit distances for most nodes, requiring only a few nodes to transmit data over long distances to the Base-Station. Since, the energy dissipation during a data transmission is proportional to distance between the transmitter and the receiver, short distance transmissions are comparatively energy efficient. Further, in order to equalize the energy distribution (or dissipation) across all nodes, it is essential to rotate the Cluster-Head based on the node's residual energy. Several clustering protocols have been proposed by the authors such as DUCS [105], MCCP [145], LCAD [146], S-LEACH and C-LEACH [147].

In [105], author has proposed a distributed clustering scheme for shallow water scenario. In this protocol, nodes organize themselves into a local cluster, and one node is selected as a single Cluster-Head. Nodes in a cluster send their data to the Cluster-Head via single hop. The Cluster-Head performs data aggregation, and forwards the data to the sink using a multi-hop route (via Cluster-Heads). Cluster-Heads are also responsible for coordination among nodes within their clusters (intra-cluster coordination) and for communication with other Cluster-Heads (inter-cluster communication). Randomized rotation of Cluster-Head is implemented in DUCS for even distribution of energy among all nodes.

In [145], authors have defined a metric termed as cluster cost for each potential cluster for the formation of clusters in UASN. This cluster cost takes the following parameters into consideration i) total energy consumption of cluster nodes, ii) residual energy of Cluster-Head and cluster nodes, and iii) relative distance of Cluster-Head and sink. The proposed protocol, termed as Minimum

Cost Clustering protocol (MCCP), selects a set of non-overlapping clusters from all potential clusters based on the cost metric assigned to each potential cluster and attempts to minimize the overall cost of the selected clusters.

In [146], authors K. R. Anupama et.al. have proposed a Location-based Clustering algorithm for Data collection (LCAD) protocol for three-dimensional hierarchical UASN. In this network architecture, sensor nodes are deployed along the entire ocean column at fixed relative depths from each other. Sensors at each tier are organized in clusters with multiple Cluster-Heads. A Cluster-Head selection algorithm based on memory and energy resources is employed at each cluster to select the Cluster-Head based on the position of the sensors in the cluster.

In [147], authors have proposed two novel cluster formulation protocols for the Low Energy Adaptive Clustering Hierarchy Protocol (LEACH) focusing on energy conservation. It states that the standard LEACH protocol [148] designed for terrestrial network perform poorly in underwater networks owing to the unique characteristics of underwater acoustic channel. Compared with the standard LEACH protocol, the proposed S- LEACH (Slotted-LEACH) protocol can avoid the collision of the advertisement packets completely by dividing the broadcast time in slots. Another improved protocol termed as C-LEACH (Controlled-LEACH) has also been proposed in this paper. In C-LEACH, control node is added at the center of the network topology, which is used to avoid the collisions between advertisement messages and also for broadcasting the advertisement messages on behalf of Cluster-Heads.

2.2 Cross-layer protocol stack for UASN

Network efficiency can be improved by using a cross-layered approach which explores the interaction between different layers of the protocol stack in order to optimize resource usage. In traditional layering approach, a network works in a suboptimal state since it lacks information sharing between network layers. Cross-layer protocol stack design can involve joint design of different functionalities, from modem design to MAC and routing, from channel coding and modulation to source compression and transport layer [4, 149].

Several researchers have proposed protocols based on the interaction between various layers or functionalities of the network [100, 101, 150–153].

In [150], authors have proposed joint allocation of frequency and power in a cross-layer protocol for UASN. Power control is used as a practical means of optimizing the overall performance across the physical, medium access control and routing layers. It is demonstrated in the paper that if the center frequency, bandwidth and transmission power are chosen according to the network condition, then average energy consumption per bit can be reduced.

In [151], Kuo et. al. have developed a tier-based distributed routing protocol for applications that requires reliability and have various delay constraints. This protocol provides an end-to-end delay guarantee, where each node distributively selects an optimal hop for forwarding data and the optimal power for transmission. In a cross-layer fashion, this proposed algorithm uses cooperation of the transmitter and receiver to achieve the desired level of reliability and data rate according to requirement of the application and channel condition.

CARP (Channel Aware Routing Protocol) [152] is a cross-layer protocol for UASN. Here, the nodes that have a history of successful transmission are selected as relays. This link quality information along with topology information (hop count) is used for finding suitable routes from source to sink.

FBR [100] uses a cross-layer approach, in which the routing protocol, medium access control and physical layer functionalities are tightly coupled by power control.

In [101], cross-layer interactions between medium access control layers and forward error correction schemes are used for determining the optimal packet size.

In [153], authors Pompili and Akyildiz have presented a brief survey of possible approaches to cross-layer design in WSN. They have explored the interactions of key underwater communication functionalities such as modulation, forward error correction, medium access control, and routing in UASN.

In Section 2.3, survey of development of research platforms for sensor networks has been provided. Development in software and hardware tools in the field of UASN has been detailed. Overview of various testbed implementations and real-time deployment has also been provided.

2.3 Research platforms for sensor networks

WSNs have matured into viable systems for sensing requisite parameters for diverse applications. The diversity of applications lead to differing requirements such as reliability, quality of sensed data, longevity and so on. WSNs have tight system integration because of resource constraints and harsh environmental conditions. Design and development of WSNs is a complex and error-prone process which requires careful planning. Functional correctness at the time of actual deployment is also utmost important. For example, a trivial issue such as placement of the node also needs to be critically examined to achieve robustness from seasonal variations of environment, proper sensing and communication coverage, and to scavenge energy from the surrounding. Even after all ingenious engineering efforts, WSN deployments fail to perform or to work at all [154, 155].

In order to deploy correct and performing system, network architecture and protocol stack design of WSN should be tested using different tools for verification and validation. WSN is the distributed system and is integration of various fields such as wireless communication, embedded system, and mechanical design. It has issues and challenges arising from all these fields. Along with these challenges, a tight integration into environment increases susceptibility and dependencies. Visibility of system is limited because of low data rate and scarce energy. In WSN, various resource consumptions are also heavily optimized. Testing of WSNs must encompass these aspects of the system. Testing tools needs to consider such complexities of distributed system.

Different system properties and functionalities of WSN can be tested using different test platforms. Test platforms consisting of simulation tools and hardware testbeds can be characterized by parameters such as a) scale, b) visibility, and c) abstraction [156].

1. Scale - Simulation tools allow the testing of large scale WSN to identify system problems at global scale. For example, simulation platforms can be used to validate the scalability, convergence of protocols, overall resource consumption and so on. On the other hand, testbeds can be used to test local problems using limited number of sensor nodes. For example, testing of i) driver circuitry for hardware interface, ii) communication link performance between two nodes, and iii) working of primitive multi-hop topology link for routing or collision is possible using WSN testbeds.

2. **Visibility** - Simulation tools can allow detail inspection of system without perturbing it. All required diagnostic information can be easily collected and analyzed. In the hardware testbed, the focus is to observe or detect a specific error or failure. Amount of data logging depends on a particular test-case being performed.
3. **Abstraction** - Test platform should abstract the features from actual final deployment of location or scenario of system. Various models for testing including device, network, communication, and environmental abstractions need to be built for proper validation. While simulation tools can help tune to the more comprehensive and complex communication model, indoor testbed can be better suited for analysis such as energy consumption profile of power supply.

Various test platforms previously used in wireless network testing such as NS-2 (Network Simulator 2) [157], GloMoSim (Global Mobile Information System Simulator) [158] have been used for simulation of WSNs as well. Simulation libraries targeted for WSNs has been provided in simulation softwares such as Castalia [159], TOSSIM (simulator for TinyOS wireless sensor networks) [160] and so on. To verify the practical deployment challenges, various testbeds have been implemented for testing WSNs such as Motelab [161] and DSN (Deployment Support network) [162]. These testbeds have been installed in laboratories or on university campuses. These testbeds help to understand characteristics of various network topologies, wireless communication links and power supplies. Testing the network using testbed is closer to reality as compared to simulation or emulation techniques, since the compiled code can be executed on the testbed in its binary form.

To have even better grasp of system's functionality and systematic validation of design, researchers suggest field trial of WSNs. Such field testings are usually prohibitive because of infrastructure and cost involved. Another novel approach is to develop an integrated testing methodology which offers simulation, emulation and testbed execution in single framework such as EmStar [163] or COOJA [164].

For testing the performance of UASN protocol stack, various academic universities and research centers across the world have developed different simulation environments. Also, on the basis of research publications, marine projects, defense projects and collaborative work, various companies

have developed modems for the use in underwater acoustic sensor networks. A brief information on the software and hardware developed in the field of UASN has been provided in next subsections.

2.3.1 Development in software for UASN

Software platforms simulate the underwater acoustic channel behavior by using various models (for example, Urick's model, ray tracing model and so on). Simulations can make use of real environmental data such as bathymetry profile, sound speed profile, bottom sediment and so on, from the ocean databases available. Simulation softwares are platforms to conduct various experiments with simplicity, repeatability and flexibility in terms of network size, mode, traffic pattern and topology. It helps in comparing and evaluating different network designs, algorithms and protocols. Prominent simulation environment have been listed here. Brief information of these simulation platforms has also been provided.

2.3.1.1 Aqua-Sim (UWSN Laboratory, University of Connecticut)

Aqua-Sim [165, 166], an NS-2 based simulation platform has been developed by the Under Water Sensor Network Laboratory (UWSN Laboratory) in the Computer Science and Engineering Department at the University of Connecticut [167]. Aqua-Sim is discrete-event driven network simulator developed on the basis of NS-2 for simulating underwater acoustic channels with high-fidelity. It can effectively simulate acoustic signal attenuation and underwater packet collisions. It has support for three-dimensional networks as well as for underwater mobile networks. It can be used to implement a complete protocol stack from physical layer to application layer. Many MAC protocols such as ALOHA, Broadcast MAC, GOAL, R-MAC, UWAN-MAC, COPE-MAC have been implemented on Aqua-Sim. (or on the recently updated version of Aqua-Sim 2). Similarly, various routing protocols such as VBF, HH-VBF, Dynamic routing have also been developed on this platform.

2.3.1.2 DESERT (Under project NAUTILUS)

DESERT (DEsign, Simulate, Emulate and Realize Test-beds for Underwater network protocols) [168, 169] is the platform developed under the NAUTILUS project funded by Italian Institute of Technology. The project is mainly carried on by the Consorzio Ferrara Ricerche (CFR). It is a complete set of public C++ libraries that extend the NS-MIRACLE [170] simulator to support the design and implementation of underwater network protocols. It aims to extend the NS-MIRACLE simulation software library, developed at the University of Padova, in order to provide several protocol stacks for underwater networks, as well as the support routines. DESERT has implementation of physical layer module, four different MAC protocols (CSMA, DACAP, Tone-Lohi, UW-Polling), four routing layer protocols (Static routing, Flooding, SUN, ICRP), an error control protocol (USR), a cross-layer protocol involving MAC and hierarchical routing (U-Fetch), two transport layer modules, two application layer modules and several mobility modules. More importantly, the simulator has an interface to communicate with acoustic modems. This interface helps in conducting experiments to test network protocols in the real world.

2.3.1.3 SUNSET (UWSN group, SENSES Lab, Sapienza University)

SUNSET (Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing) [171, 172] is a new solution developed by the UWSN Group, to seamlessly simulate, emulate and test the novel communication protocols in real-life. SUNSET allows the user to develop the code with the help of NS-2 and NS-2 MIRACLE. It is a powerful toolkit for implementation and testing of novel protocol solutions for underwater sensor networks. SUNSET enables testing and enhancement of underwater solutions using a controlled simulation environment. Simulation code can be transparently ported to various hardware and underwater platforms for protocol emulation and actual in-field testing. SUNSET was the first open source framework for seamless simulation, emulation and actual at-sea testing of novel underwater systems. The new SUNSET version 2.0 has been recently released to the community in October 2013.

When running simulations on SUNSET, it can use different underwater acoustic channel models such as empirical formulas, Bellhop ray tracing and so on. On the other hand, in emulation mode,

SUNSET can be interfaced with actual hardware such as i) acoustic modems for underwater communication, ii) sensing platforms for data collection, iii) the navigation system of AUVs (Autonomous Underwater Vehicles) and ASVs (Autonomous Surface Vehicles) to control the vehicle, and so on. The designed architecture is flexible and open enough to allow the integration of any external device provided the API's are available to control the operation.

2.3.1.4 WOSS (Patavina Technologies, NATO Undersea Research center)

WOSS (World Ocean Simulation System) [173, 174] is a multi-threaded C++ framework that permits the integration of any existing underwater channel simulator that expects environmental data as input and provides channel realization as output. It incorporates a Bellhop ray tracing tool for a more realistic reproduction of underwater propagation and NS-MIRACLE for flexible programming of protocol stack. Here, user only has to specify the location in the world and the time where the simulation should take place. WOSS is integrated with free world databases for environmental parameters which are used in the simulation of the UASN protocols.

2.3.1.5 UNET (NUS, Singapore)

The UNET (Underwater NETWORK) project [175, 176] started at the Acoustic Research Laboratory (ARL) [177] of the National University of Singapore in 2004. UNET project is composed of UnetStack (an agent-based network stack) and an underwater network simulator that can be used for development and testing of underwater network technology. The simulator is easy to learn and use. It permits agent implementations to be shared between deployment environment and simulation environment. Essentially, once a protocol is developed and tested in simulation, it is ready for deployment and at-sea testing with the help of UnetStack-compatible modems.

2.3.2 Development of hardware tools for UASN

A number of hardware tools for underwater acoustic communication have been developed over the years, for commercial, military and research purposes [8]. These platforms are essential for providing support for testing and field use. List of some popular modems has been provided in

Table 2.1. A detailed survey of design of underwater sensor node has been provided in [178]. This paper provides a comprehensive study of all elements that compose a sensor node, including microcontrollers, memories, sensors, and batteries.

Company or Institute	Name of product	Band (kHz)	Bit rate (bps)	Range (km)
Aquatec	AquaModem 1000 [179]	7.5 -12	2000	5
Desert Star Systems	SAM-1 [180]	33-42	150	1
DSPComm	AquaComm [53]	16-30	480	3
EvoLogics	S2CM 48/78 [181]	48-78	31200	1
LinkQuest	UWM2200 [52]	53-89	35700	1
MIT	AquaNode [182]	30	300	0.4
NUS	UNET-2 [183]	27	10000	3
Teledyne-Benthos	960 series ATM 965 [184]	9-14 16-21 22-27	15360	2 to 6
TriTech	Micron data modem [54]	20-24	40	0.5
UCI	Software Acoustic modem [185]	0.4-1.4	48	0.01
UCLA	UANT (System) [186]	100-175	5k - 15k	NA
UCONN	OFDM acoustic modem [187]	9.0 -15.0	3.2 (SISO) 6.4 (MIMO)	NA
UCSB	Aquamodem [188]	22-27	133	0.3
USC	Low power acoustic modem [189]	18	600	0.5
WHOI	Micro-modem [190]	23-27	5400	2

Table 2.1. List of various modems for UASN.

2.3.3 Testbeds and real-time deployments

Huge amount of interest in underwater networks has resulted in a great deal of research in the development of laboratory based testing environments using simulators, but the field experiments still remain difficult and costly. Table 2.2 gives an overview of various testbed implementations and deployments in laboratories or real-time deployments in fields.

Organization	Project or Testbed	Comment
US Navy (NPS)	Project Seaweb [191]	1) Used for undersea acoustic sensing, signaling, communications and networks. 2) Over 40 deployments.
USC -ISI	Testbed [192]	1) Deployment using WHOI modems. 2) Harbour site -Marina del Rey.
UCLA	UANT [186]	1) Software defined underwater acoustic networking platform. 2) Complete end-to-end networking platform.
UCONN, WHOI	Aqua-Lab [193]	1) Hosted at UWSN lab, UCONN. 2) Provides field test experience in lab controlled environment.
WHOI, MIT, USC/ISI, UCONN, UMA	ORTUN [194]	1) Open research testbed for underwater ad hoc and sensor networks. 2) First open testbed infrastructure.
Rutgers University	Testbed [195]	1) For underwater vehicle team formation and steering algorithm. 2) User can configure communication parameters.
WHOI	UAN-Testbed [196]	1) Infrastructure for evaluating and developing network protocols. 2) Nodes can be remotely controlled.
La-Sapienza, WHOI	SENSES [197]	1) Framework to seamlessly simulate, emulate and test at-sea. 2) Can be ported on various available commercial modems.
USC-UCLA	DATUNR [198]	1) Development of an always-available testbed for underwater networking research 2) Remote-accessibility to the testbed
UCONN,UCLA, UW, TAMU	Ocean-TUNE [199]	1) A community ocean testbed for underwater wireless networks. 2) Deployment at 4 different sites for diverse coverage of the US coast.

Table 2.2. Various testbed implementations and real-time deployments.

2.4 Summary

UASN design and development is very different and challenging as compared to terrestrial RF-based WSN. It requires a fairly different approach for designing protocols and network architecture. A distributed cross-layer protocol stack can be designed for UASN by integrating various communications and networking functionalities with an aim to jointly optimize the network resources such as memory, power and bandwidth. Various network management protocols

such as localization, time synchronization and clustering can be accommodated in a single common framework. Cross-layer modules can help in improving end-to-end network performance in terms of energy consumption and throughput. Researchers so far have developed or suggested protocols for various layers and planes in UASN without doing any inter-layer dependency analysis. For example, in case of time-based MAC protocols, it is assumed that the time synchronization has already been done. In case of geographical routing protocols, it is assumed that location information is already available.

Though a large volume of work has been done in area of cross-layer protocol stack development for wireless networks and terrestrial WSNs, this is a nascent area of research in UASN. This research work is one of the attempts to build a cross-layer protocol stack for three-dimensional UASN. Energy efficiency has been the main focus of the development for the longevity of the deployed network.

Simulation platforms can be a very effective way for initial testing of the network performance. Appropriate modeling of channel, modem and battery can be used in order to verify the performance of network in terms of reliability, delivery ratio, network latency, energy efficiency and resource utilization. More importantly, emphasis needs to be given on hardware based testing of protocols by using either laboratory based testbed or field testing. Protocols developed on software platforms can be ported on the modems or can be independently developed for testing.

Though there are various tools developed for UASN, there is no universal development environment or operating system for underwater research. In terms of simulation platforms, in many instances, the accuracy in modeling the physical layer is either poor; limiting the usability of such tools or very complex; making it unsuitable for overall analysis. Among hardware testbeds, some of the platforms are now open to research communities. Certain platforms are available for laboratory use and few others are deployed in specific (restricted) geographical areas. Actual at-sea testing still remains fairly difficult and costly operation. It is of paramount importance now to develop universally accepted platform (analytical, computational, hardware testbed and field experiments) and link the various research ideas, approaches and initiatives together. It is also important to specify the options in selecting the complete networking solution along with specific tool for simulation or emulation when the application's requirement are provided.

Chapter 3

Design of Network Architecture and Physical Layer for Three-Dimensional UASN

3.1 Introduction

In this work, a problem of designing a three-dimensional Underwater Acoustic Sensor Network (UASN) has been considered. A static and structured three-dimensional network architecture has been proposed to implement this UASN. In the proposed architecture, the network is arranged in the form of clusters of sensor nodes at various depths in the ocean. Some of the possible applications of this network are pollution monitoring or ecosystem observation of ocean column. In these applications, it is important to continuously monitor certain parameters such as dissolved oxygen, chlorophyll, salinity, pressure, temperature and so on. Rather than having tethered network or intermittent use of underwater vehicles, one-time deployment of an UASN can be effectively utilized for monitoring the ocean environment over a long time without disturbing the natural ambiance.

The proposed three-dimensional network architecture has been described in this chapter. Development of physical layer is closely linked with the network architecture. In this chapter, design and modeling of physical layer has been covered for the proposed network architecture.

The chapter has been organized as follows: In Section 3.2, the three-dimensional UASN architecture has been described in its generic form. Motivation of the network design as well as assumptions made in design of this architecture have been stated. In Section 3.3, details of the three-dimensional architecture have been provided with the help of specific scenario for implementation. For designing physical layer, deep water acoustic channel model has been studied in detail from the available literature. The various factors of this model affecting the three-dimensional network architecture have been briefly covered in Section 3.4. Based on the properties of deep water acoustic propagation, physical layer design parameters have been suggested for implementation in Section 3.5. Finally, a brief summary of the chapter has been provided in Section 3.6.

3.2 Generic three-dimensional UASN architecture

In this work, a generic three-dimensional UASN architecture has been designed for monitoring of ocean column over a long time. This network consists of multiple levels of clusters, deployed at varying depths from sea-surface. Each cluster further consists of number of nodes in horizontal plane. Maximum distance of nodes is R_u meters from center of column. Multiple such clusters have been arranged vertically below each other in the column. Vertical distance between two successive clusters has been denoted by d_u meters. Total depth of cylindrical column has been denoted by D_u meters and radius of column has been denoted by R_u meters. At sea-surface, an energy efficient node has been deployed with the help of underwater buoy. This node acts as an underwater Base-Station for the network and has sufficient power available with it.

A generic three-dimensional architecture has been shown in Figure 3.1. Various levels have been indicated as Level 1, Level 2, Level 3 and so on from top to bottom of the network. The nodes have been address configured as LXX, wherein L denotes the level number and XX denotes the two-digit number of a node. For example, nodes at level 2 have addresses as 200, 201, 202 and so on. Similarly, nodes at level 3 have address numbers as 300, 301, 301 and so on. Node at

level 1 has been configured with address 100 and it acts as a Base-Station (BS) node. BS has been deployed using surface station or a buoy as shown in Figure 1.2. Multiple BS nodes can be made available on surface station for redundancy in case of failures. Total number of levels in the network has been denoted by L_T .

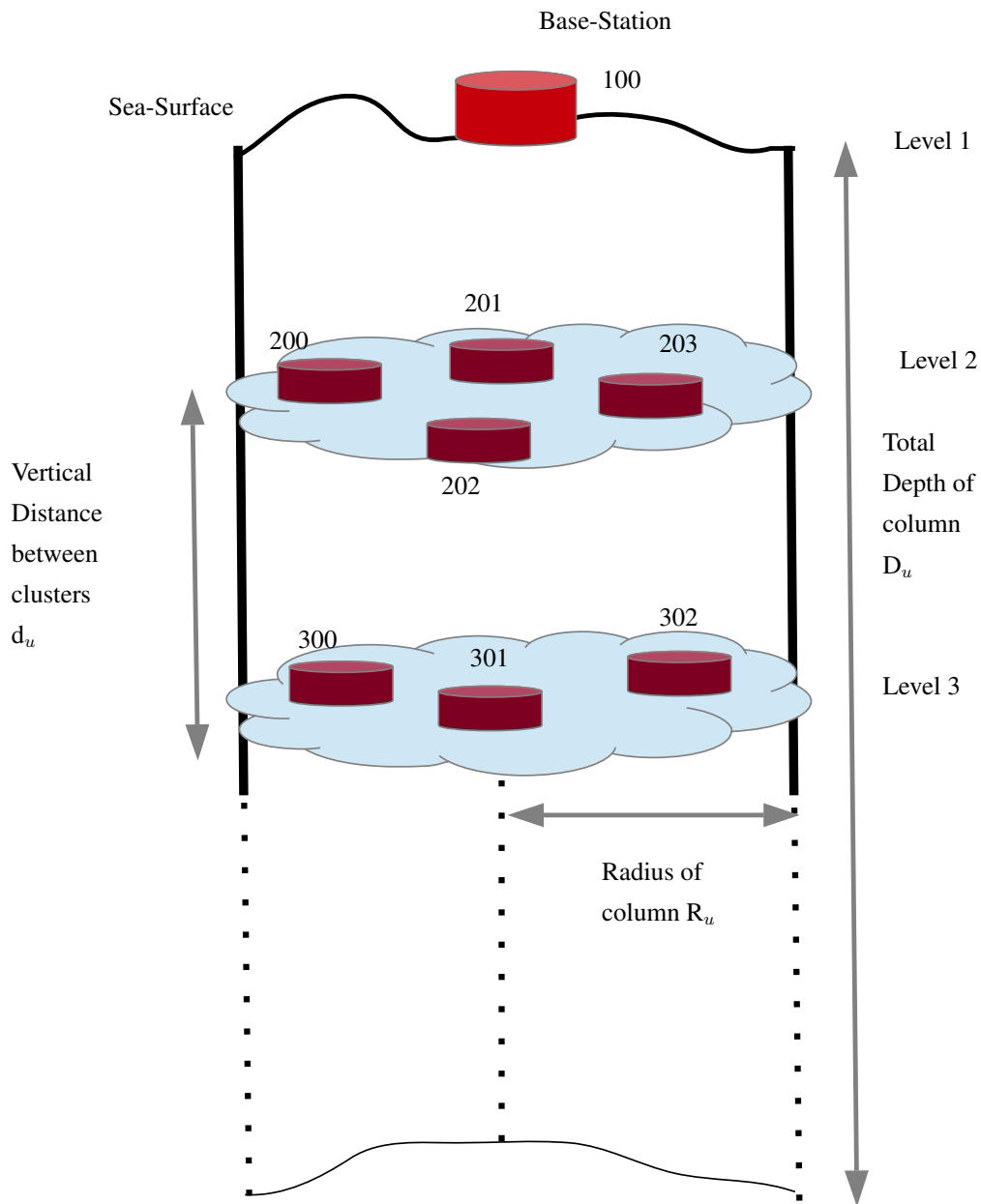


Figure 3.1. A generic three-dimensional architecture of UASN.

In the design of this three-dimensional architecture, following assumptions have been made:

1. It has been assumed that $D_u \gg R_u$. This network has been deployed to observe the ocean parameters along the depth of ocean. It is known that, the variations of ocean parameters

along the horizontal plane is insignificant as compared to variations along the depth. Hence, a column of smaller radius has been considered as compared to depth.

2. At the time of initial deployment, all nodes have been assumed to be placed at their respective levels and positions. Hence, nodes have been assumed to be aware of their locations. It has been assumed that suitable mechanical/electrical arrangement is available to keep the sensor nodes floating at suitable depths.
3. It has been assumed that, $D_u > d_u > R_u$. It suggests that the vertical distance between clusters is larger than the radius of cluster. With this setting, an effective power level management strategy can be used for parallel intra-cluster communication.
4. All nodes have been assumed to have multiple power level for communication. A node is able to communicate till distance R_u with lower power level setting and can communicate till distance d_u with higher power level setting. Power level of a node can be dynamically changed during run-time.
5. At each level, multiple nodes are available in a cluster. Each sensor node has been assumed to be equipped with a particular type of sensor along with processing unit, communication unit and battery unit. All sensor nodes have been assumed to have bidirectional acoustic link available for underwater communication. The choice of sensors is application specific. For example, for pollution monitoring, one sensor node can monitor dissolved oxygen; another node can monitor chlorophyll and so on. Multiple nodes carrying various sensors can be deployed at each level.

Density of nodes or number of particular type of sensors depend upon the required spatial resolution of parameter to be monitored. Coverage of sensing field is one of the performance metric of sensor network, which reflects how well the sensor field is monitored [200]. Sensors may fail because of severe weather conditions or because of possible hardware or software malfunction. Higher node density is helpful in achieving fault-tolerance of coverage as well as connectivity in sensor network [201]. Also, if the region is covered by more than one sensor, it guarantees accuracy, integrity and validity of data. In the proposed architecture, there can be different number of nodes at different levels, but it would be suggested to maintain certain lower and upper bound on the number of nodes.

3.3 Details of three-dimensional network architecture for UASN - a specific case

The three-dimensional network architecture has been designed to cover the ocean column of cylindrical shape. To study a specific case for implementation and analysis, the total depth D_u of the column has been taken as 2000 m and radius R_u has been taken as 10 m. This column has been further divided into five levels of 500 m each ($d_u = 500$ m).

As illustrated in Figure 3.2, clusters have been deployed at every 500 m depth. Each cluster has one Cluster-Head node and ten cluster member nodes. There is a provision in each cluster to accommodate extra five nodes for future expansion.

Sensor nodes in this architecture can belong to any of the three categories, Base-Station node (termed as BS), Cluster-Head node (termed as CH) and Cluster member Nodes (or simply Cluster Nodes termed as CN). Functionalities of these sensor nodes have been defined as follows:

1. **BS:** This is a special node positioned at the sea-surface. BS has acoustic communication link to communicate with CH underwater. It also has an RF link to communicate with surface station at the shore. BS receives control information or commands from surface station over this RF link. By using acoustic links, BS transfers control information related to network management (such as localization, time synchronization, node addition or deletion in a cluster, and rotation of CH) to a CH below it. It also collects data information from CH using an acoustic link. It utilizes the RF link to transfer collected data information to the surface station at the shore.
2. **CH:** Each cluster has a Cluster-Head node denoted as CH. CH receives the control information from CH above it. It then forwards this control information to its cluster member nodes as well as CH situated geographically below it. On the opposite link CH collects data from its cluster member nodes as well as from CH below it; in regular intervals. It then transmits this information to CH above it after appending its own data.
3. **CN:** These are cluster member nodes (or cluster nodes). These nodes receive control information from their respective CH. These nodes generate data packets at regular intervals and transmit it to their respective CH.

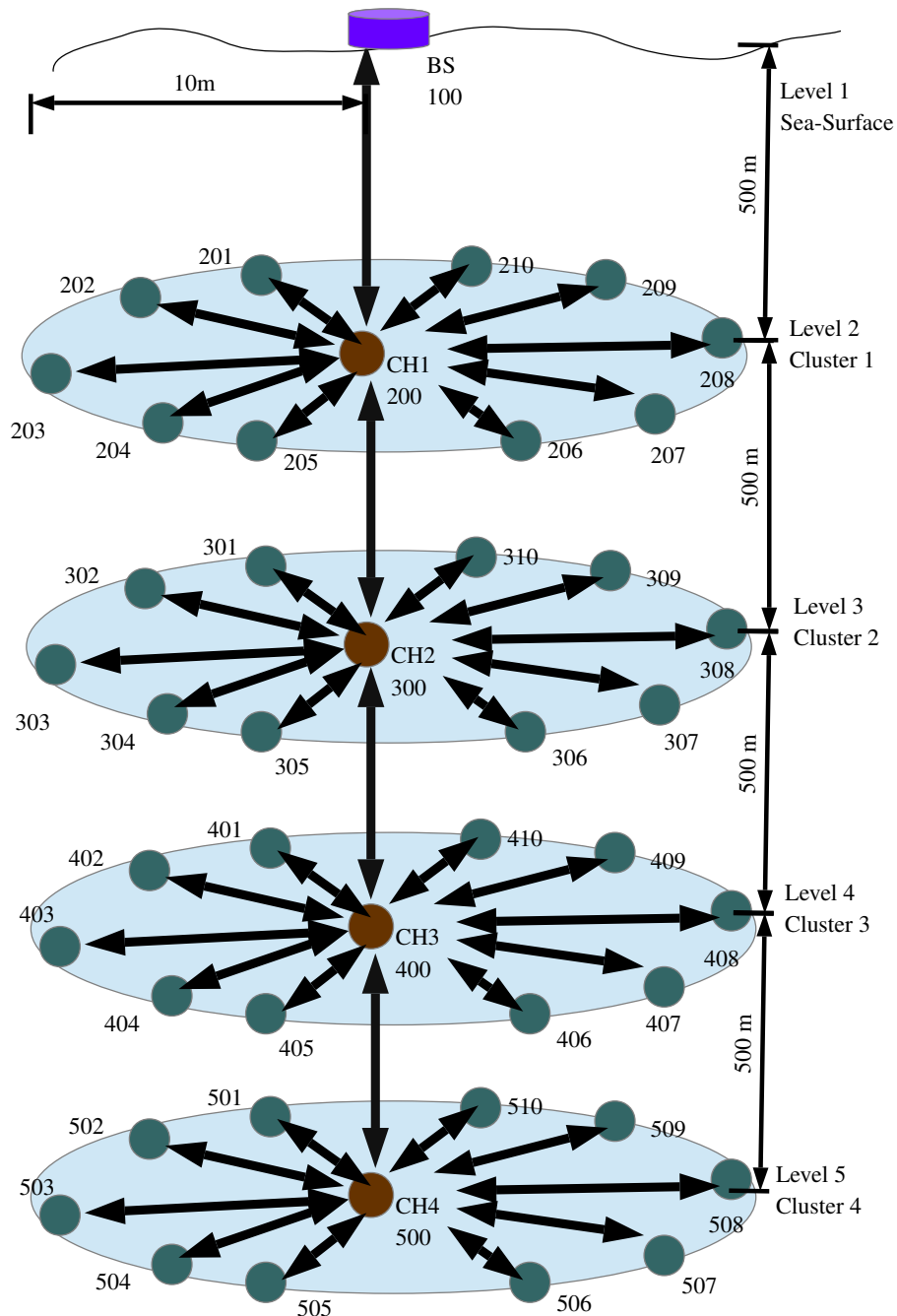


Figure 3.2. Proposed three-dimensional architecture of UASN.

At the time of deployment, nodes have been programmed according to their category and role in the network. Node address has been determined based on the level of deployment as discussed earlier. For example, from Figure 3.2, it can be observed that, BS is the single node in level 1 having address 100. CH nodes of clusters 1, cluster 2, cluster 3 and cluster 4 termed as CH1, CH2, CH3, and CH4 have addresses as 200, 300, 400 and 500 respectively at time of deployment. Addresses of CN nodes of cluster 1 are 201 to 210. Similarly, addresses of CN nodes in cluster

2 are from 301 to 310; in cluster 3 from 401 to 410, and in cluster 4 from 501 to 510. Along with these nodes, 5 more addresses have been made available for accommodating 5 nodes in each cluster as possible future expansion of network. For example, in cluster 1, nodes with addresses from 211 to 215 can be added later. Since the role of node can dynamically change during the working period of network, a node has been referred by its role as well as address. For example, CH1(200) is the CH of cluster 1 with address 200, CN(203) is the CN node in cluster 1 with address 203 and so on.

Multiple columns of similar structure as shown in Figure 3.2 can be deployed in the wider area to be observed. These columns can work in parallel. Each column has a BS, which can collect the data from the respective column. Each BS can transmit data to the on-shore station by separate links or can relay data hop-by-hop in order to transmit to end point.

One node at each level has been initially appointed as CH. CH receives control information related to network (such as time synchronization) from a CH of level above it. After performing necessary action based on the received information, CH forwards the information to another CH of level below it. CH also forwards this information to all corresponding CN nodes. This communication link from top to bottom (that is from BS to CN via CH) has been denoted as forward link. Forward link has been used for transferring control information. On the other hand, a bottom-up communication link from CN to BS via CH has been denoted as reverse link. This link has been utilized for data collection. A communication link consisting of only CH nodes has been denoted as backbone link of network.

CH communicates with other CH by utilizing a higher power level of transmission. CH also communicates with all of its CN nodes. While communicating at horizontal level of smaller distance, CH utilizes lower power level of transmission. Because of these communications, CH may drain energy faster than CN nodes. Cluster-Head selection policy has to be implemented based on the energy consumption of a node.

In this work, a cross-layer protocol stack has been developed for this three-dimensional network architecture, which mainly integrates the concepts of physical layer, data link layer, network layer, transport layer as well as application layer. Various protocols of network management planes such as time synchronization, power level management, and Cluster-Head selection has also been incorporated in the cross-layer protocol stack. Since this network is deployed in deep

water, understanding deep water channel model of acoustic communication is very crucial. The parameters of channel model influences the design of protocol stack. The deep water acoustic channel model has been studied from the available literature. The details of this model has been briefly covered in Section 3.4.

3.4 Deep water channel of underwater acoustic communication

The factors of underwater acoustic communication that influences the underwater networking has been described in detail in the literature [19, 28, 31, 33, 202–207]. Brief investigation on the phenomenon of underwater acoustic communication has been provided in this section with a focus on deep water channel.

- **Transmission Loss** - Transmission Loss (TL) is a sonar parameter defined as the accumulated decrease in acoustic intensity when an acoustic pressure wave propagates outwards from a source [29]. Transmission Loss experienced by a narrow-band acoustic signal along a distance r is given by Urick's propagation model [28] as follows:

$$\text{Transmission Loss}(r, f, T, D) = k 10 \log r + \alpha_t (f, D, T) \times r \times 10^{-3} \quad (3.1)$$

In (3.1), the first term refers to spreading loss and second term refers to absorption loss. Here, k is the spreading factor, and α_t is the absorption coefficient with unit dB/km, r is the range in meters, f is the frequency of operation in kHz, D is depth in meters, T is temperature of water in $^{\circ}C$.

The first term in (3.1) represents the spreading loss, and the second term represents the absorption loss. The spreading factor k accounts for geometrical spreading, which refers to spreading of sound energy as a result of the expansion of the wavefronts. There are two types of geometrical spreading: spherical ($k = 2$), characterizing the deep water communication, and cylindrical ($k = 1$), characterizing a shallow water communication.

The absorption coefficient can be expressed empirically, using Thorp's formula [202] which gives α_t in dB/Km for frequency f in kHz as follows :

$$\alpha_t = 0.11 \frac{f^2}{1 + f^2} + 44 \frac{f^2}{4100 + f} + (2.75 \times 10^{-4})f^2 + 0.003 \quad (3.2)$$

This formula is generally valid for frequencies above a few hundred Hz. For lower frequencies, the following formula may be used:

$$\alpha_t = 0.002 + 0.11 \frac{f^2}{1 + f^2} + 0.011f^2 \quad (3.3)$$

Thorp's formula provides an approximation of TL as a function of frequency (depth is not considered here). More accurate results can be obtained by considering the various important mechanisms of sound attenuation in ocean such as viscous absorption and ionic relaxation. Several different formulas are available in literature for calculating absorption coefficient α_t such as Fisher and Simmons [208], Francois and Garrison [209, 210], Schulkin and Marsh [211], and Ainslie and McColm [212]. These equations have the comparable accuracy as well as computational requirements. For illustration purposes, Francois and Garrison equation has been used in this chapter for the calculation of TL.

In [210], authors have provided the equation for calculating sound absorption in water depending on acoustic frequency, pressure, acidity, temperature and salinity. This equation is valid for the frequency range of 100 Hz to 1 MHz. Here, total absorption is calculated as summation of three terms 1) contribution of boric acid, 2) contribution of magnesium sulfate and 3) contribution to absorption by pure water component.

For the acoustic frequency f in kHz, the absorption coefficient α_t can be calculated in dB/km as follows:

$$\alpha_t = \frac{A_1 P_1 f_1 f^2}{f^2 + f_1^2} + \frac{A_2 P_2 f_2 f^2}{f^2 + f_2^2} + A_3 P_3 f^2 \quad (3.4)$$

The first term in (3.4) represents the contribution to absorption by boric acid, second term represents the contribution by magnesium sulphate and third term refers to contribution due to pure water. Terms A_1 , A_2 and A_3 represent the boric acid, magnesium sulphate and pure water component respectively (in dB km⁻¹ kHz⁻¹). f_1 , f_2 and f_3 represent the relaxation

frequency for boric acid, magnesium sulphate and pure water (in kHz). P_1 , P_2 and P_3 represent the depth pressure for boric acid, magnesium sulphate and pure water respectively.

Boric acid contribution can further be calculated as follows:

$$A_1 = \frac{8.86}{c} \times 10^{(0.78\text{pH}-5)} \quad (3.5)$$

$$P_1 = 1 \quad (3.6)$$

$$f_1 = 2.8 \times \left(\frac{S}{35}\right)^{0.5} \times 10^{\left(\frac{4-1245}{\theta}\right)} \quad (3.7)$$

where c is the sound speed in m/s, given by

$$c = 1412 + 3.21 T + 1.19 S + 0.0167D \quad (3.8)$$

In (3.8), T is the temperature in $^{\circ}C$, $\theta = 273 + T$, S is salinity in ppt, and D is the depth in meters.

Magnesium sulfate contribution can be calculated as follows:

$$A_2 = 21.44 \times \frac{S}{c} \times (1 + 0.025T) \quad (3.9)$$

$$P_2 = 1 - 1.37 \times 10^{-4}D + 6.2 \times 10^{-9}D^2 \quad (3.10)$$

$$f_2 = \frac{8.17 \times 10^{\left(\frac{8-1990}{\theta}\right)}}{1 + 0.0018(S - 35)} \quad (3.11)$$

The contribution of pure water is given as follows:

$$A_3 = \begin{cases} 4.937 \times 10^{-4} - 2.59 \times 10^{-5}T + 9.11 \times 10^{-7}T^2 \\ \quad - 1.5 \times 10^{-8}T^3 \quad (For T \leq 20^{\circ}C) \\ 3.964 \times 10^{-4} - 1.146 \times 10^{-5}T + 1.45 \times 10^{-7}T^2 \\ \quad - 6.5 \times 10^{-10}T^3 \quad (For T > 20^{\circ}C) \end{cases} \quad (3.12)$$

$$P_3 = 1 - 3.83 \times 10^{-5}D + 4.9 \times 10^{-10}D^2 \quad (3.13)$$

From (3.1) and (3.4), transmission loss can be calculated for deep water. Figure 3.3 and Figure 3.4 present the total TL in deep water for a range of 500 m and 10 m for

communication at a depth of 1000 m in water. Here, frequency has been varied from 20 kHz to 140 kHz. Salinity value has been taken as 35 ppt. Acidity value pH has been considered as 8. Figures have been plotted for various temperature values.

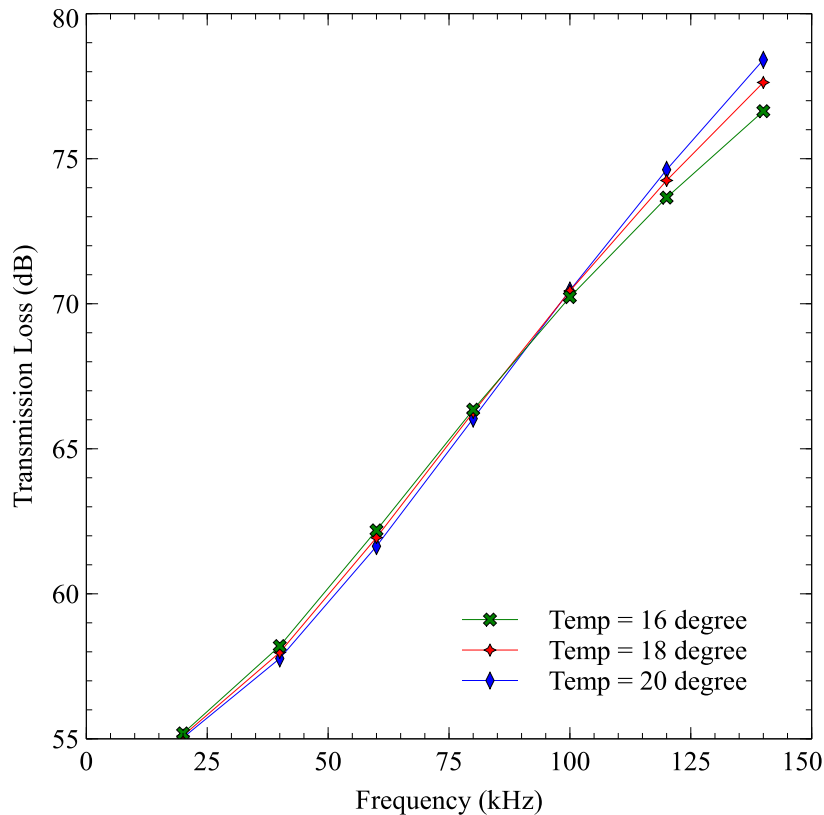


Figure 3.3. Transmission Loss (TL) for range of 500 m at water depth = 1000 m.

Figures illustrate that TL increases with increase in operating frequency as well as with the increase in communication range (or distance). This is the peculiar property of underwater acoustic communication which limits the usable bandwidth.

By choosing the operating frequency as 30 kHz, Figure 3.5 and Figure 3.6 have been plotted to illustrate the impact of depth on TL. Figure 3.5 has been plotted for the 500 m range of communication at various depths, whereas Figure 3.6 has been plotted for 10 m range at varying depth in ocean. Here, salinity value has been taken as 35 ppt and acidity value pH has been considered as 8. It can be observed that, TL significantly decreases when the depth increases. This effect is more pronounced for frequencies higher than few hundred kHz.

Figure 3.7 indicates the TL for varying distance of communication at various depths in ocean. For this graph, frequency of operation has been taken as 30 kHz, salinity value has

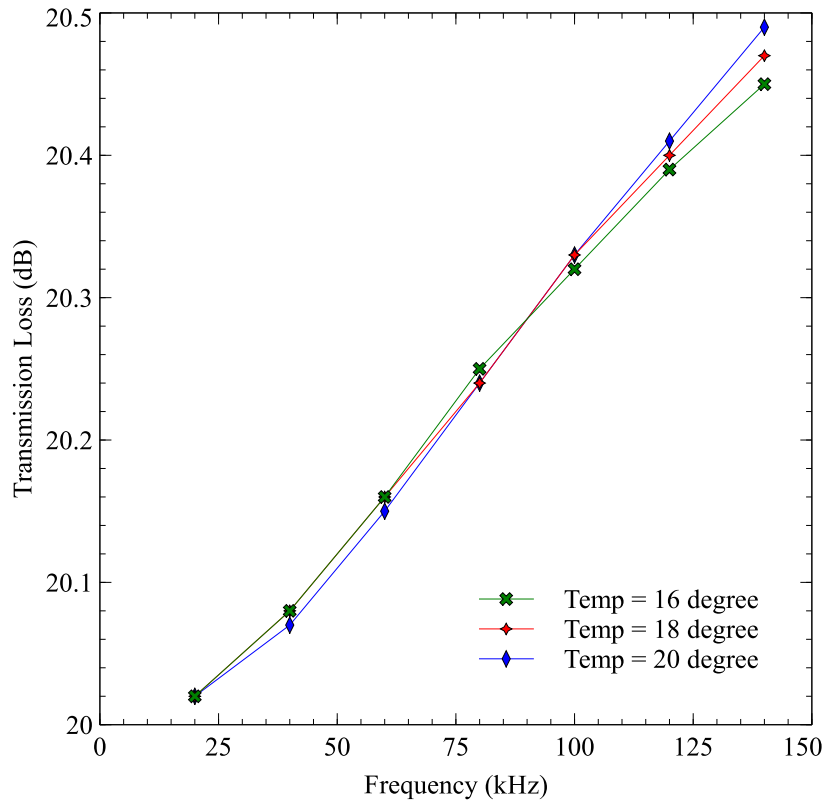


Figure 3.4. Transmission Loss (TL) for range of 10 m at water depth = 1000 m.

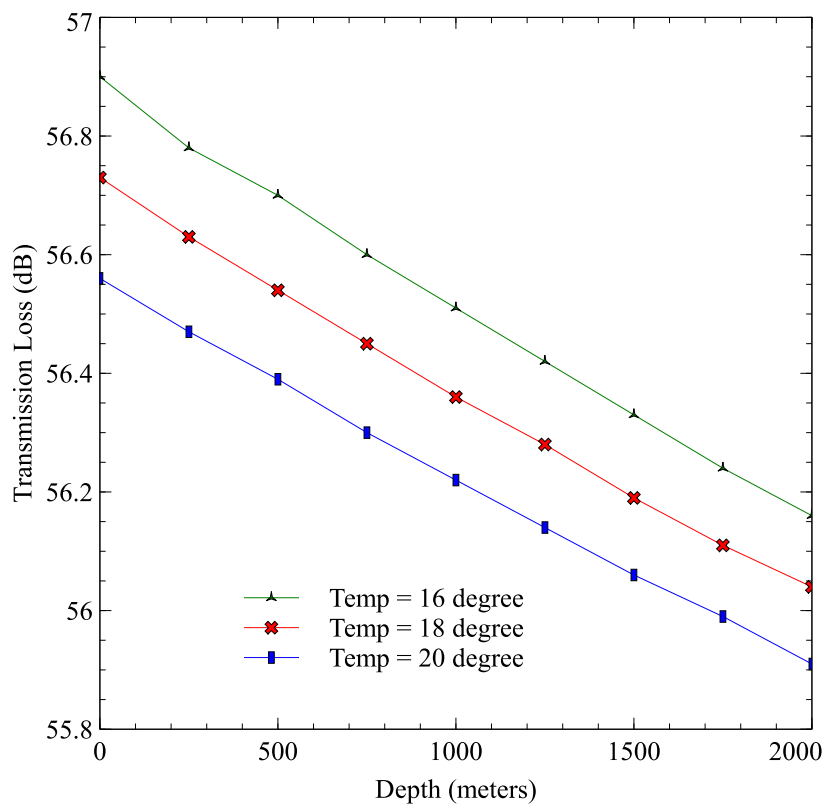


Figure 3.5. Transmission Loss (TL) for range of 500 m with operating frequency = 30 kHz.

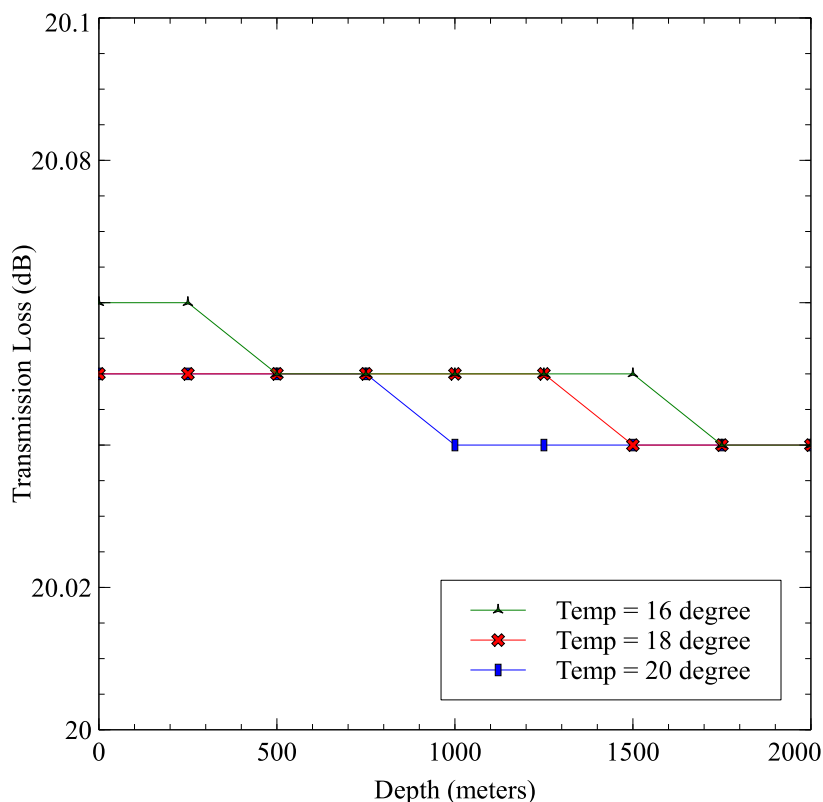


Figure 3.6. Transmission Loss (TL) for range of 10 m with operating frequency = 30 kHz.

been taken as 35 ppt, acidity value pH has been taken as 8. This figure also reiterates the fact that TL increases with increasing range but decreases with the increase in depth.

- **Noise** - The ambient noise in the ocean can be modeled using four sources: turbulence, shipping, waves, and thermal noise. Most of the ambient noise sources can be described by Gaussian statistics and a continuous power spectral density (p.s.d.). The following empirical formula give the p.s.d. of the four noise components in dB re μPa per Hz as a function of frequency in kHz [204].

$$10 \log N_t(f) = 17 - 30 \log f \quad (3.14)$$

$$10 \log N_{sh}(f) = 40 + 20 (sh - 0.5) + 26 \log f - 60 \log (f + 0.03) \quad (3.15)$$

$$10 \log N_w(f) = 50 + 7.5w^{1/2} + 20 \log f - 40 \log (f + 0.4) \quad (3.16)$$

$$10 \log N_{th}(f) = -15 + 20 \log f \quad (3.17)$$

Where N_t is the noise due to turbulence, N_{sh} is the noise due to shipping, N_w is the noise due to wind, and N_{th} represents the thermal noise, f is the frequency in kHz. Turbulence

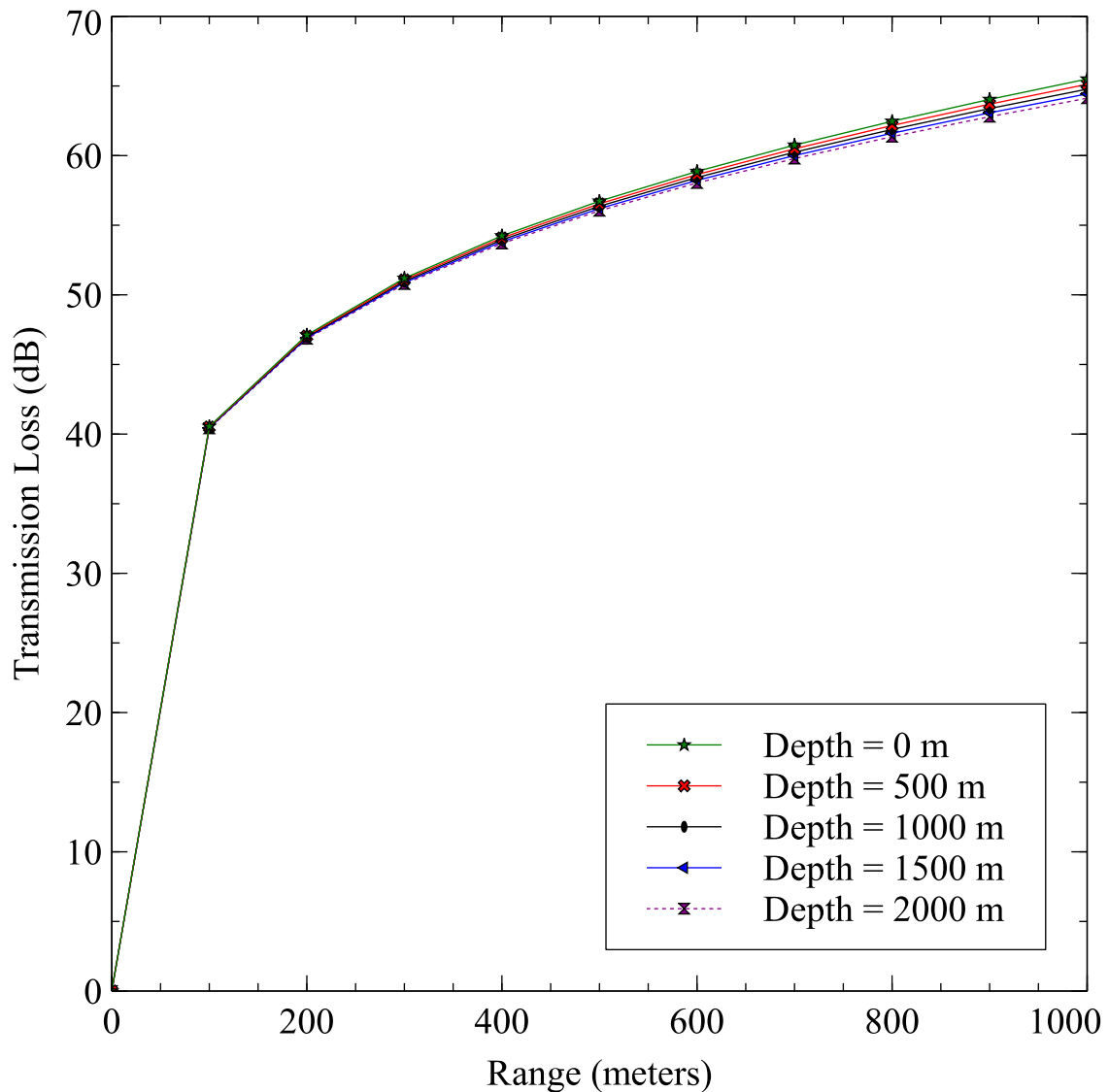


Figure 3.7. Transmission Loss (TL) for varying distance of communication at various depth in ocean.

noise influences only the very low frequency region ($f < 10$ Hz). Noise caused by distant shipping is dominant in the frequency region 10 Hz - 100 Hz, and it is modeled through the shipping activity factor sh , whose value ranges between 0 and 1 for low and high activity, respectively. Surface motion, caused by wind-driven waves (w is the wind speed in m/s) is the major factor contributing to the noise in the frequency region 100 Hz - 100 kHz (which is the operating region used by the majority of acoustic systems). Finally, thermal noise becomes dominant for $f > 100$ kHz. The overall noise power spectral density for a given frequency f is then:

$$N(f) = N_t(f) + N_{sh}(f) + N_w(f) + N_{th}(f). \quad (3.18)$$

Power spectral density of ambient noise has been illustrated in Figure 3.8. It can be easily observed that, for increase in wind speed and shipping activity, ambient noise in ocean increases.

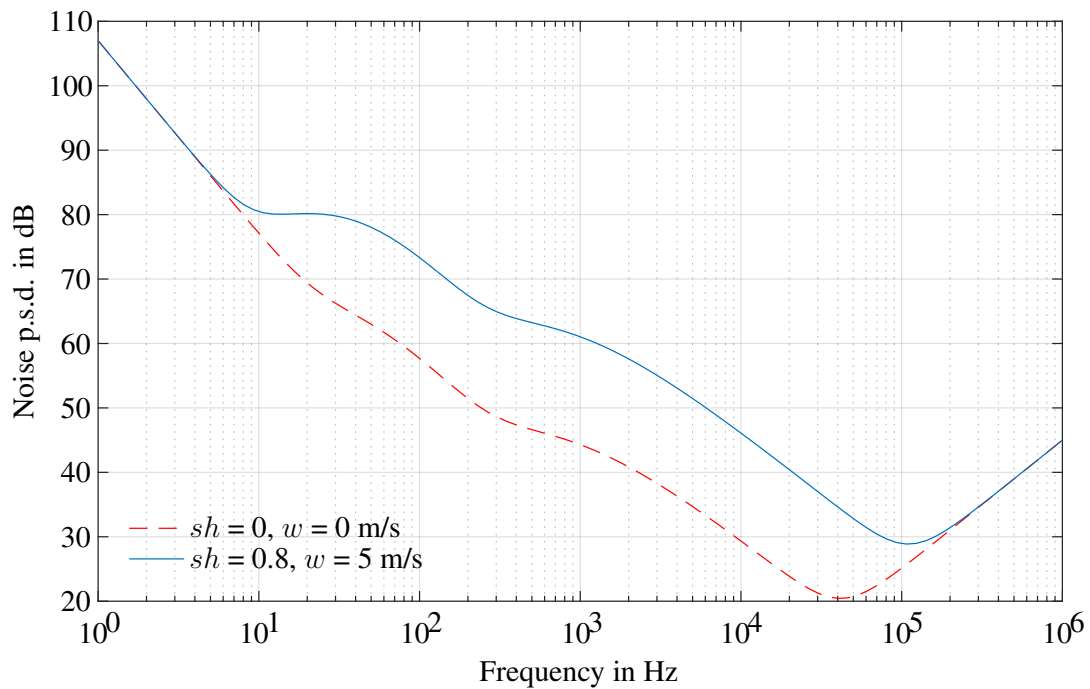


Figure 3.8. Power spectral density of ambient noise in ocean [31].

- **High Delay and Delay Variance** – Speed of sound in water is four times faster than the speed of sound in air, but is five orders of magnitude smaller than the speed of light. Slow speed of sound affects the performance of communication system and the design of network protocol stack as well. Acoustic propagation speed depends on factors such as salinity, depth and temperature. Temperature of ocean further depends on the weather, time of the day, environmental factors and depth. This temperature variability has a large impact on acoustic speed. Underwater acoustic propagation speed can be modeled using various formula from literature such as i) Coppens formula [213], ii) Del Grosso equation [214], iii) UNESCO equation [215] or iv) Mackenzie [216].

A nine term equation of Mackenzie for calculation of sound speed in ocean is as follows:

$$c = 1448.96 + 4.591T - 5.304 \times 10^{-2}T^2 + 2.374 \times 10^{-4}T^3 + 1.340(S - 35) + 1.630 \times 10^{-2}D + 1.675 \times 10^{-7}D^2 - 1.025 \times 10^{-2}T(S - 35) - 7.139 \times 10^{-13}TD^3 \quad (3.19)$$

Where c is the speed of sound in water expressed in m/s, T is the temperature in $^{\circ}C$, S is the salinity in ppt, and D is the depth in meters.

A vertical profile of sound speed in sea water using Mackenzie equation has been shown in Figure 3.9. Here, Salinity is 35 ppt and surface temperature has been varied from $16^{\circ}C$ to $20^{\circ}C$. In Figure 3.10, variation of sound speed has been shown with respect to varying salinity. Here, surface temperature has been assumed as $18^{\circ}C$ and salinity has been varied from 30 to 40 ppt.

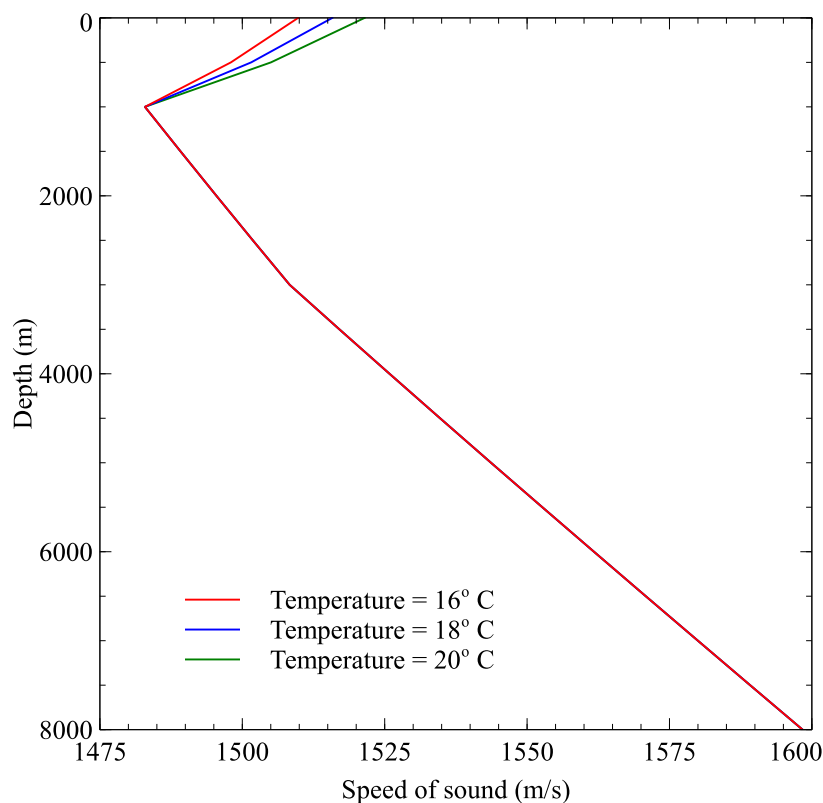


Figure 3.9. A vertical profile of sound speed in seawater as the function of depth for varying surface temperature [28].

Based on temperature variations, ocean water can be divided in three layers. Water depth upto 200 m is the first layer. This layer is most variable because of wind mixing and daytime

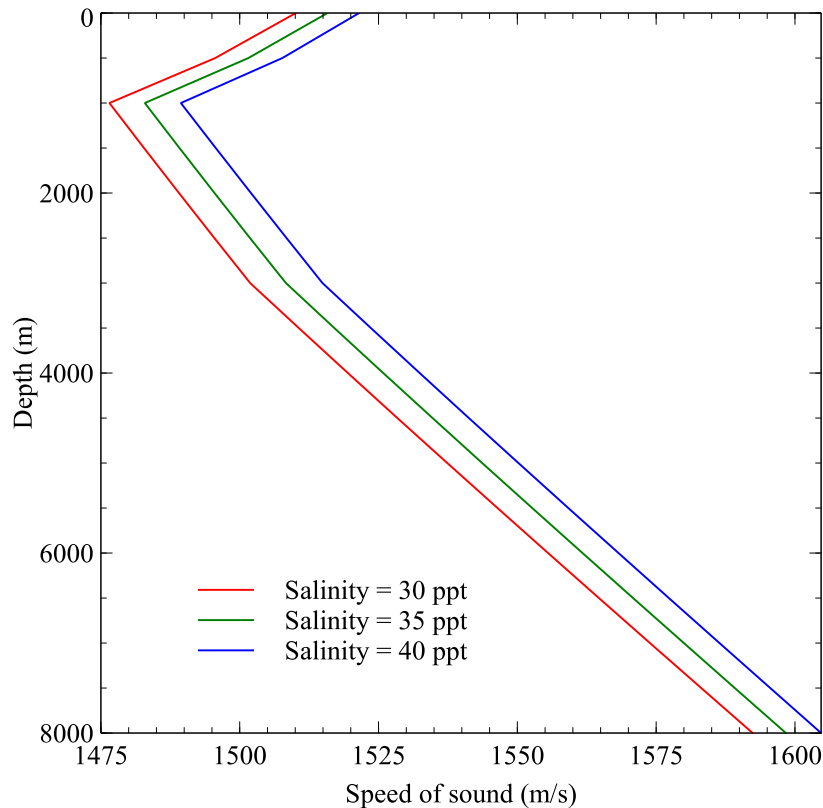


Figure 3.10. A vertical profile of sound speed in seawater as the function of depth for varying salinity [28].

heating effect. This layer usually consists of two sublayers, surface layer (from surface to around 100 m) and seasonal thermocline layer (from 100 m to 200 m). Sound speed has negative gradient in this layer (typically in summer season), since the water at surface is warmer than water below. Sound speed decreases with depth in this layer. Second layer is the main (permanent) thermocline layer upto around 1000 m and is associated with minimum sound speed. Sound speed decreases primarily because of decrease in temperature. Below this layer, from 1000 m till the ocean floor is the third layer termed as deep isothermal layer. Temperature in this layer is almost constant. Linear increase in sound speed in this layer is mainly because of increase in depth giving rise to positive speed gradient.

Sound speed in ocean water is highly variable depending upon factors such as temperature, salinity and depth. Usual range of sound speed under various operating conditions is from 1450 m/s to 1550 m/s. In literature, an average value of 1500 m/s is considered for simplicity. This slow speed results in large and variable propagation delay (0.67 s/km), and reduces the throughput of system considerably [4].

- **Multipath** – An acoustic signal can travel via multiple paths from transmitter to receiver. In case of shallow water communication, sound propagates by repeated reflections from sea-surface and sea-floor, causing multipath propagation. Signal suffers higher attenuation because of these reflections, based on frequency, incident angle, and sediment type [29]. In case of deep water, surface and floor reflections can be neglected if the transmitter and receiver are not located near the sea-surface or floor. However, the spatially varying sound speed can create wave refractions leading to significant multipath effect in deep water. A channel impulse response of a multipath channel is given by following equation,

$$h(\tau, t) = \sum_{i=1}^P \gamma_i(t) \exp(j\theta_i(t)) \delta(\tau - \tau_i(t)) \quad (3.20)$$

wherein P is the number of multiple paths, γ_i and θ_i are the amplitude and phase of the signal arrived by i th path, and τ_i is the time-delay of the i th path.

From the P multiple distinct paths, there can be multiple arrivals of the signal at the receiver. If the delay spread (difference between first and last arrival) is significantly larger than the symbol duration, then it might lead to very high ISI.

- **Doppler spread** : Motion of the transmitter or receiver contributes to changes in the underwater acoustic channel response. This occurs through Doppler effect, which causes frequency shifting as well as additional frequency spreading. The magnitude of the Doppler effect is proportional to the ratio $a = \frac{v}{c}$ of the relative transmitter-receiver velocity to the speed of sound. Because the speed of sound is very low compared to the speed of electromagnetic waves, motion-induced Doppler distortion of an acoustic signal can be extreme [205]. For example, assuming the device velocity as $v = 2$ m/s, carrier frequency $f = 30$ kHz, and sound speed as $c = 1500$ m/s, Doppler shift frequency is $f_d = \frac{vf}{c} = 40$ Hz. If the symbol duration $S_d = 0.1$ ms, (or a data signal bandwidth of 1 kHz), then the normalized Doppler per symbol is $f_d S_d = 0.04$. This example shows that a large Doppler shift reduces the coherence time of channel. It also indicates that in case of mobile networks, channel variation needs to be accounted on symbol to symbol basis.

From the above analysis of acoustic propagation, it is evident that the aspects of underwater acoustic communication are significantly different than the terrestrial RF communication. These

characteristics of physical layer along with network architecture have a huge impact on design of protocol stack of UASN. In this chapter, a physical layer has been designed for the proposed three-dimensional network architecture. Details of this design has been provided in the next section.

3.5 Design and modeling of physical layer of UASN

The details of proposed three-dimensional architecture for UASN have been provided in Section 3.3. This is a structured and static deployment for a long-term ocean column monitoring. All nodes have been assumed to have equal capabilities, making the network homogeneous in nature. Vertical long-range communication link is dominant in the network as compared to small-range horizontal links at various depths. For this reason, communication links having direct path for communication have been assumed in the further analysis. Multipath phenomenon which mainly affects the horizontal links has been neglected. Also, Doppler effect is assumed to be absent. In the underwater acoustic channel, TL increases with frequency as well as distance. A short-range system can exploit more bandwidth and data rate as compared to long-range system. Hence, a multi-hop communication is preferred in UASN over single hop communication over a longer range. In the proposed network architecture, multi-hop communication links have been incorporated inherently in the design. Nodes send their data to the Base-Station via Cluster-Head node using multi-hop, and cannot reach to Base-Station by direct single hop communication.

For designing physical layer, parameters such as i) frequency of operation, ii) modulation technique, iii) data rate, and iv) power level of source has to be specified. Calculation of Signal-to-Noise Ratio (SNR) is of paramount importance in the physical layer design. For determining SNR, a passive equation of SONAR (SOund NAvigation and Ranging) [28] can be utilized. According to this equation, SNR at receiver side can be obtained as follows:

$$\text{SNR} = \text{SL} - \text{TL} - \text{NL} - \text{DI} \quad (3.21)$$

where SL is source level of transmitter, TL is transmission loss, NL is ambient noise level in ocean and DI is directivity index. DI is usually set to zero value. In underwater acoustic domain, these values are expressed in dB re μPa , where the reference value of 1 μPa amount to $0.67 \times$

10^{-18} W/m^2 . Commonly, shorthand notation dB is used in the literature to signify dB re μPa . In rest of the work, this shorthand notation has been used for simplicity.

The level of received signal at receiver can be obtained by using equation of Received Level (RL) as follows:

$$\text{RL} = \text{SL} - \text{TL} \quad (3.22)$$

where RL is expressed in dB as,

$$\text{RL} = 10 \log \frac{\text{received intensity}}{\text{reference intensity}} \quad (3.23)$$

Reference intensity in underwater acoustic is taken as $0.67 \times 10^{-18} \text{ W/m}^2$.

In a similar way, Source Level (SL) is expressed in dB as,

$$\text{SL} = 10 \log \frac{\text{transmission power intensity}}{\text{reference intensity}} \quad (3.24)$$

where transmission power intensity (I_t) at 1 m distance from the source is expressed in terms of transmission power P_t (Watt) as,

$$I_t = \frac{P_t}{4\pi \times 1m} \quad (3.25)$$

Source Level can be expressed in terms of radiated source power P_t (at 1 m distance from the source) directly as,

$$\text{SL} = 10 \log \left(\frac{10^{12} \rho c P_t}{4\pi} \right) \quad (3.26)$$

where $\rho = 1000 \text{ Kg/m}^3$ is water density and c is speed of sound in water.

For a receiver with narrow band B around operating frequency f , the SNR is expressed as,

$$\text{SNR} = \text{SL} - \text{TL} - \text{NL} - 10 \log B \quad (3.27)$$

Based on the analysis of acoustic communication shown in Section 3.4, the optimal frequency of communication can be around 20-30 kHz. Here, the operating frequency of 30 kHz has been assumed. For this frequency, TL is around 57 dB (max) at various depths with a range of 500 m on vertical link. TL is significantly lower at around 20 dB at various depth locations for a

communication range of 10 m on horizontal link. Noise level NL is also minimum at around 40 dB (considering $w = 5$ m/s and $sh = 0.5$).

In Table 3.1, the values of SL have been provided for various power levels of transmission P_t in Watts. RL for the range of 500 m (of vertical link) has been given in column 4, whereas RL for range of 10 m (of horizontal link) has been given in column 5. In columns 6 and 7, SNR for 500 m and 10 m have been given respectively considering bandwidth of receiver as 10 kHz.

Sr.No	P_t (W)	SL (dB)	RL (dB) (for range of 500 m)	RL (dB) (for range of 10 m)	SNR (dB) (for range of 500 m)	SNR (dB) (for range of 10 m)
1	1	170.77	113.77	150.77	33.77	70.77
2	10^{-1}	160.77	103.77	140.77	23.77	60.77
3	10^{-2}	150.77	93.77	130.77	13.77	50.77
4	10^{-3}	140.77	83.77	120.77	3.77	40.77
5	10^{-4}	130.77	73.77	110.77	-6.23	30.77
6	10^{-5}	120.77	63.77	100.77	-16.23	20.77
7	10^{-6}	110.77	53.77	90.77	-26.23	10.77

Table 3.1. SNR calculations for various transmission power levels.

From Table 3.1, it can be observed that the same power level P_t results in different achievable SNR at 500 m and 10 m range of communication. This difference is mainly due to large TL over 500 m range as compared to 10 m range. For example, using power level $P_t = 0.01$ W leads to SNR of 13.77 dB at 500 m distance whereas it gives SNR value of 50.77 dB at a distance of 10 m. If SNR of 13.77 dB is sufficient for decoding of packet at receiver (depending on modulation technique), then the SNR of 50.77 dB in horizontal plane is extra-ordinarily large and wasteful. It suggests that use of same power level to communicate till 10 m and 500 m distance is not advisable. In the three-dimensional UASN, a CH node is required to communicate with other CH node over vertical link of distance 500 m. For communication on this link, a CH node can choose higher power level. But, a CH node should adopt to lower power level for communication with cluster member nodes in horizontal plane at (maximum) distance of 10 m. If the SNR of 10 dB is assumed to be sufficient for the decoding of packet, then CH node can choose power level $P_t = 0.01$ W for CH to CH communication and $P_t = 10^{-6}$ W for CH to CN communication. These power level values are chosen assuming the NL is around 40 dB and that the low-power acoustic modems are available. In case of higher noise levels, the transmission power levels have to be increased accordingly.

From these power levels, probability of bit error has been calculated for different modulation techniques such as BFSK, BPSK, QPSK, and 16-QAM. Data rate of 10 kbps has been assumed throughout the calculations. The ratio of E_b (Energy per bit) to N_o (Noise power spectral density) can be obtained from chosen SNR as follows:

$$\frac{E_b}{N_o} = \left(10^{\frac{\text{SNR(dB)}}{10}} \right) \frac{B_N}{R} \quad (3.28)$$

where $B_N = B = 10$ kHz is the noise bandwidth. Using (3.28), E_b/N_o value obtained for power level $P_t = 0.01$ W at a distance of 500 m is 23.82. Similarly, E_b/N_o value for power level $P_t = 10^{-6}$ W at a distance of 10 m is 11.94.

The results of different modulation techniques have been tabulated in Table 3.2. It can be observed from the table, that the probability of bit error for BFSK and 16-QAM is higher as compared to BPSK and QPSK. From these observations, a modulation scheme of BPSK is recommended. QPSK scheme also gives approximately same performance of that of BPSK but it has more stringent phase synchronization requirement.

Sr. No.	Modulation scheme	Probability of bit error P_b	P_b at 500 m range	P_b at 10 m range
1	BFSK	$P_b = Q \sqrt{\frac{E_b}{N_o}}$	5.3×10^{-7}	2.7×10^{-4}
2	BPSK	$P_b = Q \sqrt{\frac{2E_b}{N_o}}$	2.6×10^{-12}	5.0×10^{-7}
3	QPSK	$P_b = Q \sqrt{\frac{2E_b}{N_o}} \left[1 - \frac{1}{2} Q \sqrt{\frac{2E_b}{N_o}} \right]$	2.6×10^{-12}	4.99×10^{-7}
4	16-QAM	$P_b = \frac{3}{4} Q \sqrt{\left(\frac{4}{5}\right) \frac{E_b}{N_o}}$	1.2×10^{-6}	1.9×10^{-4}

Table 3.2. Probability of bit error for different modulation schemes.

Use of different power levels for different range in the proposed three-dimensional UASN also leads to possibility of parallel intra-cluster communication. It can be easily proved using the given recommended case. From Table 3.1, it can be seen that, for transmission power level $P_t = 10^{-6}$ W, SNR is 10.77 dB at 10 m horizontal distance. Signal-to-Interference-plus-Noise Ratio (SINR) at a receiver node in one cluster, due to ongoing parallel communication in adjacent clusters can be

calculated using (3.29) as follows:

$$\text{SINR} = \frac{\text{Power of received Signal}}{\sum_{i=1}^{i_0} (I_i) + \text{Noise power}} \quad (3.29)$$

wherein, I_i stands for power of interference signal from i th path.

Using the values from Table 3.2, SINR at any CN node in presence of parallel ongoing communication in two adjacent interfering clusters is obtained as 10.75 dB. The SINR value is almost equivalent to SNR, stating that the interference from adjacent clusters is negligibly small. In this way, an effective strategy of power level management is in place to exploit spatial reuse of channel. It leads to improved channel utilization of otherwise bandwidth limited acoustic channel.

In this way, it has been demonstrated that the parameters of physical layer (such as frequency of operation, power levels, data rate and modulation techniques) can be obtained using analytical modeling for the given three-dimensional network architecture. For the deployment of three-dimensional UASN with different dimensions, this procedure can be used to obtain corresponding design parameters of physical layer. A brief concluding remarks from the design of network architecture and physical layer have been provided in Section 3.6.

3.6 Summary

In this chapter, a three-dimensional UASN architecture has been proposed for the application of ocean column monitoring. The proposed architecture consists of number of nodes deployed at varying depth in the ocean in the form of clusters. Motivation and assumptions in the design of this architecture have been stated in the chapter. A specific case of network architecture have been used to showcase the coverage of ocean column having 2000 m depth and 10 m radius. Description of various categories of nodes such as Base-Station, Cluster-Head and cluster member nodes has been provided in detail.

For designing a physical layer for this network architecture, a deep water channel model has been studied in detail from the available literature. By suitably choosing the dimensions of column and locations of nodes of this UASN, the design parameters of physical layer of protocol stack have been provided.

To summarize, the desirable properties of an acoustic modem are as follows:

- Power levels - Dynamically adoptable power levels.
- Frequency - Frequency of operation in range of 25 kHz to 30 kHz.
- Range - Communication range of 500 m to 1000 m.
- Depth - Depth of operation as maximum 2500 m.
- Modulation Technique - Preferably BPSK.
- Data rate - Data rate of around 10 kbps.

Chapter 4

Development of Testbed for UASN

4.1 Introduction

Numerous applications of UASN are possible in the scientific, commercial and military domain. These applications have promoted a huge amount of research in the area of underwater acoustic networking, but this research is mostly limited to theoretical investigations and software simulation work. It is a common understanding in the research community that “Simulations are doomed to succeed.” Actual testing of protocol stack on hardware testbed is important but it necessitates either the development of indoor laboratory based testbed or an access to outdoor water bodies such as lakes or rivers.

The advantages and disadvantages of building an indoor testbed can be stated as follows:

Advantages of developing indoor testbed are :

- Accessibility - Indoor testbed developed in the laboratory is available and accessible all the time for testing. Access to testbed is not dependent on weather condition or time of the day.
- Proximity of resources - All resources such as power supply, hardware tools are readily available. Uninterrupted long-duration testing can be ensured.
- Security - Testing on an indoor testbed is safe and secure. There is no risk of losing the instruments during long-run testing. No special guarding is required.

- Easy control - An indoor testbed offers an easy control in deploying and changing topology of the network.

Disadvantages of indoor testbed design are :

- Cost - Underwater acoustic modems are costlier as compared to terrestrial WSN nodes. In addition, cost of building a water tank is also huge.
- Requirement of space - Water tank and the accessories consume laboratory space.
- Maintenance - Stagnant water in the tank has to be replaced periodically.

On the other hand advantages and disadvantages of outdoor testing are stated as follows:

Advantages of outdoor testing of UASN are :

- Realistic environment - Outdoor testing in the water sources such as rivers, lakes, dams can provide more realistic testing before the actual at-sea deployment. Effect of seasonal and daily variations on the protocols can be analyzed.

Disadvantages of outdoor testing of UASN are :

- Inaccessibility - Performing on actual water sources might need permissions from the local authorities. Water resources which are ideal for testing may be at farther distance and hence may consume extra time in transportation. Severe weather conditions such as rains, high temperature can also be barriers in outdoor testing.
- Infrastructure - Outdoor testing would need certain infrastructure such as boat, power supplies, availability of trained divers. Huge cost might be incurred in transportation and availing these infrastructure along with high cost of acoustic modems.
- Safety feature - Performing long-duration outdoor testing may not be always safe. Cautious guarding of all resources would be required. Recovery of deployed nodes would not be always possible.

In this work, an indoor laboratory based testbed set-up has been developed for UASN [217]. Objective in testbed development is to build an autonomous UASN. Since this testbed can be used to identify issues related to implementation of the network protocol stack on actual hardware set-up; it serves as a platform for validation of theoretical and simulation results of protocols developed for UASN. The effect of limitations of hardware on the protocol stack can also be tested.

In this chapter, the details of main components of testbed have been provided. Procedure of setting-up experiments on this testbed has been discussed in detail. The chapter has been organized as follows: In Section 4.2, details of various components of the testbed has been provided. In Section 4.3, software set-up of the testbed has been discussed. Different topologies and protocols deployed on this testbed have been listed in the Section 4.4. Characteristics of testbed have been listed in Section 4.5. Brief summary of chapter has been provided in 4.6.

4.2 Components of testbed

4.2.1 The underwater acoustic node

The underwater acoustic node has been built using Simple Acoustic Modems (SAMs) [180] , TelosB motes [218], interfacing units, and power supplies.

- **Simple Acoustic Modem (SAM)** - SAM is a miniature acoustic modem. It is developed by Desert Star Systems LLC, an underwater technology manufacturer located in Monterey Bay, California. These modems have been chosen for the indoor testbed development because these are simple, small-size and low-cost. Details of SAM modem have been provided in Appendix B.
- **TelosB motes** : TelosB is an ultra low power wireless module for use in sensor networks, monitoring applications, and rapid application prototyping. TelosB mote can be programmed using TinyOS environment. TinyOS is a tiny (fewer than 400 bytes), flexible operating system built from a set of reusable components that are assembled into an

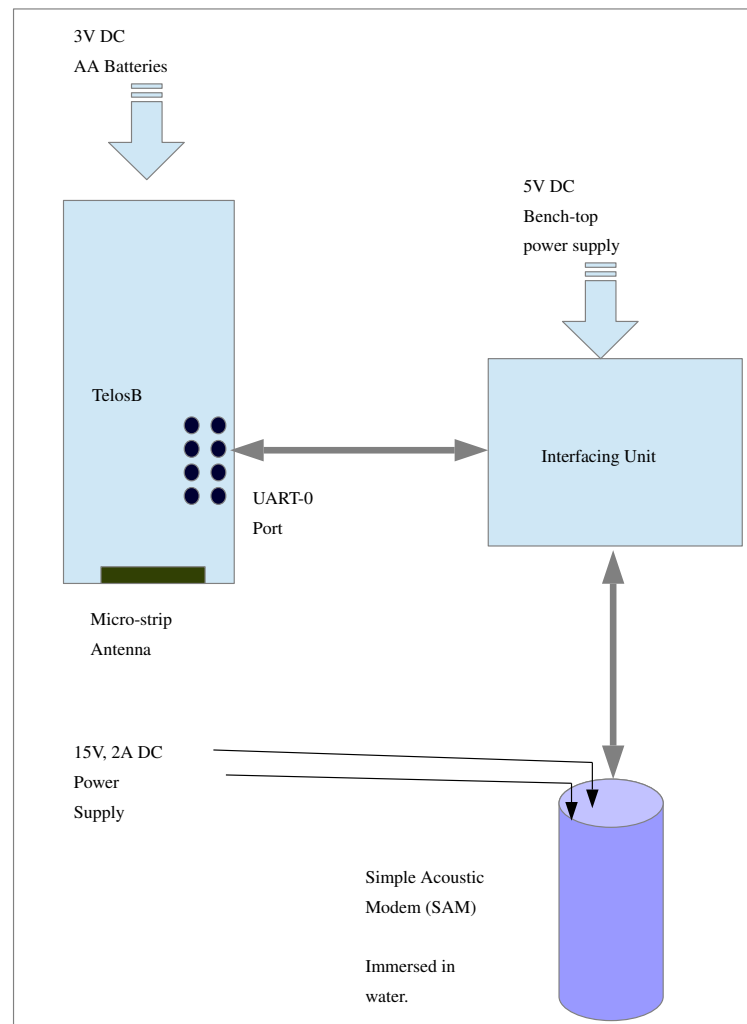


Figure 4.1. Building an underwater acoustic node using Commercial Off-The-Shelf components (COTS).

application-specific system [219]. Source codes for various network protocols can be developed in TinyOS and installed on the mote for testing. In the testbed set-up, TelosB mote has been connected to the modem via UART pins.

- **Interfacing Units** : The interfacing units have been utilized for converting the data from the RS232 compatible format to TTL format for the TelosB mote (and vice-versa.)

Interconnection of various components used to build an acoustic node has been shown in Figure 4.1. As shown in this figure, a TelosB mote has been connected to the acoustic modem using interfacing unit. Usually, in terrestrial wireless scenario, TelosB mote makes use of a embedded micro-strip antenna and IEEE 802.15.4 compliant RF transceiver module CC2420 for wireless

RF communication. By default, the TinyOS network protocol stack for serial communication on the TelosB is routed through UART1 port. In this testbed, TelosB mote has been connected to acoustic modem using UART0 expansion pins to achieve underwater acoustic communication. UART0 is a part of USART module (Universal Synchronous/Asynchronous Receiver/Transmitter) on TelosB. To enable the connection via UART0 port, the serial communication protocol stack (SerialAM stack) in TinyOS has been modified. With this connection, any information passed by TelosB mote on UART0 port is modulated by the acoustic modem for transmission in underwater medium. Similarly, any information received by modem is demodulated and passed on to TelosB for processing. In this way, an underwater acoustic node has been developed using the Commercial Off-The Shelf components (COTS).

TelosB mote allows bus arbitration, and is itself an example of complex arbitration. Bus arbitration is a process of sharing a bus between multiple requesting modules. In TelosB, USART module provides SPI bus to UART0 and CC2420. To use CC2420 antenna, SPI bus arbitration is essential. Such a mechanism is required only for one acoustic node which acts as a Base-Station node. This Base-Station node has an underwater acoustic modem to communicate with other acoustic nodes underwater, and also has a RF antenna to communicate with ground based control station. One Base-Station node has been designed in this testbed.

UART0 pins of TelosB operates at CMOS (Complementary Metal Oxide Semiconductor technology) level, while acoustic modem works on RS-232 level. A signal conversion is essential for this connection. The interfacing unit has been used for this purpose.

During the testing, it has been assumed that the data is available from on-board sensors for transmission. By connecting the actual oceanographic sensors to TelosB and ensuring a proper hermetic sealing for the complete platform, a simple underwater acoustic sensor node can be developed. SAM has a depth rating of 300 m and a range of 1000 m in ideal, deep ocean. These acoustic nodes can be deployed at various depths using a suitable buoyancy mechanism. A Base-Station node can be deployed on the sea-surface. This Base-Station node can collect sensor data from various nodes deployed underwater. After collecting this data, the Base-Station node can transmit the data to the ground based control station using RF communication. It can also receive control information from the ground based control station using RF communication.

In the testbed, several acoustic nodes have been utilized to deploy a network. The initial version of testbed developed using small glass tank filled with water has been shown in Figure 4.2. The communication architecture deployed on this testbed set-up has been illustrated in Figure 4.3. A larger testbed has been later developed using a water tank of 800 liters capacity. This water tank dimensions are 2625 mm (length), 1375 mm (width), and 320 mm (height). This current version of testbed, wherein underwater acoustic modems can be placed according to the desired topology of network architecture for protocol stack testing have been shown in Figure 4.4.

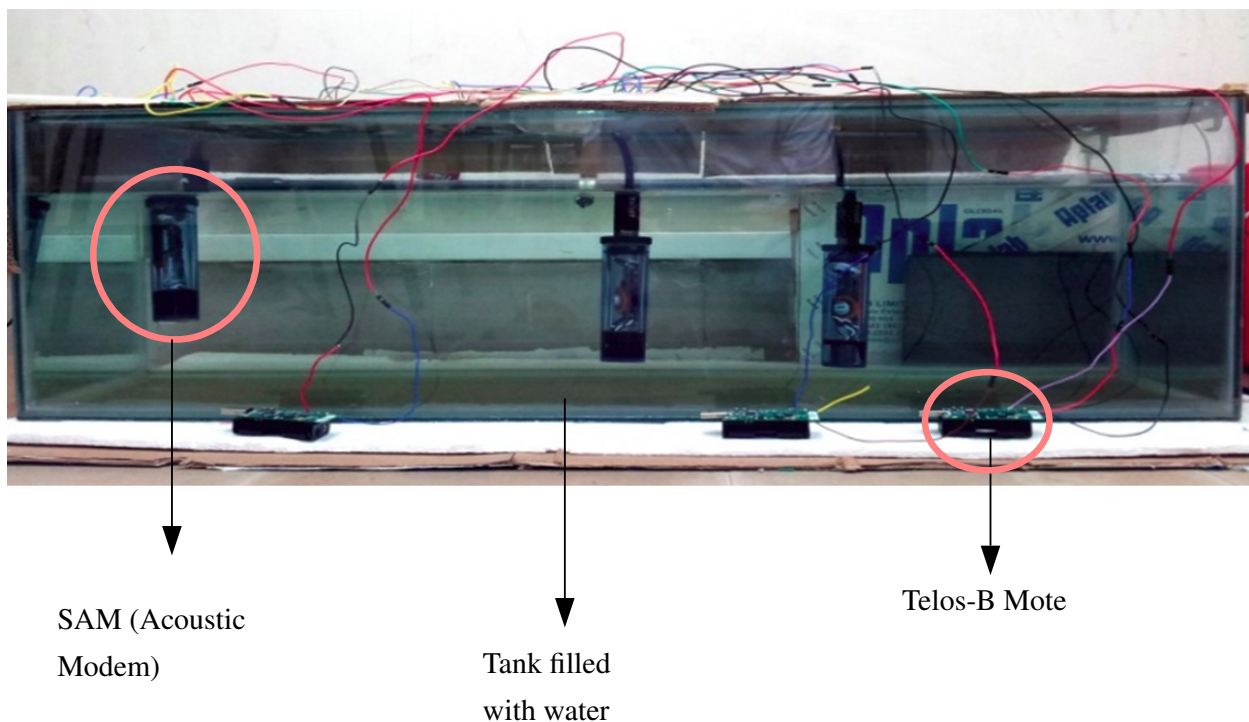


Figure 4.2. Initial version of UASN testbed set-up.

4.3 Software set-up

The software set-up of the testbed has been done in two steps :

1. Programming TelosB motes
2. Configuration of SAMs

In this section, details of the steps involved in setting up testbed have been provided.

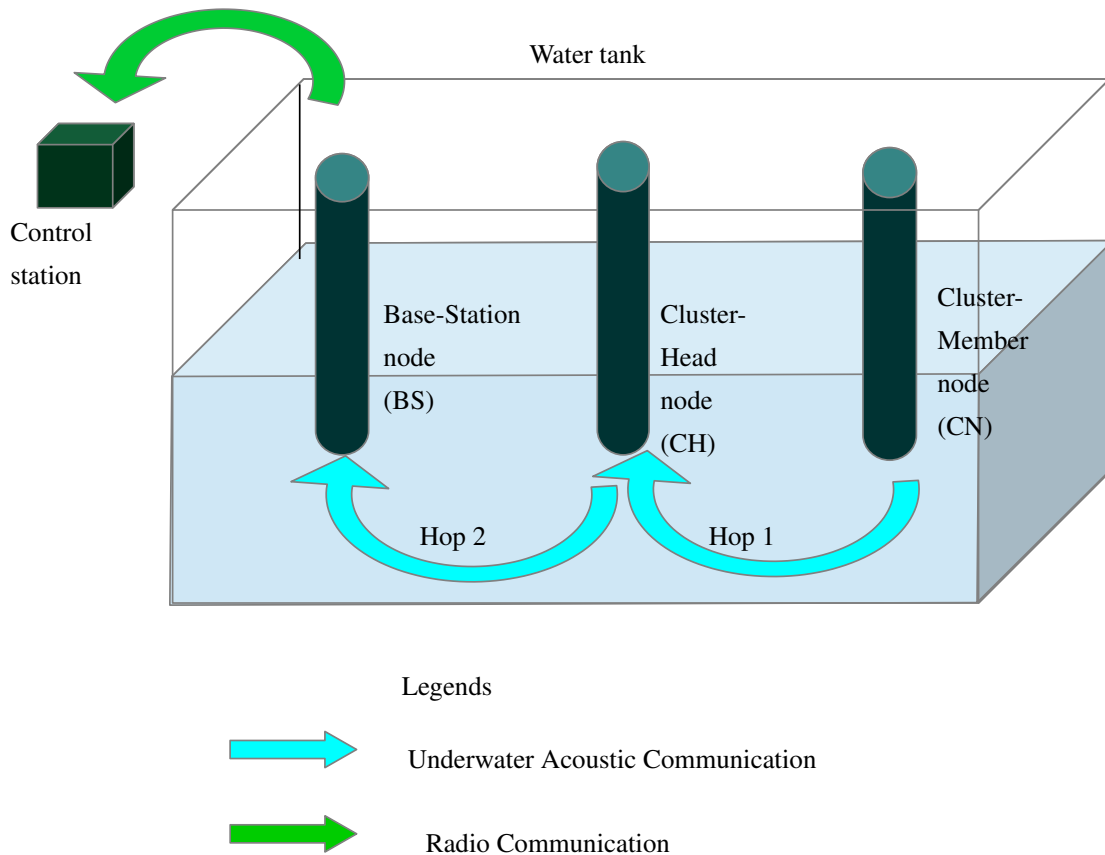


Figure 4.3. Communication architecture deployed on UASN testbed.

1. **Programming TelosB nodes** – The required protocol has been coded in TinyOS for implementation on TelosB nodes. These source codes have been written in nesC language. nesC (network embedded systems C) is a component-based, event-driven programming language used to build applications for the TinyOS embedded operating system. NesC is built as an extension to the C programming language with components “wired” together to run applications on TinyOS. A programmed TelosB node has been used as a host device for SAM.
2. **Configuration of SAMs** – SAMs have to be properly initialized in order to successfully operate in a given surrounding. To configure the modem, initially it has been connected with a Personal Computer (PC), using Sea-link cables provided with the modems. This Sea-link cable has a 5-pin DIN male connector on one end of the cable which can be connected to the SAM. Other end of the cable has DB9 connector as well as a adapter connector for DC power supply. Using DB9 to USB converter, the modem has been connected to PC for programming the SAM. SAM communicates with the host and receives power using the

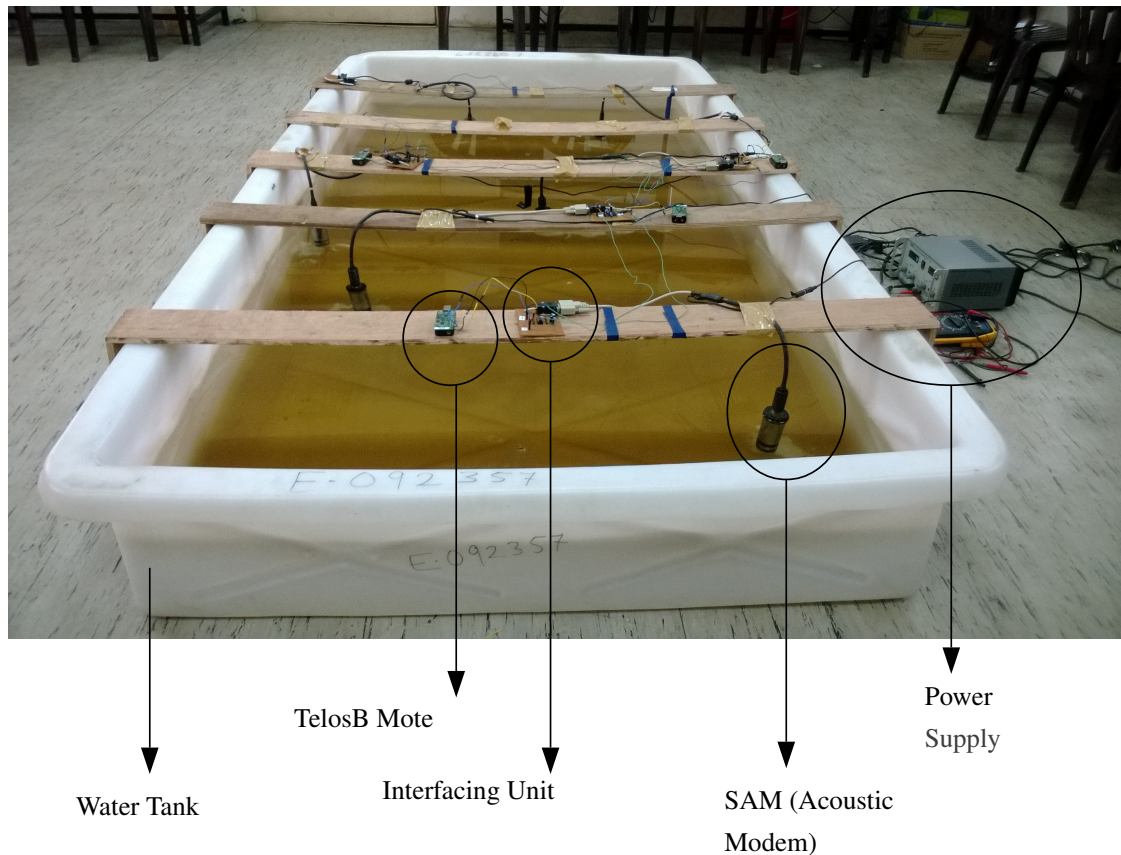


Figure 4.4. Current version of UASN testbed set-up.

5-pin DIN connector. SAM requires around 21 mA current in receive mode, whereas in transmit mode, upto 2 A current is drawn from the source. After power-up, SAM sends an identifier string through the serial data port. The data string identifies the software version and the mode of operation. The modem is ready for operation immediately after the transmission of this string.

The modem has a data mode and a control mode. Data Mode is used to transmit and receive data. Control Mode is used to change modem operation parameters.

SAM is automatically set in control mode after power-up, to program the modem. Simple commands can be used to configure various parameters of the modem, such as acoustic transmit data speed, acoustic receive data speed and receiver detection threshold.

For a single channel operation, SAM supports following transmit data speeds,

- Command S4: User data throughput = 3 bps. Raw bit transmission speed =5 bps.
- Command S5: User data throughput = 8 bps, Raw bit transmission speed =13 bps.

- Command S6: User data throughput = 23 bps, Raw bit transmission speed =38 bps.

Corresponding receive data speeds are,

- Command R4: User data throughput = 3 bps. Raw bit reception speed =5 bps.
- Command R5: User data throughput = 8 bps, Raw bit reception speed =13 bps.
- Command R6: User data throughput = 23 bps, Raw bit reception speed =38 bps.

The sensitivity of the receiver modem can also be adjusted by setting the detection threshold. A low detection threshold makes the modem very sensitive. This increases the maximum data exchange range, but at the same time makes the modem more susceptible to noise. A high detection threshold means greater immunity to noise but a shorter transmit range. In theory, a doubling of the threshold setting will cut the maximum range in half. In reality, this effect is less pronounced, depending on the acoustic environment.

The operation of SAM is very simple:

- Any data received via the serial data interface is transmitted via sonar.
- Any data received via sonar is transmitted via the serial data interface.

The serial data interface operates at 4800 baud with a configuration as 8-N-1 (8 data bits, no parity bit, 1 stop bit).

By using two acoustic modems, a two-way acoustic data link can be configured and tested. Once this data link is successfully established, PC can be replaced with the programmed TelosB motes. One TelosB mote can be programmed as source node to transmit the sensed data after periodic intervals, and other TelosB mote can be programmed as sink node to receive and process data.

4.4 Different topologies and protocols implemented on UASN testbed set-up

Following topologies and protocols have been successfully implemented on this testbed set-up:

1. Multi-hop communication architecture [220].

2. Multi-hop Tri-message time synchronization protocol [221, 222].
3. DCB-TDMA (Dynamic Cluster-Based Time Division Multiple Access) MAC protocol [223].

In the following subsection, the implementation of multi-hop communication architecture on the testbed set-up has been discussed in detail. The protocol of multi-hop Tri-message time synchronization has been described in Chapter 5. Implementation of DCB-TDMA MAC protocol has been described in Chapter 6.

4.4.1 Multi-hop communication architecture

Objective in this topology deployment has been to set up a multi-hop communication architecture of UASN. Multi-hop topology is preferred in wireless networks due to energy efficiency. The shorter the communication distance, lesser is the energy consumed. Additionally, multi-hop transmission can achieve higher spectral efficiency than single-hop transmission over a longer distance in UASN.

For this deployment, three underwater acoustic nodes have been developed using the Commercial Off-The Shelf (COTS) components as described in Section 4.2. Each underwater acoustic node consists of SAM, TelosB mote, interfacing unit and power supplies as shown in Figure 4.1. All nodes can communicate underwater using acoustic communication links. SAMs have been configured in mode S4 for transmission and in mode R4 for reception, with the user data throughput as 3 bps. A single frequency channel of 33.8 kHz has been used. One of the node has been designed to have both the type of communication links, that is, RF CC2420 for radio communication and underwater acoustic link for underwater communication. A software bridge between radio link and acoustic link has been established using TinyOS. This bridge has ensured smooth transition of payload through two different packet formats. TelosB mote programmed with this feature has been connected to the acoustic modem, providing this additional feature to the node. In the set-up, this node has been used as a Base-Station node (BS) as shown in the Figure 4.3. Remaining two nodes have been used in the formation of cluster. One node (center node in the figure), acts as a Cluster-Head (CH) node, whereas other node (on right corner in set-up), acts as a cluster member (CN) node. Actual testbed set-up installed in the laboratory has been shown

in Figure 4.2. For the actual underwater deployment, this architecture can be visualized as shown in the Figure 4.5. Figure shows the CN, CH and BS nodes deployed in ocean and connected using underwater acoustic links. RF link has been shown between BS node and ground based control station.

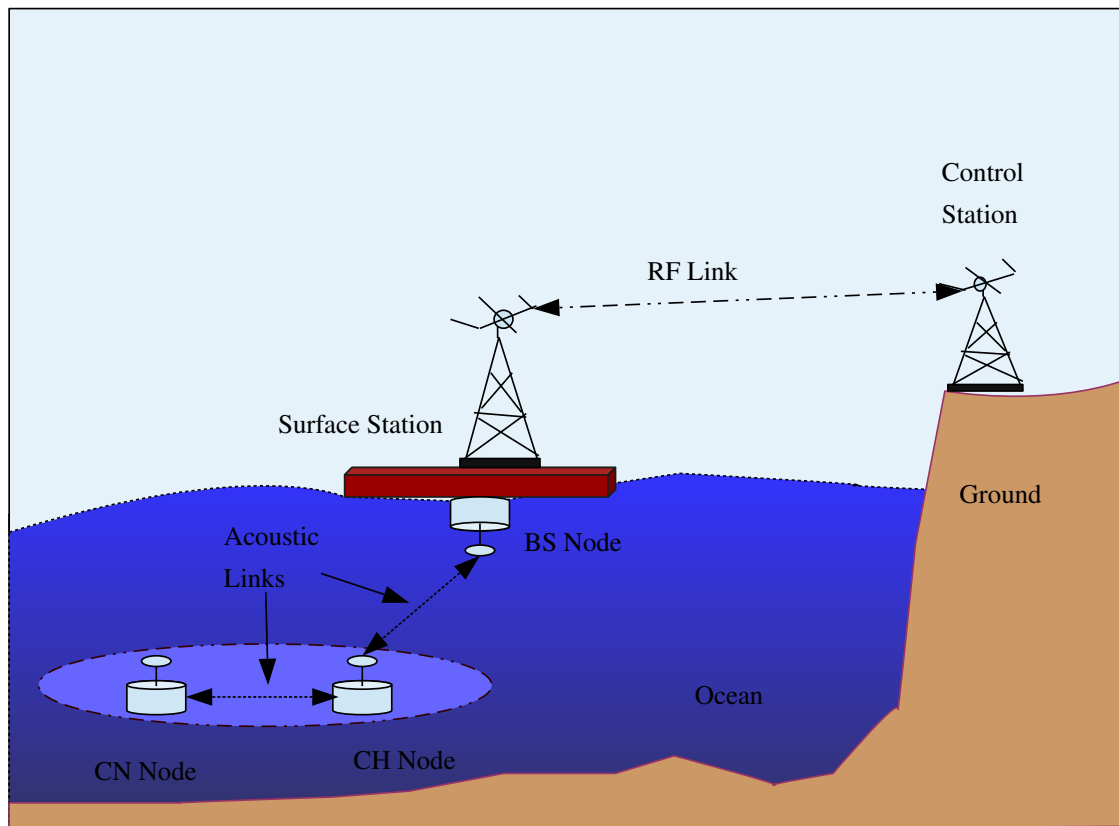


Figure 4.5. Possible real-time deployment of multi-hop network architecture of UASN testbed.

Traffic on this multi-hop network has been set up as follows:

1. Data packet has been created by CN node using data from sensors. A packet of 20 bytes has been transmitted at regular intervals of 3 minutes by this node. For example, data packet has been transmitted by CN node at 3 minute, 6 minute, 9 minute and so on. Acoustic link from CN node to CH node has been termed as hop 1.
2. Data packet has also been created by CH node using the sensing information available from the sensors on board. Data packet has been received by CH node from the CN node at regular intervals of time. Averaging of the data values has been performed by the CH node and a packet of 20 bytes has been transmitted to the BS node. CH node also has repetition period of 3 minutes, but the data communication has been staggered by 1 min as compared to CN

node. For example, transmission of data from CH node to BS node has been scheduled at 4 minute, 7 minute, 10 minute and so on. This staggering has been used to accommodate the slow data rate and slow speed of acoustic propagation. Acoustic link from CH node to BS node has been termed as hop 2.

3. Once the BS node has collected the data from the CH node, it has been scheduled to transmit the data immediately to the control station. TelosB mote connected to the PC has been used as a control station. Control station has been used to analyze the received data.

In the multi-hop architecture deployed on the testbed, the communication and networking parameters have been set up as follows:

- Packet Size (P_S) = 20 bytes.
- Average propagation Speed of acoustic signal in water (c) = 1500 m/s.
- Bit Rate (R) = 3 bps.

For the delay calculations, parameters such as Packet Transmission Time (\hat{T}), Propagation Time (\hat{P}) and Packet Delivery Time (\hat{D}) can be obtained using following equations,

$$\hat{T} = \frac{P_S}{R} \quad (4.1)$$

$$\hat{P} = \frac{r}{c} \quad (4.2)$$

$$\hat{D} = \hat{T} + \hat{P} \quad (4.3)$$

In this scenario,

$$\hat{T} = 53.33 \text{ s.}$$

\hat{P} value has been neglected, since the modems are very close in the testbed set-up.

Hence, it has been considered that, $\hat{D} = \hat{T} = 53.33 \text{ s.}$

As the \hat{D} is 53.33 s, one minute staggering has been essential between transmission of CN and CH nodes.

In Table 4.1, the results of ten different runs on this testbed have been tabulated. The data rate and sleep duration has been varied in every test-run. The sleep duration is the timing period between

completion of one multi-hop cycle (that is, control station receiving the data) and start of next cycle (that is, CN node transmitting data to CH node) when the nodes are idle. Overall duration of test-run has been chosen randomly.

Test run	CN Node - No. of packets Transmitted	CH node- No. of packets Received	CH node- No. of packets Transmitted	BS node- No. of packets Received	Hop 1 (%PDR)	Hop 2 (%PDR)
1	47	41	47	21	87.23	44.68
2	46	39	46	41	84.78	89.13
3	50	20	50	38	40.00	76.00
4	17	14	16	11	82.35	68.75
5	37	35	37	30	94.59	81.08
6	34	26	33	19	76.47	57.58
7	16	16	16	15	100.00	93.75
8	15	12	15	10	80.00	66.67
9	14	12	14	12	85.71	85.71
10	55	27	53	36	49.09	67.92
				Average	78.02	73.13

Table 4.1. Percentage Packet Delivery Ratio (PDR) of multi-hop communication architecture.

It can be observed from the table that the underwater acoustic link is very poor and unreliable. For example, in test-run 3, CN node has transmitted 50 packets, but CH node has received only 20 packets leading to mere 40% Packet Delivery Ratio (PDR). On the other hand, in the test-run 7, all 17 packets transmitted by CN node has been received by CH node on hop 1, providing 100% PDR. Similarly on the hop 2, worst case result is for test-run 1, whereas, best case is test-run 7 from the perspective of PDR on hop 2. It should be noted that, the number of packets transmitted by the CH node is same as the number of packets transmitted by the CN node in most of the cases. This is because the CH node is configured to transmit its own data packet, even if it has not received the packet from CN node. Average PDR of hop 1 is 78.02% and that of hop 2 is 73.13%. Overall average PDR on testbed is 75.58%.

The poor and unreliable performance of multi-hop architecture is mostly because of i) reverberation effect of small water tank, ii) nature of underwater acoustic communication, and iii) lack of time synchronization between the nodes.

In a testbed set-up, there is no probing possible to analyze the actual reason for packet drop. One more underwater acoustic node has been used as the interceptor (or listener node) to observe the transmission and reception of packets. Certain anomalies might have been introduced because of this observing node itself.

4.5 Characteristics of testbed

Two different versions of testbed have been shown in Figure 4.2 and Figure 4.4. On the current version of testbed, various topologies of network can be tested. Testbed has provided ease of access and provision for testing various protocols. This testbed also has certain limitations, stated as follows:

- Size of testbed is small. Nodes reside close to each other in a testbed. Effect of propagation delay can not be adequately observed. Depth setting is not available on the testbed.
- Acoustic modem does not have a facility to explore sleep-wake pattern.
- Acoustic modem does not provide adaptive power level control. Frequency of operation is also fixed.
- Reverberation effect from the testbed boundaries is high. To measure the impact of this reverberation, hydrophone would be required for measuring signal levels.

4.6 Summary

In this chapter, motivation of developing an indoor laboratory based testbed set-up has been stated. For deploying an indoor testbed, a prototype of underwater acoustic sensor node has been built using Commercial Off-The Shelf (COTS) components such as TelosB mote, acoustic modem and

power supplies. It has been shown that this prototype can be extended to form an underwater acoustic sensor node as a real-time end-product for deployment at ocean. Detailed description of development of this prototype along with the software set-up has been provided.

Details of implementation of multi-hop communication architecture on this testbed has been given in this chapter. It has been observed that the Packet Delivery Ratio (PDR) of the network on testbed set-up is around 75%. Also, different protocols implemented on this testbed set-up has been stated in this chapter. Overall, this testbed has served as a unique, novel and cost-effective validation platform for testing the UASN protocols at the laboratory.

Chapter 5

Time Synchronization for UASN

5.1 Introduction

Nodes in a sensor network collaborate with each other to perform tasks such as i) distributed sensing, ii) coordinated or adaptive sensing, iii) sensor fusion, iv) localization, v) task scheduling, vi) task execution, vii) security, viii) routing, and ix) topology control [224–226]. Energy awareness can be achieved in an optimum manner by having collaboration among different nodes and resources. For example, energy can be conserved by using a sleep-wake pattern for the sensor nodes [227–229]. To achieve effective sleep-wake pattern and the many collaborative tasks mentioned above, a common idea of time-base across sensor network is necessary. The process of achieving and maintaining a common time-base across sensor network is termed as time synchronization.

The desired accuracy of time synchronization depends upon the requirement of the application and the specific purpose of synchronization. Certain applications of sensor networks need the right chronology of the events to be detected (for example, target tracking application), while some applications need the absolute time of the events to be made available (for example, disaster prevention system). In a sensor-actuator network, correct reference of timing is required for operation of the actuator. Many important protocols in the sensor network (for example, TDMA based MAC protocol) cannot work without achieving time synchronization [230]. For such cases, time synchronization protocol in sensor network is indispensable.

Achieving time synchronization in the initialization phase of sensor network is very challenging, since the nodes might be turned “ON” at different time intervals. Even after achieving time synchronization in initialization phase, deviations from reference time might occur later because of i) inaccuracies in node’s clock, ii) changes in topology, iii) inclusion or failure of nodes, iv) mobility of the nodes and so on. For this reason, synchronization needs to be performed at regular intervals.

In sensor networks, a multi-hop communication link is preferred over direct single hop communication. Usually, a Base-Station can reach to all the nodes in a network using multi-hop links. Base-Station also has access to a precise clock. Base-Station can act as a reference node and can synchronize the network nodes within its communication distance over single hop. These nodes can then provide time synchronization to other nodes in their respective single hop. In this manner, all the nodes in the network can be time synchronized by using multi-hop communication links.

In this work, a multi-hop time synchronization protocol has been developed for UASN. This protocol is the extension of “Tri-message time synchronization protocol” described in [137]. A new protocol developed has been termed as “Multi-hop Tri-message time synchronization protocol.” This protocol has been implemented in simulation as well as on the hardware testbed set-up of UASN. Details of these implementations have been provided in this chapter.

This chapter is organized as follows: In Section 5.2, various issues and challenges in design and development of time synchronization protocols have been discussed. In Section 5.3, detailed analysis of two prominent time synchronization protocols of UASN, namely TSHL (Time Synchronization for High Latency) protocol [136] and Tri-message protocol has been discussed. Comparative evaluation of these protocols has been provided using MATLAB simulation tool. Details of the proposed extension of Tri-message time synchronization protocol for multi-hop topology has been provided in Section 5.4. This proposed extension has been simulated for multi-hop topology using MATLAB. The results of simulation and analysis of results has been provided in this section. Further, multi-hop Tri-message time synchronization has been implemented on hardware testbed. Details of this implementation along with results of implementation have been provided in Section 5.5. Brief summary of the chapter has been provided in Section 5.6.

5.2 Background and related work

5.2.1 Time in distributed systems

The clocking circuits of nodes in a sensor network are typically implemented using low-cost crystals. Moreover, the nodes of WSN may be turned on at different times during initialization phase, which can create the difference between the clocks of the nodes. This difference in the timing of two nodes is termed as offset. Assuming clock x to be a perfect clock (or real time), another clock y is said to be correctly operating if $y = x$. If clock y is characterized as $y = x + \alpha$, then value α is referred as offset. Offset values can be calculated by a single message exchange between two nodes. Clock skew is the difference in the frequency or rate of two clocks. If the clock y is characterized as $y = \beta x + \alpha$, ($\beta \neq 1$) then value β is termed as clock skew. Figure 5.1 represents these different characteristics.

Most common clock hardware on nodes is not very accurate because the frequency (that makes time increase) is never exactly right. Even a frequency deviation of just 0.001% would cause a clock error of about one second per day. This is also a reason why the clock performance is often measured with very fine units like one part per million (ppm). Accuracies are affected by temperature, supply voltage fluctuations, aging and so on. For TelosB motes, the upper bound on clock drift is 40 ppm. This means a clock in mote can loose up to 40 microseconds in a second. Most of the time synchronization protocols ignore clock skew when synchronizing an offset. But the nodes that are synchronized to a common offset will drift out of synchronization over time because of clock rate differences. These differences can be corrected through resynchronization at regular intervals. Though most of the time synchronization protocols do not aim to provide perfect clock operation, relative differences of offset and clock skew has to be minimized, thus ensuring common reference frame.

5.2.2 Challenges in time synchronization

All network time synchronization methods rely on some sort of message exchange between nodes. A packet carrying the time-stamp message experiences variable amount of delay, between the transmitter and receiver. This variable delay is the main source of error which causes the

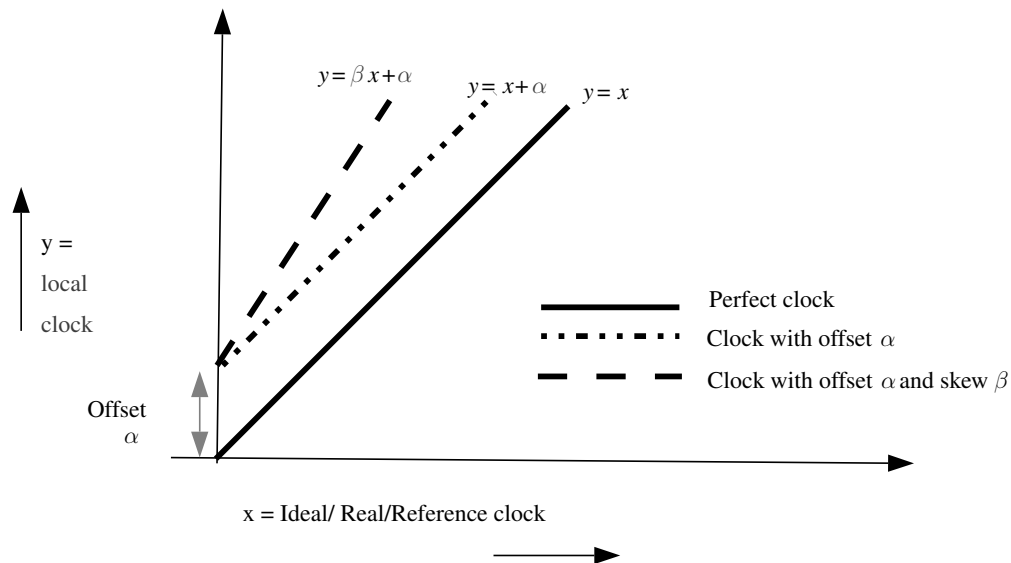


Figure 5.1. Characteristics of clock [156].

inaccuracies in the synchronization process. The sources of error in network time synchronization methods can be decomposed into following basic components [231] :

- **Software access time:** When an application requests the time value, system returns the current value to the application. The delay incurred here is dependent on the operating system, stack and loading of the system. The software access time appears on transmitter side while constructing a packet for transmission and transferring it to the network interface. Software access delay on receiver node appears while traversing the packet upwards in the protocol stack. These delays are largely variable and non-deterministic in nature.
- **Medium access time:** This is the time spent in waiting for accessing the channel for transmission. Message experiences non-deterministic delay in accessing the channel.
- **Transmission time:** This is the time required in transmission of complete packet over the channel. It is largely deterministic and dependent on transmission rate and packet size.
- **Propagation Time:** Time spent in propagation of the message to receiver is known as propagation time. If the speed of propagation is assumed constant, then this delay is deterministic. Under the large and variable propagation speed, these delays might be nondeterministic. The propagation timings might as well have path asymmetry.

- **Reception Time:** Time required in receiving the complete length of packet on receiver side is termed as reception time. This is dependent on bandwidth and packet size. Reception time is deterministic.

A delay experienced by the message in round trip time has been shown in Figure 5.2. In this example, message has been transmitted from Node A at time t_1 to Node B. Delay incurred by this message are software access time (\hat{S}_1), medium access time (\hat{M}_1), transmission time (\hat{T}_1) and propagation time (\hat{P}_{12}). On the receiving Node B, the message has been received at t_2 and the delays incurred are reception time (\hat{R}_2) and software access time (\hat{S}_2). Further, the Node B has processed and prepared ACK message at time t_3 . Considering ρ_B as the drift rate at Node B, the time difference would be $(1 \pm \rho_B)(t_3 - t_2)$. Delays experienced by the ACK message are software access time (\hat{S}_3), medium access time (\hat{M}_3), transmission time (\hat{T}_3), propagation time (\hat{P}_{21}), reception time (\hat{R}_4) and software access time (\hat{S}_4) till it is received by application at Node A. This round trip time in sending message from Node A and receiving ACK message from Node B can be given by the following equation:

$$t_4 - t_1 = \hat{S}_1 + \hat{M}_1 + \hat{T}_1 + \hat{P}_{12} + \hat{R}_2 + \hat{S}_2 + (1 \pm \rho_B)(t_3 - t_2) + \hat{S}_3 + \hat{M}_3 + \hat{T}_3 + \hat{P}_{21} + \hat{R}_4 + \hat{S}_4 \quad (5.1)$$

Time synchronization protocol has to account or eliminate these sources of error in clock timing. Synchronization protocols are evaluated using evaluation metrics such as (i) precision, (ii) scalability, (iii) energy efficiency, (iv) robustness, (v) low complexity and memory requirement [232]. Single scheme cannot satisfy all these metrics, so certain trade-offs are necessary depending on the application's requirement.

5.3 Time synchronization methods for UASN

Time synchronization protocols developed for terrestrial WSNs assume that the propagation latency is negligible. Protocols such as RBS [133] and FTSP [134] cannot be applied to the UASN scenario, since these protocols assume nearly instantaneous and simultaneous reception, a phenomenon impossible for the underwater acoustic communication. Other protocols such as TPSN [40], LTS [135] ignore the clock drift during synchronization. Synchronization phase

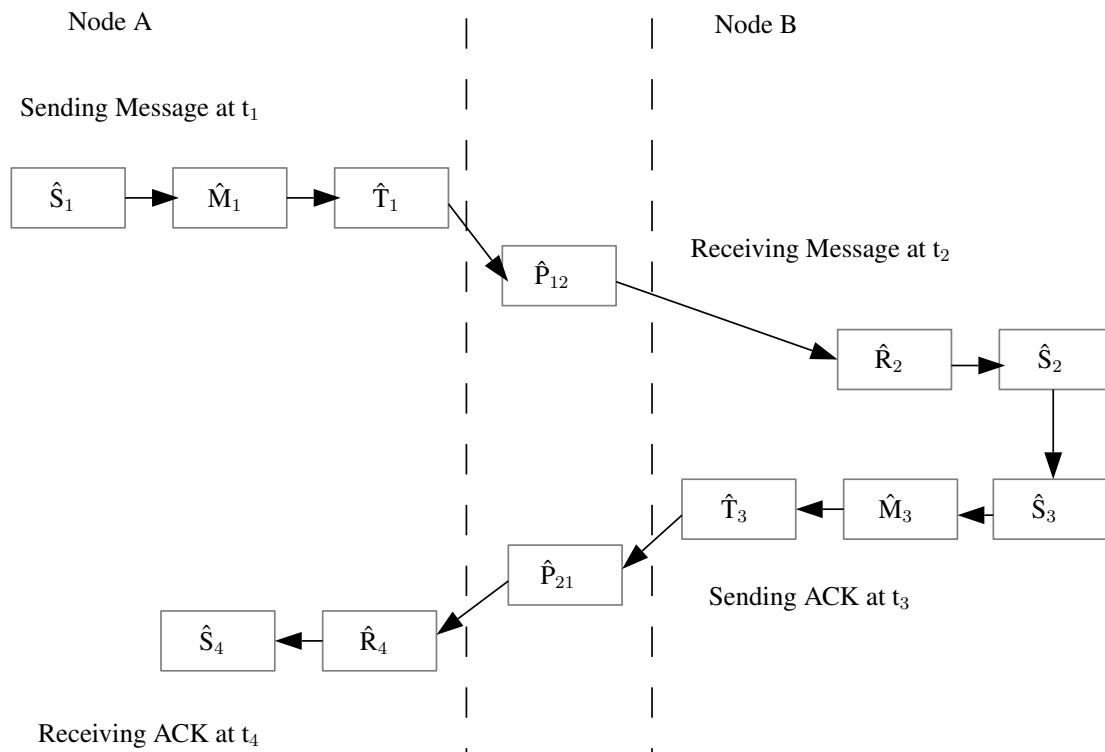


Figure 5.2. Round-trip time of message between Node A and Node B [156].

itself takes significant time for UASN networks due to high propagation delays; therefore such protocols cannot provide desired accuracy in UASN. There are certain time synchronization protocols developed particularly for high latency based underwater acoustic network. First among these approaches is TSHL protocol [136] and another prominent protocol is Tri-message time synchronization protocol [137]. These time synchronization protocols has been considered for the discussion in the following sections.

5.3.1 TSHL (Time Synchronization for High Latency) protocol

In [136], authors have introduced the protocol titled ‘Time Synchronization for High Latency’ (TSHL) that compensates for high-latency communication. This is the first protocol proposed for time synchronization in UASN. TSHL protocol consists of two phases. Phase 1 is used to model clock skew by using linear regression over multiple beacon values. In phase 2, a two way message exchange is used for correcting the clock offsets. Here, it is assumed that i) the clocks are short term skew stable and ii) time-stamping is possible at MAC layer of protocol stack.

The phases of TSHL time synchronization protocol has been shown in Figure 5.3. In this figure, Node A has been taken as the anchor node and has been used for synchronizing Node B. At time t_1 , Node A starts phase 1 by sending multiple beacons. Time t_1 corresponds to time-stamp A_1 on the Node A. (Notation A_m has been used for time-stamps of Node A, wherein m indicates the message index. Similar notation B_m has been used for Node B.) Number of beacon messages depend upon the desired accuracy in the protocol. Phase 1 of TSHL ends at time B_1 . Total time taken for phase 1 beacon message transmission has been shown in Figure 5.3 as $TSHL_{\text{phase1}}$. Node B enters into skew synchronized state after phase 1. After a time interval $TSHL_{11}$, Node B starts the two-way message exchange process. It sends a message at time B_2 to Node A. Node A receives this message at time A_2 after the propagation delay PD. Node A then transmits another message to Node B after a time interval of $TSHL_{12}$ at time A_3 . Node B receives this message at time B_3 and can correct the clock offset.

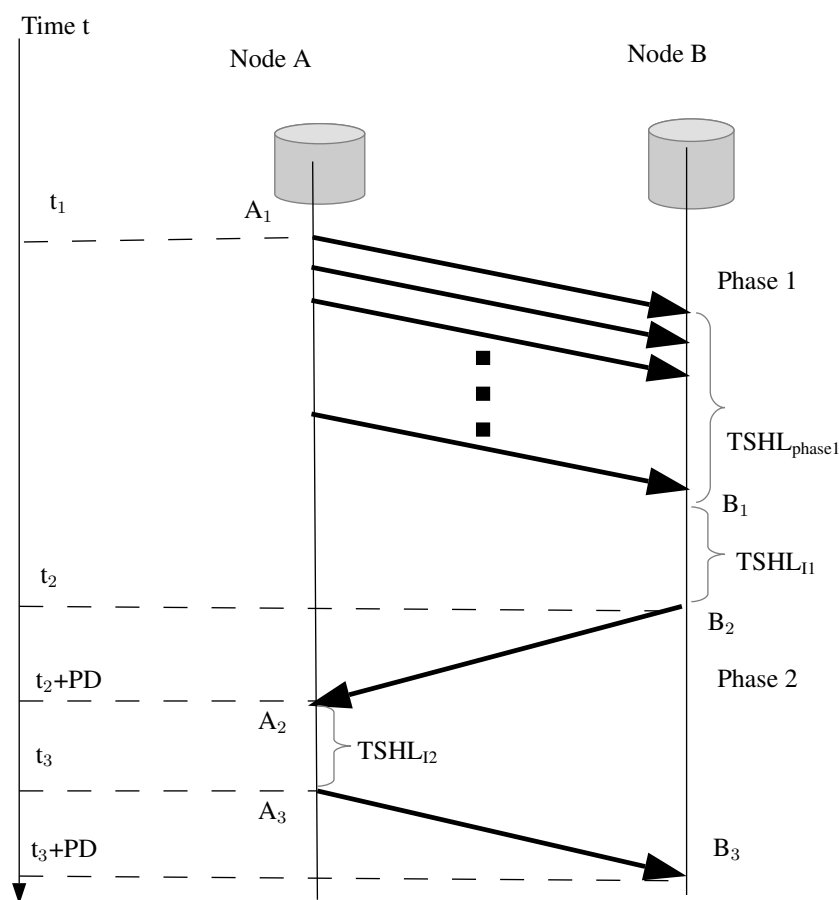


Figure 5.3. TSHL time synchronization protocol [137].

Assuming that the clock of Node B is modeled using skew a_s and offset b_o as follows:

$$B = a_s t + b_o \quad (5.2)$$

The offset b_o can be calculated as:

$$b_o = \frac{[(B_3 + B_2) - a_s(A_2 + A_3)]}{2} \quad (5.3)$$

In the next subsection, detailed analysis of TSHL has been provided.

5.3.1.1 Analysis of TSHL protocol

Phase 2 of TSHL with reference to actual time t has been illustrated in Figure 5.4, in the presence of jitter δ .

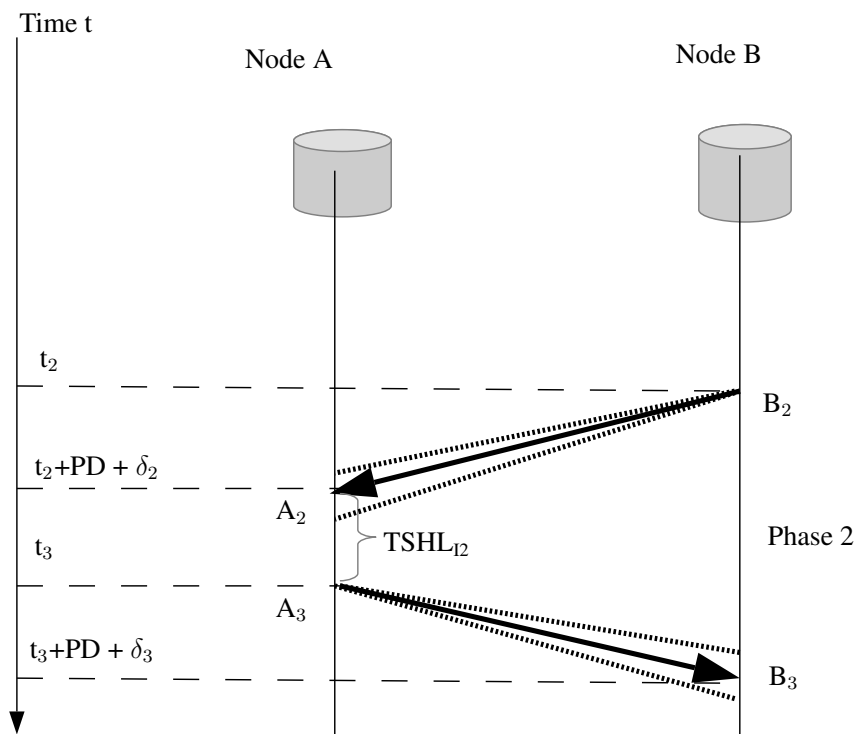


Figure 5.4. Phase 2 of TSHL time synchronization protocol with jitter [137].

Basic equations of time-stamps in TSHL can be written as follows:

$$A'_2 = t_2 + PD + \delta_2 \quad (5.4)$$

$$B_2 = a_s t_2 + b_o \quad (5.5)$$

$$A_3 = t_3 \quad (5.6)$$

$$B'_3 = a_s(t_3 + PD + \delta_3) + b_o \quad (5.7)$$

$$a'_s = a_s(1 - SE) \quad (5.8)$$

In the above equations A'_2 and B'_3 indicates the receive time A_2 and B_3 with the presence of jitter in the communication link. PD indicates the propagation delay, and SE refers to relative skew error. Offset b'_o in the presence of jitter can then be calculated as follows:

$$\begin{aligned} b'_o &= [(B'_3 + B_2) - a'_s(A'_2 + A_3)]/2 \\ &= [b_o + a_s t_2 + a_s(t_3 + PD + \delta_3) + b_o - a_s(1 - SE)(t_2 + \delta_2 + t_3 + PD)]/2 \\ &\approx b_o + [a_s(\delta_3 - \delta_2) + a_s SE(t_2 + t_3 + PD)]/2 \end{aligned} \quad (5.9)$$

After certain interval I from the last synchronization, clock timing t'_k can be obtained using (5.10) as follows:

$$\begin{aligned} t'_k &= (B_k - b'_o)/a'_s \\ &\approx (1 + SE)t_k + (\delta_2 - \delta_3)/2 - SE(t_2 + t_3 + PD)]/2 \end{aligned} \quad (5.10)$$

Time estimation error or offset error ERR_{TSHL} in TSHL protocol can be calculated using (5.11) as follows:

$$\begin{aligned} ERR_{TSHL} &= t'_k - t_k \\ &\approx (\delta_2 - \delta_3)/2 + SE[t_k - ((t_2 + t_3 + PD)/2)] \\ &= (\delta_2 - \delta_3)/2 + SE[t_3 + PD + I - ((t_2 + t_3 + PD)/2)] \\ &= (\delta_2 - \delta_3)/2 + SE[PD + I + TSHL_{12}/2] \end{aligned} \quad (5.11)$$

In the next subsection, brief introduction of another time synchronization protocol termed as ‘Tri-message time synchronization protocol’ proposed Tian et. al. in [137] has been provided.

5.3.2 Tri-message time synchronization protocol

In [137], ‘Tri-message time synchronization protocol’ has been suggested for high latency networks, keeping the resource constrain as main focus. Authors Tian et. al. have suggested the idea of using only three messages for achieving the time synchronization, thus reducing the time and energy spent in the process. This protocol has been designed assuming several factors such as (i) constant propagation delay over the duration of message exchange, (ii) time-stamping at lowest possible layer of protocol stack, and (iii) short-term skew-stable clocks.

For the operation, a two node topology has been considered; one anchor Node A, and other Node B, to be synchronized with Node A. As shown in Figure 5.5, Node A sends the message at time A_1 , and includes the time-stamp A_1 in this message. Node B receives this message at time B_1 . It records time-stamp A_1 and B_1 after propagation delay PD. Later, Node B sends the message back to Node A, at time instant B_2 (and records the time stamp B_2). Time interval or gap of duration Tri_I1 is used after receiving first message to allow the processing time. Node A receives this message at time instant A_2 . Node A then sends the third message at time A_3 , including time-stamps A_2 and A_3 . (Duration Tri_I2 is used by Node A after receiving second message as processing time. In general, the time interval used by any node after receiving i th message is termed as Tri_Ii.) Node B receives the third message at time instant B_3 and thus have all the 6 time-stamps available with Node B. Clock skew (β) and offset (α) can then be calculated easily.

Time stamps of the Node A (that is, A_1, A_2, A_3) can be written in terms of global time reference t as follows:

$$A_1 = t_1 \quad (5.12)$$

$$A_2 = t_2 + PD \quad (5.13)$$

$$A_3 = t_3 \quad (5.14)$$

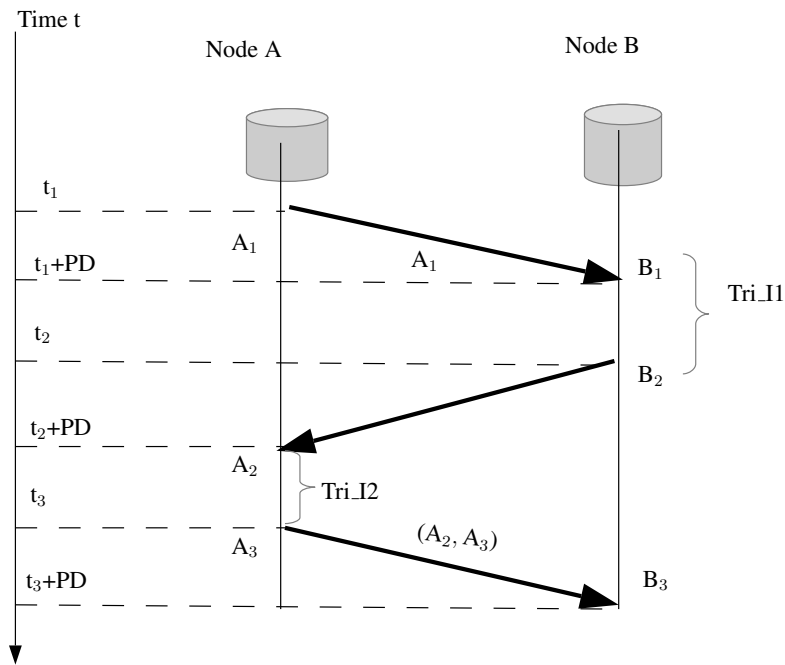


Figure 5.5. Tri-message time synchronization protocol [137].

Time stamps of the Node B (that is, B_1, B_2, B_3) can be written in terms of global time reference t as follows:

$$B_1 = \beta(t_1 + PD) + \alpha \quad (5.15)$$

$$B_2 = \beta(t_2) + \alpha \quad (5.16)$$

$$B_3 = \beta(t_3 + PD) + \alpha \quad (5.17)$$

Timing values t_2 and t_3 can be expressed in terms of t_1 as follows:

$$t_2 \approx t_1 + PD + \text{Tri_I1} \quad (5.18)$$

$$\begin{aligned} t_3 &\approx t_1 + PD + \text{Tri_I1} + PD + \text{Tri_I2} \\ &\approx t_1 + 2PD + \text{Tri_I1} + \text{Tri_I2} \end{aligned} \quad (5.19)$$

From the above equations, clock skew (β) of Node B can be calculated as follows:

$$\beta = \frac{B_3 - B_1}{A_3 - A_1} \quad (5.20)$$

Similarly, offset (α) of Node B can be calculated as follows:

$$\begin{aligned}\alpha &= \frac{\beta(t_1 + \text{PD} + t_2) + 2\alpha}{2} - \frac{(t_1 + \text{PD} + t_2)\beta}{2} \\ \alpha &= \frac{\mathbf{B}_1 + \mathbf{B}_2}{2} - \frac{(\mathbf{A}_1 + \mathbf{A}_2)\beta}{2}\end{aligned}\quad (5.21)$$

In the next subsection, analysis of Tri-message time synchronization protocol has been discussed.

5.3.2.1 Analysis of Tri-message time synchronization protocol

In the presence of receive jitter δ , the time-stamps in Tri-message time synchronization protocol can be written as follows:

$$\begin{aligned}\mathbf{A}'_2 &= \mathbf{A}_2 + \delta_2 \\ &= t_2 + \text{PD} + \delta_2\end{aligned}\quad (5.22)$$

$$\begin{aligned}\mathbf{B}'_1 &= \beta(t_1 + \text{PD} + \delta_1) + \alpha \\ &= \mathbf{B}_1 + \delta_1\beta\end{aligned}\quad (5.23)$$

$$\begin{aligned}\mathbf{B}'_3 &= \beta(t_3 + \text{PD} + \delta_3) + \alpha \\ &= \mathbf{B}_3 + \delta_3\beta\end{aligned}\quad (5.24)$$

Using (5.23), (5.24) and (5.20), clock skew β' can be calculated as given :

$$\begin{aligned}\beta' &= \frac{\mathbf{B}'_3 - \mathbf{B}'_1}{\mathbf{A}_3 - \mathbf{A}_1} \\ &= \beta \left(1 - \frac{\delta_1 - \delta_3}{t_3 - t_1} \right) \\ &= \beta \left(1 - \frac{\delta_1 - \delta_3}{2\text{PD} + \text{Tri.I1} + \text{Tri.I2}} \right) \\ &= \beta(1 - \tau)\end{aligned}\quad (5.25)$$

wherein,

$$\tau = \frac{\delta_1 - \delta_3}{2\text{PD} + \text{Tri.I1} + \text{Tri.I2}}\quad (5.26)$$

Similarly, clock offset can be obtained as follows:

$$\begin{aligned}\alpha' &= \frac{B'_1 + B_2}{2} - \frac{(A_1 + A'_2)\beta'}{2} \\ &\approx \alpha + \frac{(\delta_1 - \delta_2)(\beta)}{2} + \frac{(t_1 + t_2 + PD)(\beta\tau)}{2}\end{aligned}\quad (5.27)$$

Time of the node's clock at instance k after certain interval of INT from last synchronization can be obtained as:

$$\begin{aligned}t'_k &= \frac{B_k - \alpha'}{\beta'} \\ &\approx \left(\frac{\beta t_k + \alpha - \alpha'}{\beta} \right) (1 + \tau)\end{aligned}\quad (5.28)$$

Estimation error (ERR_{TRI}) in time of node's clock is given as follows:

$$\begin{aligned}ERR_{TRI} &= t'_k - t_k \\ &= t_k\tau + \left(\frac{\alpha - \alpha'}{\beta} \right) (1 + \tau) \\ &= (t_3 + PD + INT)\tau - \frac{(\delta_1 - \delta_2)}{2} - \frac{(t_1 + t_2 + PD)\tau}{2} \\ &= \left(\frac{INT}{2PD + Tri_I1 + Tri_I2} + \frac{4PD + Tri_I1 + 2Tri_I2}{4PD + 2Tri_I1 + 2Tri_I2} \right) (\delta_1 - \delta_3) \\ &\quad + \frac{\delta_2 - \delta_1}{2}\end{aligned}\quad (5.29)$$

In the next subsection, comparison of TSHL and Tri-message time synchronization protocols has been provided.

5.3.3 Comparison of TSHL and Tri-message time synchronization protocols

Equations (5.11) and (5.29) representing time estimation error in TSHL and Tri-message has been used to evaluate the performance of these protocols using MATLAB simulation platform. Three metrics have been considered for analysis of the protocols namely a) effect of delay on the estimation, b) effect of receive jitter distribution, and c) effect of time interval from last synchronization.

Delay variations : In this simulation set-up, delay has been varied between two nodes from 1 s to 20 s. Though the delay here refers to the propagation delay between transmitter and receiver, it is important to consider the message delivery time for more practical scenario. In UASN, since data rate is very low, total delivery time of message is very large. For example, two nodes which are 1.5 km apart and communicating at data rate of 100 bps has message delivery delay as 9 s for message of 100 bytes in length. Jitter in the network is centered around this delay. Moreover, in analytical modeling, the difference between receive time-stamp and transmit time-stamp is taken as PD with the assumption that the time-stamps are done at lowest possible layer in network protocol stack. If the time-stamps are done at higher layers, then the difference between receive time-stamp and transmit time-stamp is the message delivery time.

Time estimation error of TSHL and Tri-message has been evaluated for the varying delay. Other parameters for simulation have been fixed as follows: i) receive jitter = 10 μ s, ii) time interval from the last synchronization INT (Tri-message) = I (TSHL) = 120 s, and iii) message interval Tri_I1 = Tri_I2 = TSHL₁₂ = 10 s. Results have been presented in graphical form in Figure 5.6. It was observed that the error in Tri-message decreases as the delay increases. This is obvious since the error in Tri-message is inversely proportional to delay. In TSHL, error in time estimation or (time offset value) increases as the delay increases, but the increase is very gradual, indicating that error is not heavily dependent on delay.

Jitter distribution : It is assumed that many sources of errors and various uncertainties in time synchronization can be treated as a receive time jitter, which follows Gaussian distribution [137]. For this simulation, the difference of receive jitter $\delta_2 - \delta_3$ of (5.11) is modeled as a single Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$. Similarly, for the analysis of ERR_{TRI} in (5.29), terms $\delta_2 - \delta_1$ and $\delta_1 - \delta_3$ have been modeled using Gaussian distribution. Results have been plotted in Figure 5.7 with reference to jitter variations from 5 μ s to 50 μ s. Other parameters for simulation have been fixed as follows: i) delay= 10 s, ii) time interval from the last synchronization INT (Tri-message) = I (TSHL) = 120 s, and iii) message interval Tri_I1 = Tri_I2 = TSHL₁₂ = 10 s. Results show that the Tri-message time synchronization protocol has higher sensitivity to jitter as compared to TSHL protocol.

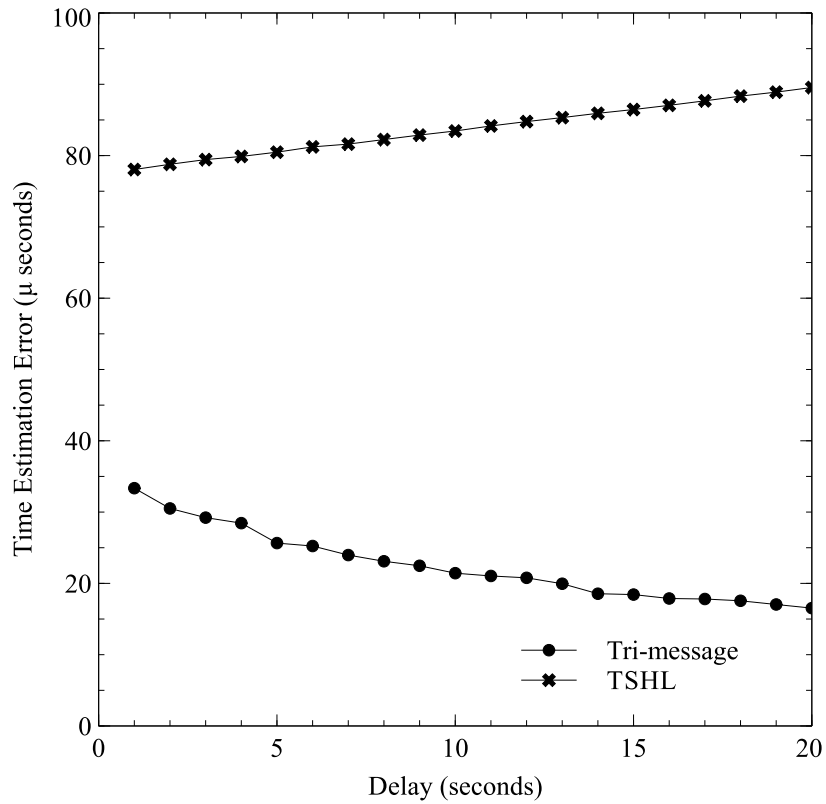


Figure 5.6. Effect of delay on time estimation error in time synchronization protocols.

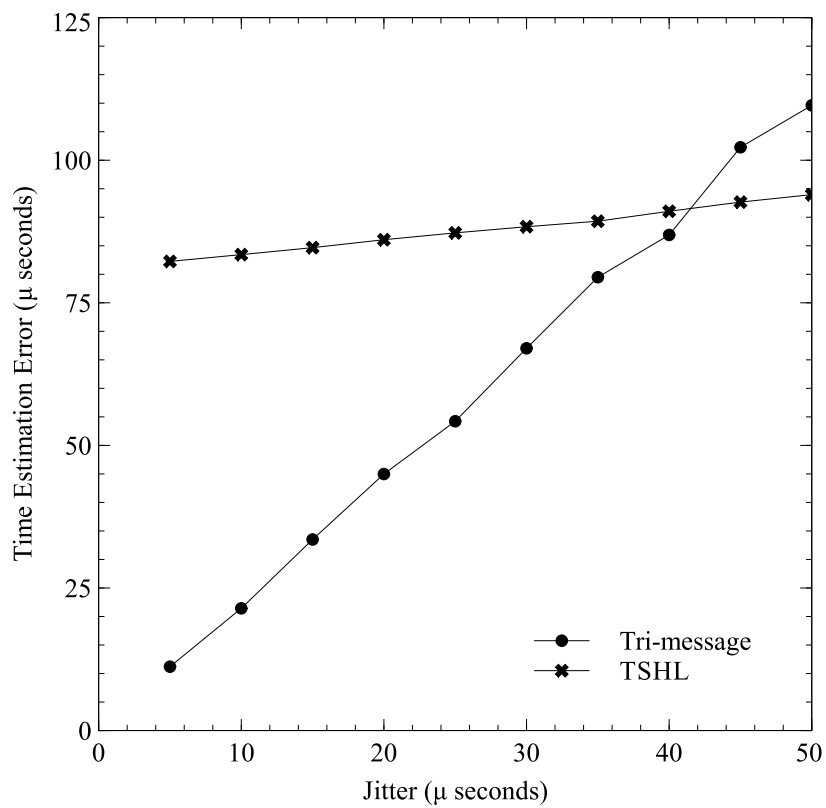


Figure 5.7. Effect of jitter on time estimation error in time synchronization protocols.

Time interval from last synchronization : In this simulation set-up, the performance of time synchronization has been evaluated with reference to time passed from the last synchronization cycle. This metric illustrates the requirement of repetition cycle of synchronization depending upon the deviation in time of node's clock. Here, the interval I in (5.11) or INT in (5.29) have been varied from 60 s to 600 s. Other parameters for simulation have been fixed as follows: i) delay = 10 s, ii) jitter = 10 μ s, and iii) message interval $Tri_{I1} = Tri_{I2} = TSHL_{I2} = 10$ s. Results have been plotted in Figure 5.8. It can be observed that TSHL shows large deviation in node's time estimate after a time gap from last synchronization.

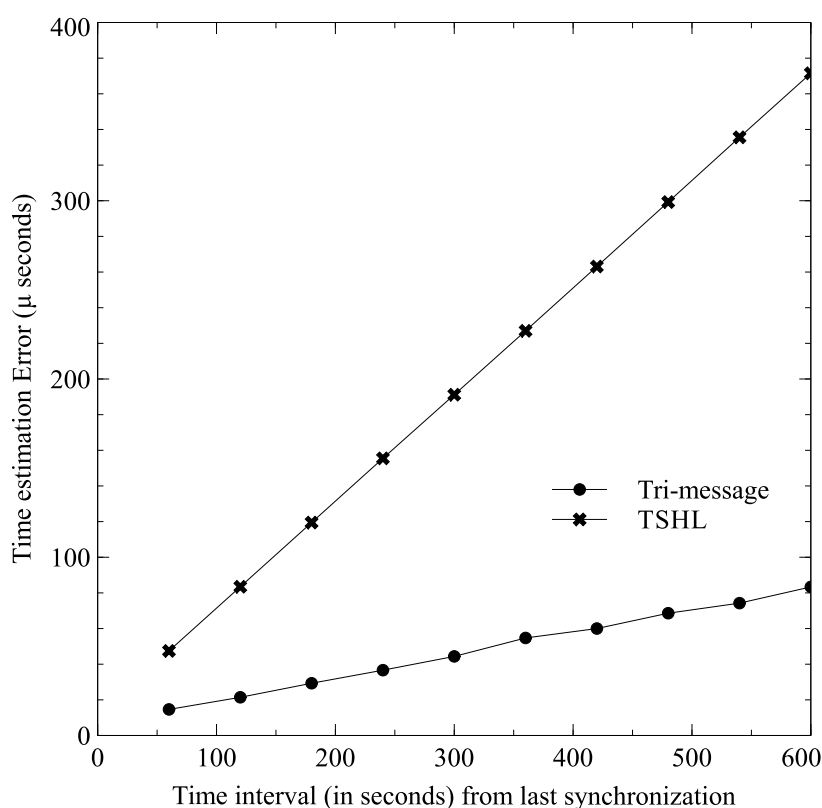


Figure 5.8. Effect of time interval after last synchronization on time estimation error.

5.3.3.1 Resource utilization of TSHL and Tri-message time synchronization protocols

As described in Subsection 5.3.1, TSHL uses linear regression technique for skew compensation in first phase of time synchronization. Accuracy of TSHL is based on number of beacon messages used in phase 1. In [136], authors have used around 25 messages in phase 1 for achieving desired accuracy. Using Least Mean Square (LMS) algorithm, number of multiplications required for

linear regression in TSHL is $6n+18$, wherein n is the number of beacon messages. For 25 messages, skew compensation needs 168 multiplications. Another 2 messages are utilized for phase 2. On the other hand, in Tri-message protocol, number of message exchanges required are only 3. Number of multiplications needed in this protocol is only 4. It indicates that the computational complexity of TSHL is higher as compared to Tri-message protocol.

Energy requirement (in joules) of these protocols can be determined as follows:

$$\text{Energy} = \text{No. of packets} \times \hat{T} \times P_c \quad (5.30)$$

wherein, \hat{T} is the packet transmission time, given by (4.1). P_c is the power consumption factor. Value of P_c has been taken as 0.001 W for energy calculations. Resource consumption of TSHL and Tri-message protocol has been tabulated in Table 5.1.

Protocol	No. of messages	No. of multiplications	Energy consumption (J)
TSHL	27	168	0.027
Tri-message	3	4	0.003

Table 5.1. Resource utilization for TSHL and Tri-message time synchronization protocols.

It can be observed from results tabulated in the Table 5.1 that, Tri-message is a lightweight time synchronization protocol consuming only 11% of energy as compared to TSHL. Also, it can be stated that the computational cost in terms of complexity, time and memory requirement is very less in Tri-message in comparison to TSHL.

5.4 Extension of Tri-message time synchronization protocol for multi-hop topology

In this work, an extension of Tri-message protocol has been proposed for achieving time synchronization in multi-hop scenario. Tri-message time synchronization protocol has been chosen since this protocol is energy efficient and computationally simpler. The amount of time required for achieving synchronization is also very small as compared to other protocols based on linear regression method.

The extension of Tri-message protocol can be done in modular fashion as follows: assume that one anchor node (Node A) is available in the network which has the correct reference timing information. This anchor node can be used for synchronizing another node (Node B) in its single hop region. This Node B can then synchronize another node (Node C) in its own single hop distance as shown in Figure 5.9. In this manner, synchronization can be implemented over multi-hop topology. Global time scaling can be readily available since the equations can be resolved in linear fashion.

Considering that $A(t)$, $B(t)$ and $C(t)$ are clocks of nodes A, B, and C respectively, $B(t)$ can be modeled in terms of clock of Node A as follows:

$$\beta_1 = \frac{B_3 - B_1}{A_3 - A_1} \quad (5.31)$$

$$\alpha_1 = \frac{B_1 + B_2}{2} - \frac{(A_1 + A_2)\beta_1}{2} \quad (5.32)$$

$$B(t) = \beta_1 A(t) + \alpha_1 \quad (5.33)$$

Similarly, $C(t)$ can be modeled in terms of clock of Node B as follows:

$$\beta_2 = \frac{C_3 - C_1}{B_6 - B_4} \quad (5.34)$$

$$\alpha_2 = \frac{C_1 + C_2}{2} - \frac{(B_4 + B_5)\beta_2}{2} \quad (5.35)$$

$$C(t) = \beta_2 B(t) + \alpha_2 \quad (5.36)$$

From (5.33) and (5.36), $C(t)$ can be modeled in terms of clock of Node A as follows:

$$C(t) = \beta_1 \beta_2 A(t) + \beta_2 \alpha_1 + \alpha_2 \quad (5.37)$$

Equations (5.36) and (5.37) represent the node's clock in terms of clock skew and offset under ideal conditions, without sources of error in the network. The time estimation error in multi-hop topology can be calculated using similar analogy as described in Subsection 5.3.2.1, wherein receive jitter is used to model the uncertainties in timing sources. The analysis of this extension has been provided in the next subsection.

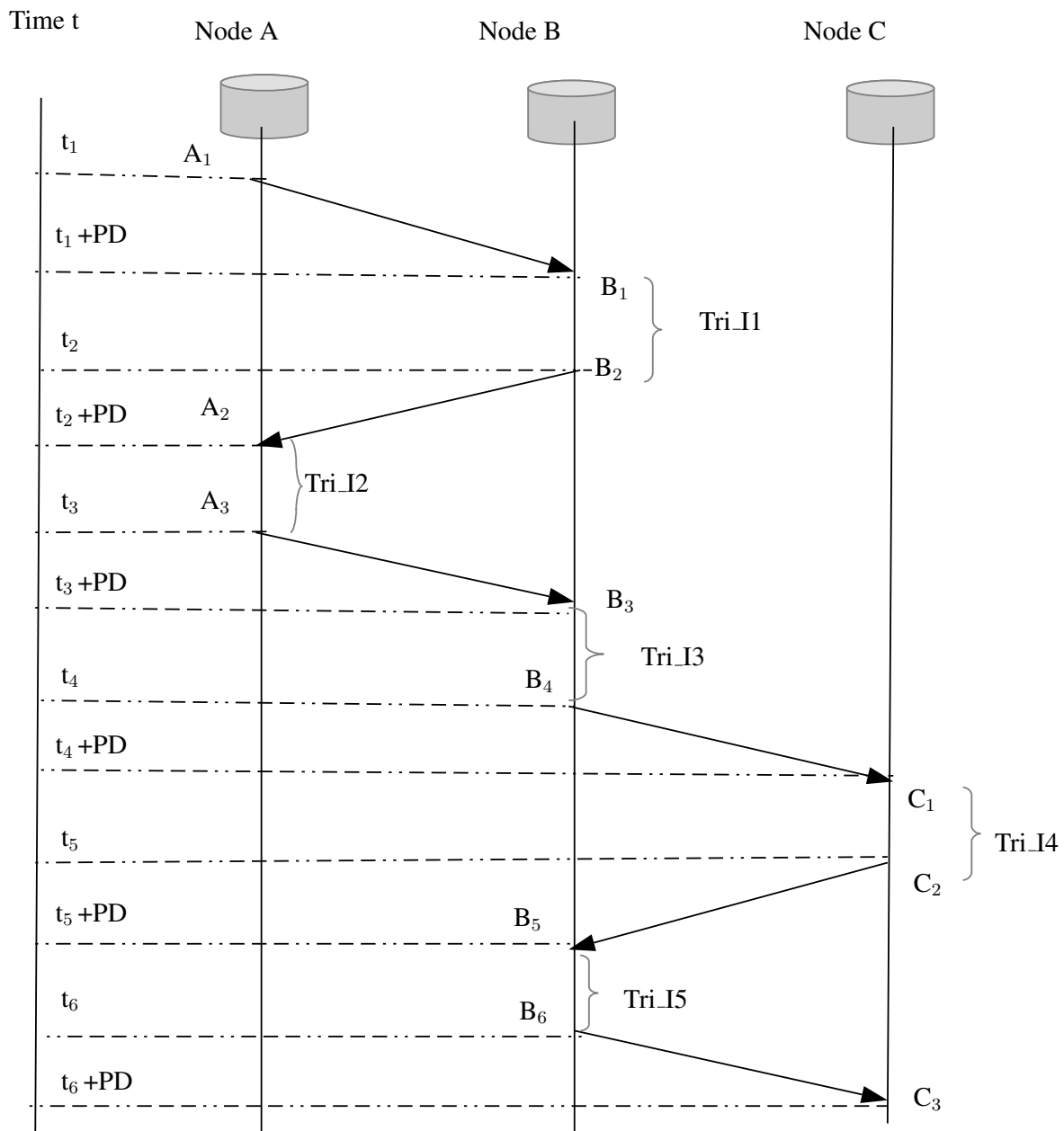


Figure 5.9. Extension of Tri-message time synchronization protocol for multi-hop scenario.

5.4.1 Analysis of multi-hop Tri-message time synchronization protocol

For the analysis of multi-hop Tri-message time synchronization protocol, three node topology as shown in Figure 5.9 has been considered. Node A has been taken as an anchor node which has correct reference timing information available. Process of time synchronization starts with Node A. Initially, this anchor node (Node A) synchronizes one node (Node B) on its single hop link. After having time synchronized, Node B acts as a reference node for another node (Node C) on its single hop link. The interval Tri_I3 is the time interval between completing synchronization

of Node B and starting synchronization of Node C. This time interval Tri_{I3} should be very less. Larger time gap would lead to larger deviation in timing of Node B, and hence percolation of error in next hops. Within the interval Tri_{I3} , timing of Node B would have developed an error given by ERR_{TRI} in (5.29). In presence of this error, timing values can be represented as follows:

$$t'_4 = t_4 + \text{ERR}_{\text{TRI}} \quad (5.38)$$

$$t'_5 = t_5 + \text{ERR}_{\text{TRI}} \quad (5.39)$$

$$t'_6 = t_6 + \text{ERR}_{\text{TRI}} \quad (5.40)$$

$$t'_5 = t'_4 + \text{PD} + \text{Tri}_{I4} \quad (5.41)$$

$$\begin{aligned} t'_6 &= t'_5 + \text{PD} + \text{Tri}_{I5} \\ &= t'_4 + 2\text{PD} + \text{Tri}_{I4} + \text{Tri}_{I5} \end{aligned} \quad (5.42)$$

The time-stamp values of second hop can be represented as follows (with and without the effect of jitter in the network):

$$B_4 = t'_4 \quad (5.43)$$

$$C_1 = \beta_2(t'_4 + \text{PD}) + \alpha_2 \quad (\text{Without jitter}) \quad (5.44)$$

$$\begin{aligned} C'_1 &= \beta_2(t'_4 + \text{PD} + \delta_4) + \alpha_2 \quad (\text{With jitter}) \\ &= C_1 + \delta_4\beta_2 \end{aligned} \quad (5.45)$$

$$C_2 = \beta_2(t'_5) + \alpha \quad (5.46)$$

$$B_5 = t'_5 + \text{PD} \quad (\text{Without jitter}) \quad (5.47)$$

$$\begin{aligned} B'_5 &= t'_5 + \text{PD} + \delta_5 \quad (\text{With jitter}) \\ &= B_5 + \delta_5 \end{aligned} \quad (5.48)$$

$$B_6 = t'_6 \quad (5.49)$$

$$C_3 = \beta_2(t'_6 + \text{PD}) + \alpha \quad (\text{Without jitter}) \quad (5.50)$$

$$\begin{aligned} C'_3 &= \beta_2(t'_6 + \text{PD} + \delta_6) + \alpha \quad (\text{With jitter}) \\ &= C_3 + \delta_6\beta_2 \end{aligned} \quad (5.51)$$

From the above equations, clock skew β'_2 of Node C can be calculated as given below:

$$\begin{aligned}
\beta'_2 &= \frac{C'_3 - C'_1}{B_6 - B_4} \\
&= \beta_2 \left(1 - \frac{\delta_4 - \delta_6}{t'_6 - t'_4} \right) \\
&= \beta_2 \left(1 - \frac{\delta_4 - \delta_6}{2PD + \text{Tri_I4} + \text{Tri_I5}} \right) \\
&= \beta_2(1 - \tau_2)
\end{aligned} \tag{5.52}$$

wherein,

$$\tau_2 = \frac{\delta_4 - \delta_6}{2PD + \text{Tri_I4} + \text{Tri_I5}} \tag{5.53}$$

Similarly, clock offset can be obtained as follows:

$$\begin{aligned}
\alpha'_2 &= \frac{C'_1 + C_2}{2} - \frac{(B_4 + B'_5)\beta'_2}{2} \\
&\approx \alpha_2 + \frac{(\delta_4 - \delta_5)(\beta_2)}{2} + \frac{(t'_4 + t'_5 + PD)(\beta_2\tau_2)}{2}
\end{aligned} \tag{5.54}$$

Time of Node C at instance m after interval INT from last synchronization is obtained as follows:

$$\begin{aligned}
t''_m &= \frac{C_m - \alpha'_2}{\beta'_2} \\
&\approx \left(\frac{\beta_2 t'_m + \alpha_2 - \alpha'_2}{\beta_2} \right) (1 + \tau_2)
\end{aligned} \tag{5.55}$$

wherein,

$$t'_m = t_m + \text{ERR}_{\text{TRI}} \tag{5.56}$$

Time estimation error in the second-hop represented as ERR_{TRI2} is given as follows:

$$\begin{aligned}
\text{ERR}_{\text{TRI2}} &= t''_m - t'_m \\
&= t'_m \tau_2 + \left(\frac{\alpha_2 - \alpha'_2}{\beta_2} \right) (1 + \tau_2) \\
&= (t'_6 + PD + \text{INT})\tau_2 - \frac{(\delta_4 - \delta_5)}{2} - \frac{(t'_4 + t'_5 + PD)\tau_2}{2} \\
&= \left(\frac{\text{INT}}{2PD + \text{Tri_I4} + \text{Tri_I5}} + \frac{4PD + \text{Tri_I4} + 2\text{Tri_I5}}{4PD + 2\text{Tri_I4} + 2\text{Tri_I5}} \right) (\delta_4 - \delta_6) \\
&\quad + \frac{\delta_5 - \delta_4}{2}
\end{aligned} \tag{5.57}$$

It can be observed that the estimation error in second-hop ERR_{TRI2} in (5.57) is same as the estimation error ERR_{TRI} obtained using (5.29), assuming that the other parameters such as jitter, delay and message intervals are same. For clarity, estimation error is represented as ERR_{TRIX} , where x represents the hop value. For example, ERR_{TRI1} is estimation error in first hop, ERR_{TRI2} is estimation error in second hop and so on. Using these notations, time estimation error of Tri-message over multi-hop topology is denoted as $ERR_{TRI(n-hops)}$. For example, error over two-hops is represented as $ERR_{TRI(2-hops)}$, error over 3-hops is represented as $ERR_{TRI(3-hops)}$ and so on.

In multi-hop time synchronization, the process of synchronization of second hop starts after an interval of Tri_I3 from completion of first hop synchronization. Hence, the error of estimation over two hops is summation of ERR_{TRI1} (measured after interval Tri_I3) and ERR_{TRI2} . Error over two-hop topology can be represented as follows:

$$ERR_{TRI(2-hops)} = ERR_{TRI1}(\text{ after interval } TRI_I3) + ERR_{TRI2} \quad (5.58)$$

On a similar lines, error over three-hops is given by:

$$ERR_{TRI(3-hops)} = ERR_{TRI1}(\text{ after interval } TRI_I3) + ERR_{TRI2}(\text{ after interval } TRI_I6) + ERR_{TRI3} \quad (5.59)$$

The performance of multi-hop Tri-message time synchronization has been tested using MATLAB simulations. Results of simulation have been provided in next subsection.

5.4.2 Simulation results of multi-hop Tri-message time synchronization

The proposed multi-hop Tri-message time synchronization protocol has been simulated for two-hop topology using MATLAB. Various parameters considered for simulation of Tri-message protocol are as follows: i) delay= 10 s, ii) jitter = 10 μ s, and iii) message interval $Tri_I1 = Tri_I2 = 10$ s. Value of Tri_I3 has been varied from 10 s to 100 s. Value of Tri_I3 is a time gap between completion of time synchronization for first-hop of the network and starting of time synchronization for second-hop. Results have been illustrated in Figure 5.10. For every interval value of Tri_I3 , the time estimation error of second-hop has been calculated after time gap of 0 s,

60 s, and 120 s respectively. Error obtained just after finishing time synchronization (that is after a gap of 0 s) is termed as instant error. Results indicate that error increases as the interval Tri_I3 increases. Also, as the time passes from finishing of time synchronization, the error in node's timing keeps on increasing.

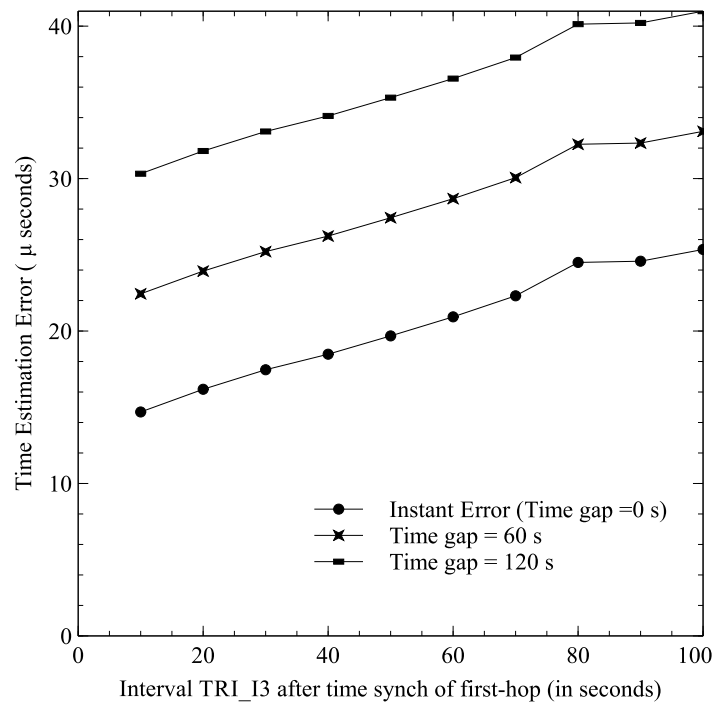


Figure 5.10. Time estimation error for two-hop topology.

Figure 5.11 illustrates the working of Tri-message protocol over multi-hop topology. Here, message intervals have been taken as 10 s for all messages. Figure shows the instant error for every hop in a five-hop network architecture. It indicates that the instant error of time estimation gets accumulated over the number of hops and increases almost linearly with increase in hops.

5.5 Implementation of multi-hop Tri-message time synchronization protocol on hardware testbed set-up

The proposed multi-hop Tri-message time synchronization protocol has been implemented on hardware testbed set-up of UASN shown in Figure 4.2 [221, 222]. Network architecture for this deployment has been represented in Figure 5.12.

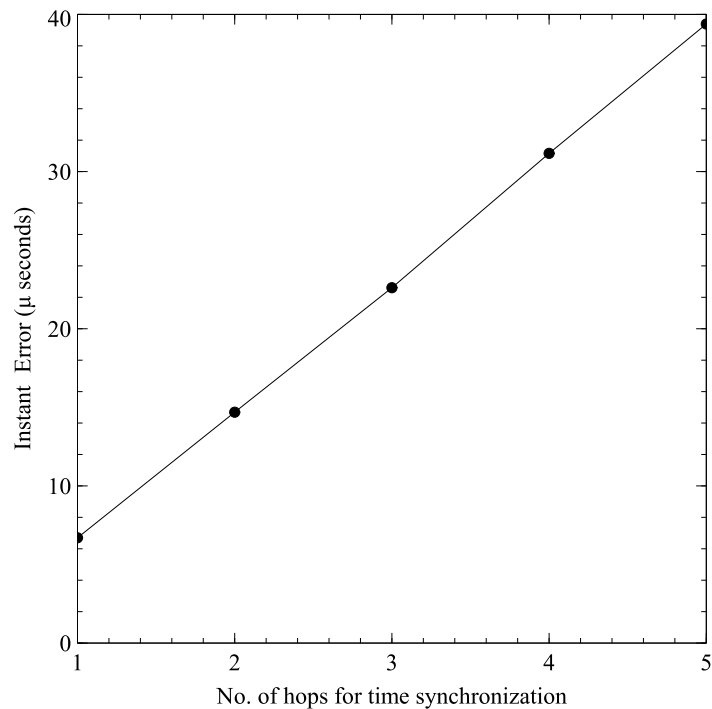


Figure 5.11. Estimation of instant error for multi-hop topology.

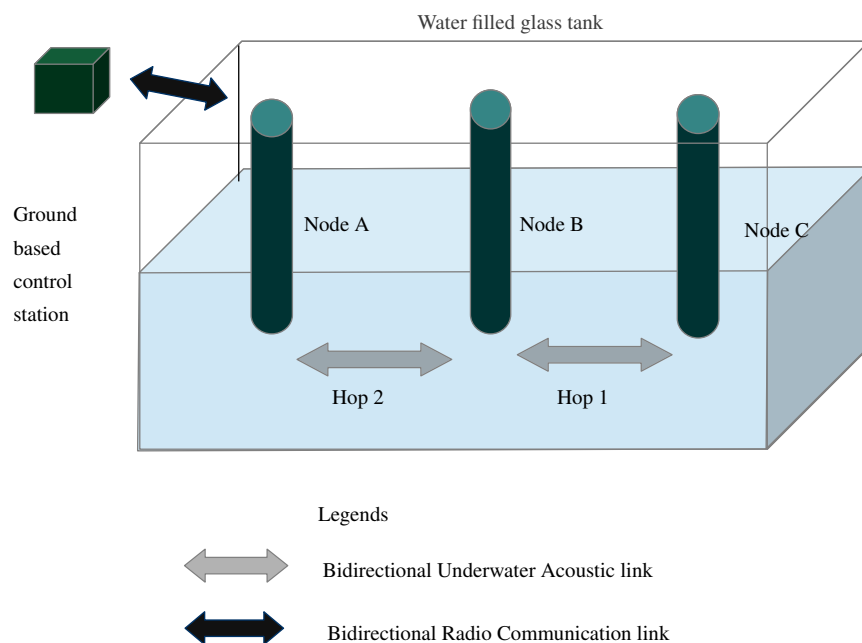


Figure 5.12. Network architecture for testbed implementation of multi-hop Tri-message time synchronization protocol.

In the implementation of multi-hop Tri-message time synchronization protocol on a testbed set-up, three acoustic nodes have been utilized. These nodes have been termed as Node A, Node B and Node C. Each underwater acoustic node has been developed using SAM, TelosB mote, interfacing units and power supplies as shown in Figure 4.1. Node A receives the timing information from the

ground based control station, and it acts as an anchor node in the multi-hop time synchronization. In the given protocol, Node A first synchronizes Node B. Three message exchanges are done between Node A and Node B over a single hop link to achieve time synchronization. Node B calculates the offset and clock skew values with respect to Node A. These values are used by Node B to correct it's timing with reference to Node A. Node B then synchronizes Node C situated one hop away. Using three message exchanges, Node C calculates offset and clock skew with respect to Node B for correction of it's clock. Node C also calculates the offset and clock skew with respect to Node A using (5.37). In this way, a global scale of timing information is available in the entire network.

In a testbed set-up, nodes are in close proximity to each other. With no provision to set up dynamic power levels and no provision of sleep mode with SAM, all the nodes reside in a single collision domain. This indicate that the nodes can overhear the communication. Though, this is possible, the overhearing mechanism has not been used in implementation of this protocol. It has been assumed that Node A can communicate with (only) Node B in a single hop communication, and can not reach to Node C. Further, Node B can reach to both Node A and Node C in a single hop link. Node C can communicate with Node B only using a single hop link.

In the implementation of multi-hop Tri-message time synchronization protocol, networking parameters have been set as follows:

1. Packet size (P_S) - Packet size (or a Frame size at MAC layer) is set as 22 bytes.
2. Bit Rate (R) - Data rate of the SAM is set as 8 bps.
3. Message interval - Interval times between messages has been set as $\text{Tri_I1} = \text{Tri_I2} = \text{Tri_I4} = \text{Tri_I5} = 10$ s. Value Tri_I3 has been varied between 10 s to 200 s.

For this scenario, using (4.1), (4.2) and (4.3), it can be observed that,

$$\hat{T} = 22 \text{ s.}$$

$$\hat{D} = 22 \text{ s. (Assuming negligible } \hat{P})$$

Time T_1 required for time synchronization on one hop is thus given by,

$$T_1 = 3 \times \hat{D} + \text{Tri_I1} + \text{Tri_I2} \quad (5.60)$$

Time T_2 required for time synchronization over two hop is given by,

$$T_2 = 6 \times \hat{D} + \text{Tri_I1} + \text{Tri_I2} + \text{Tri_I3} + \text{Tri_I4} + \text{Tri_I5} \quad (5.61)$$

From (5.60) and (5.61), it can be found out that $T_1 = 86$ s. and, $T_2 = 182$ s (with $\text{Tri_I3} = 10$ s). This are the optimum values possible for time synchronization if all packet transmissions are successful. In this testbed set-up, PDR of 75% has been observed as described in Subsection 4.4.1. Time synchronization is critical control information for the sensor networks. For this reason, a mechanism of retransmission of messages has been provided in case of packet drops or packet errors. In Tri-message time synchronization, there is implicit acknowledgment in the message structure. For example, when Node A transmits (first) message to Node B, it expects the (second) message from Node B. If it does not receive this second message or receives an erroneous message, then Node A can retransmit the first message again. Three attempts of retransmission has been provided on the network.

To obtain the actual time-stamp values of the packet transmission and reception, the network has been probed at MAC layer. In order to obtain values of offset and clock skew via probing, one more type of message (fourth message) has been added in the protocol. This (fourth) message is transmitted by the node after it is time synchronized. In this message, node transmits the calculated values of offset and clock skew to the synchronizing node. For example, consider that Node A is synchronizing Node B. After three message exchanges, Node B calculates the time synchronization parameters, offset and skew. Node B then transmits these calculated values to Node A. This feature has helped us to probe the actual values of the test-runs.

An example of two-hop Tri-message time synchronization has been discussed in detail here. The following data represent the four messages obtained in the testbed set-up during a time synchronization of Node A and Node B. A data packet in the TinyOS 2.x serial stack uses HDLC framing.

Message 1 - 7E 45 00 FF FF 00 00 09 00 20 01 00 00 75 3A 00 00 75 3A 98 77 7E

Message 2 - 7E 45 00 FF FF 00 00 09 00 21 02 00 00 F2 49 00 00 CB 39 CD D7 7E

Message 3 - 7E 45 00 FF FF 00 00 09 00 22 01 00 01 6F 76 00 01 48 66 36 CD 7E

Message 4 - 7E 45 00 FF FF 00 00 09 00 23 02 05 F6 FF A0 FF FF 21 ED 37 E2 7E

Various fields of above messages represent following information:

byte[1] = 0x7E - Framing byte of HDLC (Start of frame)

byte[2] = 0x45 - Protocol type

byte[3] = 0x00 - Sequence number (0x00 indicates it is not used.)

byte[4,5] = 0xFFFF - Packet format dispatch type - (0xFFFF indicated broadcast type)

byte[6,7] = 0x0000 - Link source address (The serial stack does not set the link source address)

byte[8] = 0x09 - Message length (Size of payload)

byte[9] = 0x00 - Group ID

byte[10] - Message ID

byte[11-19] - Payload of the frame

byte[20-21] - CRC bytes

byte[22] = 0x7E - Framing byte of HDLC (End of frame)

In the payload of the frame, one byte has been used to indicate packet ID, one byte for node ID, 4 bytes for time-stamp 1 and 4 bytes for time-stamp 2. Node A and Node B has node ID numbers as 01 and 02 respectively.

Four messages given above provides valuable information, and can be deciphered as follows:

1. Message 1 indicates that the packet with ID number 20 has been transmitted by Node 01 (Node A) and has time stamp value $A_1 = 0x0000753A = 30.010$ s. In this packet, time stamp A_1 is inserted twice.
2. Message 2 indicates that the packet with ID number 21 has been transmitted by Node 02 (Node B). This packet contains two time-stamps B_2 followed by B_1 . From the message, it can be observed that, the value of B_2 as $0x0000F249 = 62.025$ s and B_1 as $0x0000CB39 = 52.025$ s.
3. Message 3 indicates that the packet with ID number 22 has been transmitted by Node 01 (Node A). This packet contains two time stamps A_3 followed by A_2 . Here, $A_2 = 0x00014866 = 84.070$ s. and $A_3 = 0x00016F76 = 94.070$ s.
4. Message 4 has the packet ID 23 and is transmitted by Node B. In this message instead of time-stamps, the payload bytes are used for transmitting Clock skew and offset. Clock skew in the message is $0x05F6FFA0 = 1.00073376$ and offset is $0xFFFF21ED = -0.056853$ s.

The value of B_3 has been obtained using interceptor node at Node B. Value of time-stamp B_3 in this test-run is $0x1C5A4 = 116.132$ s. (Note that, in the message structure, various scaling factors have been used to achieve desired accuracy in readings. These scaling factors have been set as 10^3 for the time-stamps, 10^8 for Clock skew and 10^6 for offset.)

By using (5.20) and (5.21), it can be noted that, $\beta_1 = 1.00073376$ and $\alpha_1 = -0.56853$, which is same as obtained by probing the messages on the network.

Another observation using this information is to calculate the variations in delivery time of the packet. On the testbed set-up, delay and jitter observed in delivery of these four messages can be obtained as follows:

For message 1, delay = $B_1 - A_1 = 22.015$ s., Jitter $\delta_1 = 0.015$ s.

For message 2, delay = $A_2 - B_2 = 22.045$ s., Jitter $\delta_2 = 0.045$ s.

For message 3, delay = $B_3 - A_3 = 22.062$ s., Jitter $\delta_3 = 0.062$ s.

Reason for this differences is the sources of errors described in Subsection 5.2.2. The main reason for the limitations in obtaining precision of the time synchronization in UASN is the variable propagation delay of underwater acoustic channel. Using (5.29), the instant error in time estimation in single-hop can be obtained as -0.028328 s.

After completing synchronization on first-hop, the interval of $Tri_{I3} = 80s$ has been used in this test-run. Messages obtained in the testbed set-up during a time synchronization of Node B and Node C in second-hop are as follows:

Message 4 - 7E 45 00 FF FF 00 00 09 00 29 02 00 02 FE 27 00 02 FE 27 0D 47 7E

Message 5 - 7E 45 00 FF FF 00 00 09 00 2A 03 00 03 7B 3E 00 03 54 2E 2B CA 7E

Message 6 - 7E 45 00 FF FF 00 00 09 00 2B 02 00 03 F8 7C 00 03 D1 6C 8D 8D 7E

Message 7 - 7E 45 00 FF FF 00 00 09 00 2C 03 05 F6 54 D0 FF FE B1 5D A9 E8 7E

From the above messages, various parameters can be obtained as given below :

Time-stamp $B_4 = 196.135$ s.

Time-stamp $C_1 = 218.158$ s.

Time-stamp $B_5 = 250.220$ s.

Time-stamp $C_2 = 228.158$ s.

Time-stamp $B_6 = 260.220$ s.

Time-stamp $C_3 = 282.262$ s.

Using (5.34) and (5.35), clock skew and offset values can be obtained as $\beta_2 = 1.00029648$ and $\alpha_2 = -0.085667$ s respectively.

Delay and jitter observed in delivery of these four messages can be obtained as follows:

For message 4, delay = $C_1 - B_4 = 22.023$ s., Jitter $\delta_4 = 0.023$ s.

For message 5, delay = $B_5 - C_2 = 22.062$ s., Jitter $\delta_5 = 0.062$ s.

For message 6, delay = $C_3 - B_6 = 22.042$ s., Jitter $\delta_6 = 0.042$ s.

Using (5.57), the time estimation error on second-hop can be observed as 0.001984 s. Whereas using (5.58), the accumulated error over two-hop topology is -0.085093 s.

Results of this test-run along with two additional test-runs have been provided in tabular form in Table 5.2 and Table 5.3. In Table 5.2, the values of time-stamps in hop1 and hop2 have been provided. Instant error for first hop, second hop along with total error over two-hop topology has been given in this table.

No.	Hop1		Hop2		Total
	Time-stamps (s)	ERR _{TRI1} (s)	Time-stamps (s)	ERR _{TRI2} (s)	ERR _{TRI(2-hops)} (s)
1	$A_1 = 30.010$	ERR _{TRI1} = -0.028328	$B_4 = 196.135$	ERR _{TRI2} = 0.001984	For Tri_I3 = 80 s ERR _{TRI1} = -0.087078 ERR _{TRI(2-hops)} = -0.085093
	$B_1 = 52.025$		$C_1 = 218.158$		
	$A_2 = 84.070$		$B_5 = 250.220$		
	$B_2 = 62.025$		$C_2 = 228.158$		
	$A_3 = 94.070$		$B_6 = 260.220$		
	$B_3 = 116.132$		$C_3 = 282.262$		
2	$A_1 = 37.138$	ERR _{TRI1} = -0.006125	$B_4 = 283.382$	ERR _{TRI2} = -0.014797	For Tri_I3 = 160 s ERR _{TRI1} = 0.013875 ERR _{TRI(2-hops)} = -0.000922
	$B_1 = 59.198$		$C_1 = 305.414$		
	$A_2 = 91.231$		$B_5 = 337.433$		
	$B_2 = 69.198$		$C_2 = 315.414$		
	$A_3 = 101.231$		$B_6 = 347.433$		
	$B_3 = 123.283$		$C_3 = 369.474$		
3	$A_1 = 40.053$	ERR _{TRI1} = -0.051500	$B_4 = 186.162$	ERR _{TRI2} = 0.040530	For Tri_I3 = 60 s ERR _{TRI1} = -0.111500 ERR _{TRI(2-hops)} = -0.070970
	$B_1 = 62.063$		$C_1 = 208.244$		
	$A_2 = 94.088$		$B_5 = 240.278$		
	$B_2 = 72.063$		$C_2 = 218.244$		
	$A_3 = 104.088$		$B_6 = 250.278$		
	$B_3 = 126.162$		$C_3 = 272.290$		

Table 5.2. Results of implementation of multi-hop Tri-message time synchronization of UASN testbed.

Results of clock skew and offset calculations are tabulated in Table 5.3. Column 2 and 3 shows the clock skew and offset values of Node B with respect to reference clock of Node A. Column 4, 5 shows the values clock of Node C with respect to Node B. Columns 6 and 7 shows the time synchronization parameters of Node C with respect to Node A.

No.	Node B w.r.t. Node A		Node C w.r.t. Node B		Node C w.r.t. Node A	
	Clock Skew β_1	Offset α_1 (s)	Clock Skew β_2	Offset α_2 (s)	Clock Skew $\beta_1\beta_2$	Offset $\beta_2\alpha_1 + \alpha_2$ (s)
1	1.000733	-0.056853	1.000296	-0.085667	1.001029	-0.142536
2	0.999875	0.021523	1.000140	-0.037115	1.000014	-0.015589
3	1.000999	-0.074533	0.998908	0.256787	0.999905	0.182335

Table 5.3. Clock skew and offset values of implementation of multi-hop Tri-message time synchronization on UASN testbed.

These test-cases also has the provision of sleep period. Time synchronization has been repeated on the testbed with a varying sleep duration. For illustration, sleep duration for test-run 1 is 900 s, for test-run 2 is 1500 s, and for test-run 3 is 600 s. Results of the time synchronization under varying sleep duration has been tabulated in Table 5.4 and Table 5.5.

No.	Hop1		Hop2		Total
	Time-stamps (s)	ERR _{TRI1} (s)	Time-stamps (s)	ERR _{TRI2} (s)	ERR _{TRI(2-hops)} (s)
1	A ₁ = 930.010	ERR _{TRI1} =-0.038265	B ₄ = 1096.167	ERR _{TRI2} = 0.014420	For Tri_I3 = 80 s ERR _{TRI1} =-0.076531 ERR _{TRI(2-hops)} =-0.090951
	B ₁ = 952.078		C ₁ = 1118.244		
	A ₂ = 984.134		B ₅ = 1150.289		
	B ₂ = 962.078		C ₂ = 1128.244		
	A ₃ = 994.134		B ₆ = 1160.289		
	B ₃ = 1016.167		C ₃ = 1182.333		
2	A ₁ = 1537.138	ERR _{TRI1} =0.005375	B ₄ = 1783.246	ERR _{TRI2} = 0.007062	For Tri_I3 = 160 s ERR _{TRI1} =-0.025375 ERR _{TRI(2-hops)} =-0.032437
	B ₁ = 1559.178		C ₁ = 1805.300		
	A ₂ = 1591.214		B ₅ = 1837.346		
	B ₂ = 1569.178		C ₂ = 1815.300		
	A ₃ = 1601.214		B ₆ = 1847.346		
	B ₃ = 1623.246		C ₃ = 1869.388		
3	A ₁ = 640.055	ERR _{TRI1} =-0.009875	B ₄ = 786.088	ERR _{TRI2} = -0.009562	For Tri_I3 = 60 s ERR _{TRI1} =-0.017375 ERR _{TRI(2-hops)} =-0.026937
	B ₁ = 662.065		C ₁ = 808.111		
	A ₂ = 694.070		B ₅ = 840.137		
	B ₂ = 672.065		C ₂ = 818.111		
	A ₃ = 704.070		B ₆ = 850.137		
	B ₃ = 726.088		C ₃ = 872.172		

Table 5.4. Results of multi-hop Tri-message time synchronization under varying sleep durations.

No.	Node B w.r.t. Node A		Node C w.r.t. Node B		Node C w.r.t. Node A	
	Clock Skew β_1	Offset α_1 (s)	Clock Skew β_2	Offset α_2 (s)	Clock Skew $\beta_1\beta_2$	Offset $\beta_2\alpha_1 + \alpha_2$ (s)
1	1.000389	-0.366300	0.999485	0.594462	0.999873	0.228350
2	0.999875	0.197522	0.999812	0.344335	0.999687	0.541820
3	1.000124	-0.080215	1.000187	-0.153552	1.000311	-0.233782

Table 5.5. Clock skew and offset values of multi-hop Tri-message time synchronization on UASN testbed with varying sleep durations.

Results of implementation of multi-hop Tri-message time synchronization shows that the coarse grained synchronization can be achieved by this protocol even under the large and variable delay. Protocol shows comparatively similar performance under varying sleep duration as well. Overall, the novel features of this protocol are as follows:

1. Time-span of synchronization - This protocol requires only three message exchanges between corresponding nodes. Packet size also is very small since it needs only the time-stamps to be included. Total time required for the time synchronization is very less.
2. Energy consumption - Compared to other linear regression based time synchronization protocols, this protocol requires lesser amount of message exchanges for time synchronization. Nodes can use effective sleep-wake schedule since overhearing is not the requirement for this protocol. Hence, this protocol consumes lesser amount of energy as compared to other time synchronization protocols developed for UASN.
3. Simplicity - Protocol design is very simple. It involves very less computational cost.
4. Memory requirement - Since the protocol does not involve regression mechanism, the amount of storage required is less.

5.6 Summary

In this chapter, issues and challenges of time synchronization in UASN has been discussed. Detail analysis of two prominent time synchronization protocols, namely TSHL protocol and Tri-message protocol has been performed using MATLAB simulation platform. The performance

of protocols has been evaluated under varying jitter, delay and time gap from last synchronization. It has been observed that Tri-message protocol provides high precision time synchronization, wherein time estimation error is in the order of tens or hundreds of microseconds under large delay, jitter and time intervals. Comparison of resource utilization of TSHL and Tri-message protocols has also been performed. It has been noted that, Tri-message protocol consumes only 11% energy as compared to TSHL protocol. It also offers significant advantages such as i) simplicity of design, ii) lesser time span of synchronization, and iii) lesser memory requirement, making it very suitable for resource-constrained UASN.

In this work, a modular extension of the Tri-message time synchronization protocol has been proposed and developed for multi-hop networking topology. Simulation results of this extension has been obtained using MATLAB for multi-hop topology. It has been observed that instant error of time estimation gets accumulated over the number of hops and increases almost linearly with increase in hops. Implementation of this protocol has been demonstrated on hardware testbed set-up of UASN. It is first such attempt to implement the protocol on actual hardware testbed of UASN. On the testbed set-up, larger variations have been observed in the delivery of packets, leading to the jitter values in range of tens or hundreds of milliseconds. Because of this jitter distribution, accuracy of Tri-message protocol has been limited to coarse grained time synchronization wherein time estimation error has been in the order of milliseconds.

Chapter 6

Dynamic Cluster-Based TDMA

MAC Protocol for UASN

6.1 Introduction

For development of MAC protocol, various TDMA based MAC protocols designed by researchers in the area of underwater networks have been explored. As stated in Subsection 2.1.2, TDMA based MAC protocols can exploit advantages in terms of simplicity, fairness and energy efficiency. Collisions, idle listening and over-hearing can be avoided in these protocols. Hidden node problem is easily solved without using extra message overhead because neighboring nodes transmit at different time slots.

In this work, a Dynamic Cluster-Based TDMA (DCB-TDMA) MAC protocol has been developed for the three-dimensional UASN proposed in Section 3.3. DCB-TDMA is an energy-aware MAC protocol which can adapt to the dynamic changes in the network topology. Here, channel utilization has been improved with the help of channel reuse by incorporating proper selection and management of transmission power levels. This protocol has also provided important features such as Cluster-Head rotation, and dynamic routing technique. DCB-TDMA protocol has been implemented using SUNSET simulation platform developed by SENSES lab, Sapienza University,

Rome [171, 172]. The details of this protocol, results of simulation runs, and analysis of the results have been provided in this chapter.

Organization of chapter is as follows: In Section 6.2, details of proposed Dynamic Cluster-Based TDMA MAC have been provided. In Section 6.3, pseudo algorithm of DCB-TDMA MAC protocol has been provided. Performance evaluation of DCB-TDMA MAC protocol for various network parameters has been provided in Section 6.4. This protocol has been implemented on SUNSET simulation platform. Results of simulation have been provided in Section 6.5. Detailed analysis of results has also been provided in this section. Performance of DCB-TDMA MAC protocol has been compared with Basic TDMA MAC protocol for the same network architecture. Results of comparison have been provided in Section 6.6. Summary of chapter has been provided in Section 6.7.

6.2 Dynamic Cluster-Based TDMA MAC protocol for three-dimensional UASN

“Dynamic Cluster-Based TDMA (DCB-TDMA)” MAC protocol is a new variant of TDMA based MAC protocol, which has been proposed for three-dimensional UASN. The term “Dynamic” has been used, since the nodes adapt power levels dynamically based on their role in the network. Protocol also allows the nodes to be dynamically added or removed from the network in run-time. And, since the protocol has been deployed on three-dimensional UASN arranged in the form of clusters, the protocol has been termed as “Dynamic Cluster-Based TDMA (DCB-TDMA)” MAC protocol.

Three-dimensional network architecture used for the implementation of DCB-TDMA MAC has been illustrated in Figure 3.2. Various terminologies and node numbering used in this architecture have been described in detail in Section 3.3.

Structure of Master cycle of DCB-TDMA MAC protocol has been shown in Figure 6.1. As shown in this figure, a complete TDMA cycle (termed as Master cycle) consists of two main periods, namely sleep period and wake period. Further, the wake period consists of two phases, Control cycle phase and Data cycle phase. Data cycle phase follows the Control cycle phase. Every node

in the network has pre-assigned time slots for transmission and reception based on the position and role in the network, in both Control and Data cycle phases. Duration of time slot has been decided from Packet Delivery Time (\hat{D}) calculated using (4.3). Time slot duration in the protocol has been set slightly larger than the \hat{D} , in order to accommodate variable propagation delay of underwater acoustic communication.

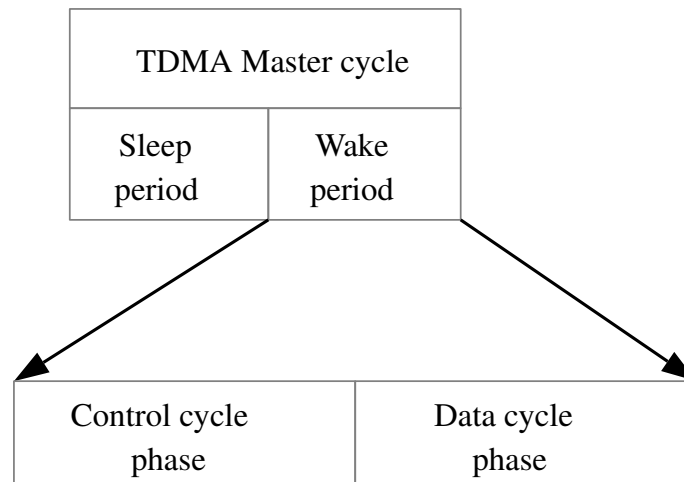


Figure 6.1. Structure of Master cycle of DCB-TDMA MAC protocol.

As shown in Table 6.1, Control cycle of DCB-TDMA MAC protocol starts with BS. It transmits the control information to CH1 in one time slot period (time slot 1 from Table 6.1). CH1 further transmits this information to CH2 in its allotted time slot (time slot 2 from Table 6.1). In time slot 3, CH2 transmits information to CH3 and in time slot 4, CH3 transmits the information to CH4. In this fashion, control information is conveyed to all Cluster-Heads using the backbone link BS-CH1-CH2-CH3-CH4. After receiving control information, each CH starts transmitting this information to their respective CN nodes. This transmission of information can take place in parallel in all the clusters. For example, in time slot 5, CH1 transmits control information to CN(201), CH2 transmits to CN(301), CH3 transmits to CN(401) and CH4 transmits to CN(501). By using subsequent time slots control information is transmitted from CH nodes to all remaining CN nodes in parallel. Parallel transmission is possible across multiple clusters using proper setting of transmission power levels as described in Section 3.5. Usually, parallel transmission in multiple clusters is done by assigning multiple frequency of communication at each cluster. In case of underwater communication, various communication characteristics such as transmission loss varies greatly with frequency limiting the usable bandwidth. In this network, a single frequency channel has been considered. In this architecture, it has been assumed that node has two different

Time slot 1	Time slot 2	Time slot 3	Time slot 4	Time slot 5	Time slot 6	Time slot 7	Time slot 13	Time slot 14	Five time slot period (time slots 15 to 19)
BS (100) to CH1 (200)	CH1 (200) to CH2 (300)	CH2 (300) to CH3 (400)	CH3 (400) to CH4 (500)	CH1 (200) to CN (201)	CH1 (200) to CN (202)	CH1 (200) to CN (203)	CH1 (200) to CN (209)	CH1 (200) to CN (210)	For future expansion
				CH2 (300) to CN (301)	CH2 (300) to CN (302)	CH2 (300) to CN (303)	CH2 (300) to CN (309)	CH2 (300) to CN (310)	For future expansion
				CH3 (400) to CN (401)	CH3 (400) to CN (402)	CH3 (400) to CN (403)	CH3 (400) to CN (409)	CH3 (400) to CN (410)	For future expansion
				CH4 (500) to CN (501)	CH4 (500) to CN (502)	CH4 (500) to CN (503)	CH4 (500) to CN (509)	CH4 (500) to CN (510)	For future expansion

Table 6.1. Structure of Control cycle phase of DCB-TDMA MAC protocol.

power levels, that is, lower power level (P_L) to communicate at shorter distances (around tens of meters), and higher power level (P_H) in order to communicate till larger distances (up to hundreds of meters). When a node is selected as CH, it chooses the P_H level to communicate to another CH, which is located at a distance of 500 m from it. It selects P_L level while communicating to its cluster member nodes located in a periphery of 10 m. Power level P_L can be chosen using analysis in Section 3.5 in such a way, that it does not interfere in parallel ongoing communication at distance 500 m away. With the help of such power management, CH to CN transfer happens parallel among various clusters. Overall, in a Control cycle phase, network management information is transferred from BS to CH and then from CH to respective CN nodes.

Table 6.1 shows that a time period equivalent to five time slots is appended to the Control cycle. These time slots can be utilized for accommodating five extra nodes in each cluster during the possible future expansion of network. Duration of this time period can be decided by network designer based on probable expansion plan of the network. Till the new nodes are added in the

network, this time period is not utilized by the network and can be considered as part of sleep period.

Time slot 20	Time slot 21	Time slot 22	Time slot 28	Time slot 29	Five time slot period (time slots 30 to 34)	Time slot 35	Time slot 36	Time slot 37	Time slot 38
CN (201) to CH1 (200)	CN (202) to CH1 (200)	CN (203) to CH1 (200)	CN (209) to CH1 (200)	CN (210) to CH1 (200)	For future expansion	CH4 (500) to CH3 (400)	CH3 (400) to CH2 (300)	CH2 (300) to CH1 (200)	CH1 (200) to BS (100)
CN (301) to CH2 (300)	CN (302) to CH2 (300)	CN (303) to CH2 (300)	CN (309) to CH2 (300)	CN (310) to CH2 (300)	For future expansion				
CN (401) to CH3 (400)	CN (402) to CH3 (400)	CN (403) to CH3 (400)	CN (409) to CH3 (400)	CN (410) to CH3 (400)	For future expansion				
CN (501) to CH4 (500)	CN (502) to CH4 (500)	CN (503) to CH4 (500)	CN (509) to CH4 (500)	CN (510) to CH4 (500)	For future expansion				

Table 6.2. Structure of Data cycle phase of DCB-TDMA MAC protocol.

All sensor nodes (CH as well as CN) collect sensor data at regular intervals. Data cycle starts after the Control cycle. At the beginning of Data cycle, CN nodes transfer their data information to their respective CH. CN nodes use the P_L level for data transmission; hence data transmission can also take place in parallel at different clusters. For example, as shown in Table 6.2, CN(201) transmits data to CH1 in time slot 20. In the same time slot, CN(301) transmits data to CH2, CN(401) transmits data to CH3 and CN(501) transmits data to CH4. Subsequently, remaining CN nodes transmits data information to respective CH nodes. Period of five time slots follows this transmissions at every cluster for the future expansion. After this time period, data transmission is done using backbone link in reverse order, i.e., CH4-CH3-CH2-CH1-BS. When the data is received by BS, it sends this data by using radio link to an on-shore station.

As shown in Table 6.2, a time period equivalent to five time slots is available in Data cycle phase for future expansion of the network. This time period is provided before the start of transmission on backbone link of the network, and has the same duration as that of time period in Control cycle phase (for future expansion). When a new node is added in any of the cluster, it occupies a time slot in Control cycle as well as Data cycle phase. For example, if a new node is to be added in cluster 2, it will be numbered as node 311 and will occupy time slot 15 of cluster 2 in Control cycle phase and time slot 30 of cluster 2 in Data cycle phase. Further addition of node in the same cluster will have node number 312, time slot 16 of cluster 2 in Control cycle phase and time slot 31 of cluster 2 in Data cycle phase. Rest of the structure of the network remains undisturbed.

In the DCB-TDMA MAC, since the time slots for transmission and reception are pre-scheduled, nodes can turn the radio off and go into sleep mode for remaining time duration of cycle to conserve energy. Once the Data cycle is over, network go to sleep mode for the equal duration of that of wake period. In this way, Master cycle of DCB-TDMA MAC have duty cycle close to 50%.

In following subsection, details of various types of control information transmitted from BS to CN nodes in Control cycle phase have been provided. Information collected in Data cycle depends on the type of sensors used in the application.

6.2.1 Various types of control information

In a Control cycle, various types of control information can be transmitted as per the requirement of network. Various types are 1) regular information, 2) node addition, 3) node deletion, 4) request of energy values and 5) Cluster-Head selection. These different types have been explained here briefly.

1. **Regular information:** BS receives control information from on-shore control station regularly over a radio link. BS transmits this control information to all CN nodes at regular interval in every Master cycle. This information can be simple location information or timing update and so on.

2. **Node addition information:** Any new node with pre-programmed time slot and cluster information can be added in the network at any desired time. When a new node is deployed, BS transmits information about newly joined node in a Control cycle phase (along with regular control information) to CH nodes in subsequent Master cycle. CH nodes forward this information further to respective CN nodes as defined earlier. New nodes receive the control information from their corresponding CH in one of the slots assigned under future expansion slots. In this manner, any number of nodes can be added in multiple clusters at a same time, depending upon availability of free slots.
3. **Node deletion information:** BS can remove certain CN nodes based on information of residual energy of the node or in case of any malfunction. This information can also be collected run-time from the network. If the node is not responding for more than 3-4 cycles then such node can be considered as non-active or dead. Information regarding dead nodes is transmitted from BS to CN nodes using forward link. Idle time slot of this node can be allocated to other nodes added in future.
4. **Request of energy values:** - In DCB-TDMA MAC protocol, BS collects the energy information from all the nodes in the network at regular intervals of time. BS uses this type of control information requesting nodes to send the values of residual energy to the BS via CH nodes on reverse link.
5. **CH selection information:** CH nodes consume more energy in any given cycle as the amount of data exchange as well as energy required for transmission is higher. Residual energy of CH (received at BS) is compared against threshold value(s). If the CH energy is below the threshold, CH is rotated in subsequent Master cycle. BS can appoint a CN node with highest residual energy available as CH in its corresponding cluster and forwards the information to a new CH using the backbone link of the network. Selection of new CH involves a handover process. During the handover, time slots of old CH nodes are transferred to new CH. Control and data information is re-routed via new CH. Also new CH requires to use varying power levels.

Pseudo algorithm of basic structure of proposed DCB-TDMA MAC protocol has been provided in the following section.

6.3 Pseudo algorithm of DCB-TDMA MAC protocol for UASN

Various terms used in the pseudo algorithm for implementation of DCB-TDMA MAC protocol are as follows:

BS - Base-Station, CH - Cluster-Head, CN - Cluster Nodes, N - node (BS or CH or CN), and TS_i - Time Slot of i th node.

Different types of control and data packets which have been used for various types of control information in the algorithm are as follows:

1. **Regular information:** - Control packet used for transmitting regular information is denoted in algorithm as CTP(Reg_info). Upon receiving this packet, all nodes prepare data packet denoted as DTP(Reg_data) and transmits this packet to CH nodes in their respective time slots.
2. **Node addition information:** - Control packet used for transmitting node addition information is denoted in algorithm as CTP(Node_addition). Upon receiving this packet, all nodes prepare data packet denoted as DTP(Reg_data) and transmits this packet to CH nodes in their respective time slots.
3. **Node deletion information:** - Control packet in case of node deletion is denoted as CTP(Node_deletion). All active nodes send DTP(Reg_data) packet to CH nodes in their respective time slots.
4. **Request of energy values:** - Control packet used by BS for requesting energy information from all the network nodes is termed as CTEP(Eng_info). All nodes append the energy information to the data while transmitting the data packet in the following Data cycle. Corresponding data packet is denoted as DTEP(Eng_info).
5. **CH selection information:** - BS sends the control information using CTP(CHR_info) packet to all the nodes in the network informing the CH rotation. In the following Data cycle, nodes send the data packet using DTP(Reg_data) packet.

Overall algorithm is stated as follows:

1. Initial address assignment

BS = 100

CH1 = 200, CH2 = 300, CH3 = 400, CH4 = 500

CN1 = {201, 202, 203, ..., 210}

CN2 = {301, 302, 303, ..., 310}

CN3 = {401, 402, 403, ..., 410}

CN4 = {501, 502, 503, ..., 510}

2. Control cycle

If (Regular control information)

If ($TS_i \in N_i$) then

BS.Tx_CTP(Reg_info) \Rightarrow CH1

CH1.Tx_CTP(Reg_info) \Rightarrow CH2

CH2.Tx_CTP(Reg_info) \Rightarrow CH3

CH3.Tx_CTP(Reg_info) \Rightarrow CH4

CH1.Tx_CTP(Reg_info) \Rightarrow CN1 || CH2.Tx_CTP(Reg_info) \Rightarrow CN2 ||

CH3.Tx_CTP(Reg_info) \Rightarrow CN3 || CH4.Tx_CTP(Reg_info) \Rightarrow CN4

Endif

Endif

If(Node addition information)

BS \rightarrow Update table entry

If ($TS_i \in N_i$) then

BS.Tx_CTP(Node_addition) \Rightarrow CH1

CH1.Tx_CTP(Node_addition) \Rightarrow CH2

CH2.Tx_CTP(Node_addition) \Rightarrow CH3

CH3.Tx_CTP(Node_addition) \Rightarrow CH4

CH1.Tx_CTP(Node_addition) \Rightarrow CN1 || CH2.Tx_CTP(Node_addition) \Rightarrow CN2 ||

CH3.Tx_CTP(Node_addition) \Rightarrow CN3 || CH4.Tx_CTP(Node_addition) \Rightarrow CN4

Endif

Endif

```

If(Node removal information) or if ( $3 \times (N_i.Rx\_DTP == \text{False})$ )
BS  $\rightarrow$  Update table entry
If ( $TS_i \in N_i$ ) then
BS.Tx_CTP(Node_deletion)  $\Rightarrow$  CH1
CH1.Tx_CTP(Node_deletion)  $\Rightarrow$  CH2
CH2.Tx_CTP(Node_deletion)  $\Rightarrow$  CH3
CH3.Tx_CTP(Node_deletion)  $\Rightarrow$  CH4
CH1.Tx_CTP(Node_deletion)  $\Rightarrow$  CN1 || CH2.Tx_CTP(Node_deletion)  $\Rightarrow$  CN2 ||
CH3.Tx_CTP(Node_deletion)  $\Rightarrow$  CN3 || CH4.Tx_CTP(Node_deletion)  $\Rightarrow$  CN4
Endif
Endif

```

```

If(Requesting Energy information)
If ( $TS_i \in N_i$ ) then
BS.Tx_CTEP(Eng_info)  $\Rightarrow$  CH1
CH1.Tx_CTEP(Eng_info)  $\Rightarrow$  CH2
CH2.Tx_CTEP(Eng_info)  $\Rightarrow$  CH3
CH3.Tx_CTEP(Eng_info)  $\Rightarrow$  CH4
CH1.Tx_CTEP(Eng_info)  $\Rightarrow$  CN1 || CH2.Tx_CTEP(Eng_info)  $\Rightarrow$  CN2 ||
CH3.Tx_CTEP(Eng_info)  $\Rightarrow$  CN3 || CH4.Tx_CTEP(Eng_info)  $\Rightarrow$  CN4
Endif
Endif

```

```

If( $(CH_i.Energy < \text{Threshold})$  and  $\text{If}(CN_i.Energy > \text{Threshold})$ )
 $CN_i \leftarrow CH_i$ 
 $CN_i(\text{Highest residual energy}) \rightarrow \text{new } CH_i$ 
BS  $\rightarrow$  Update table Entry
If ( $TS_i \in N_i$ ) then
BS.Tx_CTP(CHR_info)  $\Rightarrow$  CH1
CH1.Tx_CTP(CHR_info)  $\Rightarrow$  CH2
CH2.Tx_CTP(CHR_info)  $\Rightarrow$  CH3

```

CH3.Tx_CTP(CHR_info) \Rightarrow CH4

CH1.Tx_CTP(CHR_info) \Rightarrow CN1 || CH2.Tx_CTP(CHR_info) \Rightarrow CN2 ||

CH3.Tx_CTP(CHR_info) \Rightarrow CN3 || CH4.Tx_CTP(CHR_info) \Rightarrow CN4

Endif

Endif

3. Data cycle

If ((N_i .Rx_CTP(Reg_info) ==True) or (N_i .Rx_CTP(Node_addition) ==True) or
(N_i .Rx_CTP(Node_deletion) ==True) or (N_i .Rx_CTP(CHR_info) ==True)) and

If ($TS_i \in N_i$)

CN1.Tx_DTP(Reg_data) \Rightarrow CH1 || CN2.Tx_DTP(Reg_data) \Rightarrow CH2 ||

CN3.Tx_DTP(Reg_data) \Rightarrow CH3 || CN4.Tx_DTP(Reg_data) \Rightarrow CH4

CH4.Tx_DTP(Reg_data) \Rightarrow CH3

CH3.Tx_DTP(Reg_data) \Rightarrow CH2

CH2.Tx_DTP(Reg_data) \Rightarrow CH1

CH1.Tx_DTP(Reg_data) \Rightarrow BS

Endif

Endif

If (N_i .Rx_CTEP(Eng_info) ==True) and

If ($TS_i \in N_i$)

CN1.Tx_DTEP(Eng_info) \Rightarrow CH1 || CN2.Tx_DTEP(Eng_info) \Rightarrow CH2 ||

CN3.Tx_DTEP(Eng_info) \Rightarrow CH3 || CN4.Tx_DTEP(Eng_info) \Rightarrow CH4

CH4.Tx_DTEP(Eng_info) \Rightarrow CH3

CH3.Tx_DTEP(Eng_info) \Rightarrow CH2

CH2.Tx_DTEP(Eng_info) \Rightarrow CH1

CH1.Tx_DTEP(Eng_info) \Rightarrow BS

Endif

Endif

Along with this basic structure of DCB-TDMA MAC protocol, further additions can be done based on applications requirement and nature of complexity. Details of certain possible features for improvement have been provided in the next subsection.

6.3.1 Additional features for improvement of network performance

Additional features as explained below can be incorporated to increase the overall efficiency and robustness of the network.

1. **Assignment of multiple slots to a node:** In the proposed protocol, some empty time slots have been provided to accommodate new nodes for any future expansion in the network. Instead of these slots idling, they can be assigned to existing nodes as additional time slots for additional information exchange. Similarly, when nodes of the existing network dies, their slots can be re-assigned to nodes that require multiple slots.
2. **Use of bidirectional link in control cycle:** In the proposed architecture, though there is a provision for bidirectional communication link, currently the Control cycle has been used for unidirectional transmission. Control mechanism can be made as bidirectional to accommodate protocols that have implicit or explicit handshaking mechanism. This is especially useful for time synchronization, localization or power management protocols.
3. **Query based data collection:** It is also possible to piggyback certain queries on control packets sent from BS to CN nodes. Additionally, control information can be sent selectively to those nodes from which data or any other information needs to be collected.
4. **Failure of backbone link:** Handshake based control information transmission can be used to detect backbone failures. If the acknowledgment is not received from the respondent within the stipulated time then BS can retransmit the control information for certain number of tries. If it still does not receive acknowledgment, then it can appoint one among the CN nodes as a CH and can resend the information to new CH. Same algorithm can be adapted by each CH node as and when they are sending information down the backbone link of the network. Detection of failure using handshake, number of retries and appointment of new

CH within the same cycle necessitates the provision of extra time slots being available, for every important link in the network.

In the Section 6.4, performance evaluation of DCB-TDMA MAC protocol has been provided for various network parameters.

6.4 Performance evaluation of DCB-TDMA MAC protocol

In this section, evaluation of various important network parameters such as maximum MAC payload, total delay, throughput, Packet Delivery Ratio (PDR) and channel utilization has been performed for DCB-TDMA MAC protocol.

6.4.1 Maximum size of MAC payload

It has been assumed that in one time slot of DCB-TDMA MAC protocol, one FEC (Forward Error Correction) encoded MAC frame is transmitted. MAC frame consists of MAC header and MAC payload. This time slot also has an extra time duration to accommodate propagation delay. In addition to this, an interval of guard time is included to accommodate factors such as 1) variations in propagation delay, 2) mismatch in time synchronization, 3) MAC processing delay, and 4) switching-on time. Node's packet will encounter collision, if the combined effect of these factors exceeds the guard time interval. For calculating maximum MAC payload in one time slot, following notations have been used:

- Size of MAC header = MH bytes.
- Size of MAC payload = MP bytes.
- FEC code rate = FEC_{CR}
- Input to the FEC block = FEC_{in} bytes.
- Output of the FEC block = FEC_{out} bytes.

- Duration of one time slot = T_S s.
- Duration of guard time = GT s.
- Bit Rate = R bps.
- Propagation time = \hat{P} s.

Input to the FEC coder block is MAC header + MAC payload; therefore,

For one time slot of DCB-TDMA MAC,

$$FEC_{in} = MH + MP \quad (6.1)$$

Output of the FEC block would therefore be,

$$FEC_{out} = \frac{FEC_{in}}{FEC_{CR}} \quad (6.2)$$

For one time slot of DCB-TDMA MAC,

$$T_S = GT + \hat{P} + \frac{FEC_{out}}{R} \quad (6.3)$$

From (6.1), (6.2) and (6.3), size of MAC payload can be obtained as follows:

$$MP = \left(\left[(T_S - GT - \hat{P}) R \right] FEC_{CR} \right) - MH \quad (6.4)$$

Since the \hat{P} on vertical link (of distance 500 m) is higher than the \hat{P} on horizontal link (of distance 10 m), maximum payload size has been calculated based on the higher value of \hat{P} . In current scenario, data aggregation policy has not been implemented on the network. It has been assumed that the packet (or frame) size is equal on both vertical and horizontal links. Effective data aggregation policy can be implemented if the packet size and time slot is allowed to be variable and adaptive. Smaller packets can be transmitted on horizontal links with smaller time slots. On vertical link, packet size as well as time slot can be gradually incremented depending on efficiency of data aggregation algorithm. In current scenario, \hat{P} on vertical link restricts the size of maximum MAC payload. Considering propagation speed as 1500 m/s, it can be verified that $\hat{P} = 0.3333$ s on

a vertical link of 500 m distance. Further, assuming $T_S = 10$ s, $GT = 0.5$ s, $FEC_{CR} = 0.96$, $MH = 10$ bytes, $R = 100$ bps, value of MP can be obtained as 100 bytes.

Maximum MAC payload size for various bit rate values have been tabulated in Table 6.3. It can be observed here, that the maximum MAC payload increases with the increase in R, provided that the time slot duration is fixed. For the application at hand, if the MAC payload size is smaller, then the time slot duration should be shortened.

Sr. No.	Bit Rate R (bps)	Max. MAC payload Size (MP) (bytes)
1	100	100
2	500	540
3	1000	1090
4	1500	1640
5	2000	2190

Table 6.3. Maximum MAC payload in DCB-TDMA MAC protocol for various bit rates.

6.4.2 End-to-end delay

Three-dimensional topology of the proposed UASN architecture can be simplified for the calculation of end-to-end delay (or latency) as shown in Figure 6.2. As explained in Section 6.2, in the Control cycle phase, initially control information is transmitted on a backbone link (BS-CH1-CH2-CH3-CH4). CH nodes then transmit the information to respective CN nodes. A link from BS to CN nodes via CH nodes has been termed as forward link as stated in Section 3.3. The end-to-end delay on this forward link denoted as $Delay_{FL}$ can be calculated as follows:

$$Delay_{FL} = (L_T - 1) \times T_S + XX \times T_S \quad (6.5)$$

wherein, L_T indicates the total number of levels in the network. XX denotes a two-digit number of node in the cluster as defined in Section 3.2.

Equation (6.5) indicates that the delay depends on total number of levels in the network. But the delay is independent of a particular cluster number of a node. For example, CN(201) experiences the same delay as that of CN(301). CN(410) experiences same delay on forward link as that of

CN(510). On the other hand, delay depends on the node's number within the cluster. For example, CN(205) experiences more delay in receiving information as compare to CN(201).

End-to-end delay has been calculated on forward link for various bit rates. The results have been tabulated in Table 6.4. For this analysis, following values have been assumed: MP = 100 bytes, MH = 10 bytes, GT = 0.5 s. and $L_T = 5$. Time slot duration T_S is calculated using (6.3). Since, delay is independent of the number of level, node 1 has been denoted as L01 and node 10 has been denoted as L10 in the table. Average is taken for these extreme values of the delay. All time values in the table are in seconds.

Bit Rate R (bps)	Time slot T_S (s)	Delay _{FL} for node L01 (s)	Delay _{FL} for node L10 (s)	Average Delay _{FL} (s)
100	10	50	140	95
500	2.66	13.3	37.24	25.27
1000	1.75	8.75	24.5	16.63
1500	1.44	7.2	20.16	13.68
2000	1.29	6.45	18.06	12.26

Table 6.4. End-to-end delay on forward link in DCB-TDMA MAC protocol for various bit rates.

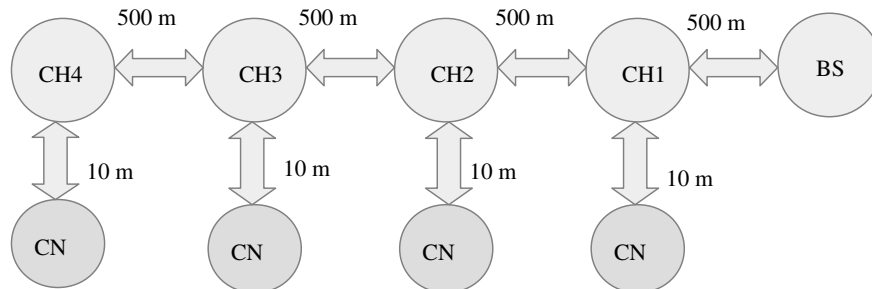


Figure 6.2. Simplified network architecture for delay calculation.

End-to-end delay on reverse link denoted by Delay_{RL} (from CN nodes to BS via CH nodes), can be obtained using equation as follows:

$$\text{Delay}_{\text{RL}} = (L_T - 1) \times T_S + (T_{\text{MAX}} - XX + 1) \times T_S \quad (6.6)$$

wherein T_{MAX} indicates maximum number of slots in the Data cycle including the slots reserved for future expansion. Delay on reverse link has been tabulated for various bit rates in Table 6.5.

Further, average of the delay values has been taken on forward link and reverse link for the calculation of throughput in the following subsection. Average values of delay corresponding

Bit Rate R (bps)	Time slot T_S (s)	Delay _{RL} for node L01 (s)	Delay _{RL} for node L10 (s)	Average Delay _{RL} (s)
100	10	190	100	145
500	2.66	50.54	26.6	38.57
1000	1.75	33.25	17.5	25.38
1500	1.44	27.36	14.4	20.88
2000	1.29	24.51	12.9	18.71

Table 6.5. End-to-end delay on reverse link in DCB-TDMA MAC protocol for various bit rates.

to various bit rates have been tabulated in Table 6.6. It can be observed that, with increase in R, average delay decreases. It is mainly because, with increase in R, time slot duration decreases for the fixed payload size of transmission.

Sr. No.	Bit Rate R (bps)	Average Delay (s)
1	100	120
2	500	31.92
3	1000	21.00
4	1500	17.28
5	2000	15.49

Table 6.6. Average end-to-end delay in DCB-TDMA MAC protocol for various bit rates.

6.4.3 Throughput

Depending on R and average delay values, throughput of the network has been calculated as follows:

$$\text{Throughput} = \frac{\text{Payload in bits}}{\text{Average Delay}} \text{ bps} \quad (6.7)$$

Using a fixed payload size of MP = 100 bytes and average delay values from Table 6.6, throughput for various R values have been calculated and tabulated in Table 6.7. It can be observed that the throughput increases considerably with the increase in R. It is mainly because of decrease in average delay with increasing R as observed in Table 6.6.

Sr. No.	Bit Rate R (bps)	Throughput (bps)
1	100	6.67
2	500	25.06
3	1000	38.10
4	1500	46.30
5	2000	51.65

Table 6.7. Throughput in DCB-TDMA MAC protocol for various bit rates.

6.4.4 Packet Delivery Ratio

In DCB-TDMA MAC protocol, since the transmissions and receptions of packets have been pre-scheduled, probability of collision is zero. By utilizing a guard time in every time slot, it is guaranteed that multiple packets will never arrive at any receiver simultaneously. This ensures that the Packet Delivery Ratio (PDR) is 100% in the network. It also indicates that no retransmissions will arise out of packet collisions, further saving the energy of a node. This protocol does not require overhearing or idle listening, and so these schemes have been completely avoided. There might still be certain packet losses due to unpredictable channel behavior or due to malfunctioning of a node. Few packet losses (of control or data packets) may affect the accuracy of information in a current cycle, but does not hamper the overall network performance or functioning.

6.4.5 Channel utilization

Channel utilization is one of the important network performance criteria. In DCB-TDMA MAC, since parallel communication at horizontal level is possible with effective management of power level, it increases channel reuse and thus channel utilization effectively. Channel utilization can be measured as follows:

$$\text{Channel utilization} = \frac{\hat{T}}{\text{Duration of Data cycle}} \times \text{channel reuse factor} \quad (6.8)$$

Since, proposed three-dimensional network architecture has 4 clusters, channel reuse factor has been taken as 4. \hat{T} can be obtained using (4.1). Duration of Data cycle can be obtained using following equation:

$$\text{Duration of Data cycle} = (T_{MAX}) \times T_S \quad (6.9)$$

Considering packet size = 100 bytes ($P_S = MP$ for simplicity), channel utilization has been tabulated in Table 6.8 for various R values.

Sr. No.	Bit Rate R (bps)	Channel utilization (%)
1	100	16.84
2	500	12.66
3	1000	9.62
4	1500	7.80
5	2000	6.53

Table 6.8. Channel utilization in DCB-TDMA MAC protocol for various bit rates.

From Table 6.8, it can be observed that the channel utilization decreases with increase in R since the \hat{T} decreases. Though the time slot duration T_S also reduces because of increase in R, it should be noted that the significant duration of time slot is provided for accommodating large propagation delay and the guard time. For example, when R is 2000 bps, time slot duration is 1.29 s. In this T_S of 1.29 s., duration of 0.33 s. is for propagation delay and GT is of 0.5 s. It can be stated easily that the channel utilization is restricted because of large and variable propagation delay.

DCB-TDMA MAC protocol has been simulated on the SUNSET simulation platform. Results of simulation and analysis of results have been provided in Section 6.5.

6.5 Simulation of DCB-TDMA MAC protocol

A three-dimensional network architecture of UASN as shown in Figure 3.2 has been implemented on SUNSET simulation platform. Proposed DCB-TDMA MAC protocol has been simulated on this network. As shown in Figure 3.2, there are total 5 levels from top to bottom, separated by a distance of 500 m. At level 1, on sea-surface, a BS node has been deployed. At every subsequent level, a cluster of 11 nodes have been deployed in circular fashion. One among these 11 nodes act as Cluster-Head node initially. Simulation has been run in terms of repetitive Master cycle, where each Master Cycle consists of Control cycle phase and Data cycle phase. One example each for node addition, node deletion and Cluster-Head rotation has been implemented as follows:

1. **Node addition** -In second Master cycle, two new nodes have been added as part of network expansion scheme. One node has been added in cluster 1 and another node in cluster 2.
2. **Node deletion** - In third Master Cycle, one node each has been removed from cluster 2 and cluster 3.
3. **Cluster-Head rotation** -Depending on the energy consumption with nodes and energy threshold setting, a Cluster-Head rotation has been simulated in fourth Master Cycle.

In the following subsection, calculations have been provided for determining the duration of time slot in DCB-TDMA MAC protocol.

6.5.1 Setting the duration of time slot

For network simulation, following parameters have been considered for a case study:

- Packet Size (P_S) = 110 bytes.
- Average propagation Speed of acoustic signal in water (c) = 1500 m/s.
- Bit Rate (R) = 100 bps.

Using (4.1), (4.2), and (4.3), parameters such as Packet Transmission Time (\hat{T}), Propagation Time (\hat{P}) and Packet Delivery Time (\hat{D}) can be obtained.

In this scenario,

$$\hat{T} = 8.8 \text{ s.}$$

$$\hat{P} \text{ (for horizontal link of 10 m) is } 0.0066 \text{ s.}$$

$$\hat{P} \text{ (for vertical link of 500 m) = } 0.3333 \text{ s.}$$

$$\hat{D} \text{ (Horizontal link) = } 8.8066 \text{ s.}$$

$$\hat{D} \text{ (Vertical link) = } 9.1333 \text{ s.}$$

Based on these calculations of delay, and considering guard time of 0.5 s, time slot duration has been taken as 10 s for all the nodes in the network.

In the next subsection, detailed calculations have been provided for setting power levels of the nodes.

6.5.2 Setting the power levels of nodes

It has been assumed that modem has capability of dynamically varying the power levels. Modem uses P_H level to communicate on a vertical link and P_L level to communicate with CN nodes. As described in Section 3.5, power levels have been chosen as $P_H = 0.01$ W and $P_L = 10^{-6}$ W. For the simulation, various parameters used have been tabulated in Table 6.9.

Sr. No.	Parameter	Value
1	Frequency of operation (f)	25 kHz
2	Bandwidth (B)	5 kHz
3	Wind speed (w)	7 m/s
4	Shipping activity factor (sh)	0.5
5	Temperature of sea-surface (T)	20 °C
6	Salinity (S)	35 ppt
7	Acidity (pH)	8

Table 6.9. Parameters used in simulation of DCB-TDMA MAC protocol on SUNSET simulation platform.

Using (3.14) to (3.18), Noise power level can be obtained as follows:

- Ambient Noise Level, NL = 41.61 dB.
- Noise Power Level (NL + 10 log B) = 78.59 dB.

Using (3.1), (3.4 - 3.13) and (3.21 - 3.27), various parameters of signal power have been obtained. These parameters have been tabulated in Table 6.10.

Transmission Power P_t	SL (dB)	RL (dB) for range of 500 m	RL (dB) for range of 10 m	SNR (dB) for range of 500 m	SNR (dB) for range of 10 m
$P_H = 0.01$ W	150.77	95.07	130.73	16.48	52.14
$P_L = 10^{-6}$ W	110.77	55.07	90.74	-23.52	12.15

Table 6.10. SNR calculations for simulation of DCB-TDMA MAC protocol in SUNSET simulation platform.

Using (3.29), value of SINR can be obtained for this scenario. Using values from Table 6.10 and considering interference from ongoing parallel communication in two adjacent clusters, the SINR value is 12.11 dB. It indicates that since the interference is negligibly small, parallel intra-cluster communication can be used in the network, improving channel utilization.

In Subsection 6.5.3, the simulation results have been illustrated in terms of actual time stamps while transmitting and receiving the packets in network.

6.5.3 Results and analysis of simulation of DCB-TDMA MAC protocol

After setting various parameters of simulation, the DCB-TDMA MAC protocol has been simulated on SUNSET simulation platform in terms of multiple Master cycles. Illustration of actual time slot structure of one run of Control and Data cycle using SUNSET simulation has been represented in Table 6.11 and Table 6.12 respectively. Details of second Master cycle have been illustrated in these tables, wherein one node each has been added in cluster 1 and 2. Actual time stamp values of packet transmission and reception have been provided with the help of simulation tool.

In Table 6.11 and Table 6.12, first row indicates the number of time slots. For simplicity, time slot numbering in every Master cycle starts from 1. Control cycle phase has time slots from 1 to 19, whereas Data cycle phase has time slots 20 to 38. Time slots 15 to 19 of Control cycle and time slots 30 to 34 of Data cycle have been provided to accommodate new nodes in the network at later stages. In second row, time stamps of transmission and reception of packet have been provided. In table, S.T. stands for Send Time (at transmitter node) and R.T. stands for Receive Time of the packet (at receiver node) respectively. All time values are in seconds. Nodes are indicated by the terminology as well as the node number. For example Base-Station node having node number 100 is represented as BS(100).

As shown in Table 6.11, BS(100) has sent a control packet at time 775.0 s in time slot 1, which has been received by CH1(200) at time 784.13 s. Further, CH1(200) has transmitted the packet in time slot 2 at time 785.0 s, which has been received by CH2(300) at 794.13 s. In this manner, control information has been sent to all CH nodes by utilizing multi-hop backbone link (BS-CH1-CH2-CH3-CH4). Later, CH1(200) has sent a control packet in time slot 5 at time 815.0 s. which has been received by CN(201) at time 823.8 s. Note that, in the same time slot 5,

Time slot 1	Time slot 2	Time slot 3	Time slot 4	Time slot 5	Time slot 6	Time slot 13	Time slot 14	Five time slot period (time slots 15 to 19)	
S.T. 775.0	S.T. 785.0	S.T. 795.0	S.T. 805.0	S.T. 815.0	S.T. 825.0	S.T. 895.0	S.T. 905.0	S.T. 915.0	Sleep period till 965.0
R.T. 784.13	R.T. 794.13	R.T. 804.13	R.T. 814.13	R.T. 823.8	R.T. 833.8	R.T. 903.8	R.T. 913.8	R.T. 923.8	
BS (100) to CH1 (200)	CH1 (200)	CH2 (300)	CH3 (400)	CH1 (200)	CH1 (200)	CH1 (200)	CH1 (200)	CH1 (200)	Sleep period of 4 time slots
	to CH2 (300)	to CH3 (400)	to CN (201)	to CN (202)	to CN (209)		to CN (210)	to CN (211)		
	CH2 (300)	CH2 (300)	CH2 (300)	CH2 (300)	CH2 (300)		CH2 (300)	CH2 (300)		
	to CN (301)	to CN (302)	to CN (309)	to CN (310)	to CN (311)		to CN (311)	to CN (311)		
CH3 (400)	CH3 (400)	CH3 (400)	CH3 (400)	CH3 (400)	CH3 (400)	CH3 (400)	CH3 (400)	CH3 (400)	Sleep period of 5 time slots	
to CN (401)	to CN (402)	to CN (409)	to CN (410)	to CN (409)	to CN (410)	to CN (409)	to CN (410)	to CN (410)	to CN (410)	
CH4 (500)	CH4 (500)	CH4 (500)	CH4 (500)	CH4 (500)	CH4 (500)	CH4 (500)	CH4 (500)	CH4 (500)	CH4 (500)	Sleep period of 5 time slots
to CN (501)	to CN (502)	to CN (509)	to CN (510)	to CN (509)	to CN (510)	to CN (509)	to CN (510)	to CN (510)	to CN (510)	

Table 6.11. Actual time slots of Control cycle phase of second Master Cycle in simulation of DCB-TDMA MAC protocol. (Values of S.T. and R.T. are in seconds.)

CH1(200) has sent a control packet to CN(201), CH2(300) has sent packet to CN(301), CH3(400) has sent a packet to CN(401) and CH4(500) has also sent a packet to CN(501). This indicates the parallel intra-cluster communication in the network. Time slots 5 to 14 has been used by CH nodes to send control information to all 10 CN nodes in the respective clusters. In this particular example, two nodes have been added in the network, CN(211) in cluster 1 and CN(311) in cluster 2. Control packets have been sent to these nodes in one of the time slots reserved for future expansion. In this case, time slot 11 has been used. In this fashion, control information has percolated to all the CN nodes of all the clusters in Control cycle phase of the algorithm. Once the CN nodes receive control packets, they can initiate procedure for sensing data along with aggregation and creation

Time slot 20	Time slot 21	...	Time slot 28	Time slot 29	Five time slot period (time slots 30 to 34)		Time slot 35	Time slot 36	Time slot 37	Time slot 38
S.T. 965.0	S.T. 975.0	...	S.T. 1045.0	S.T. 1055.0	S.T. 1065.0	Sleep period till 1115.0	S.T. 1115.0	S.T. 1125.0	S.T. 1135.0	S.T. 1145.0
R.T. 973.8	R.T. 983.8	...	R.T. 1053.8	R.T. 1063.8	R.T. 1073.8		R.T. 1124.13	R.T. 1134.13	R.T. 1144.13	R.T. 1145.13
CN (201) to CH1 (200)	CN (202) to CH1 (200)	...	CN (209) to CH1 (200)	CN (210) to CH1 (200)	CN (211) to CH1 (200)	Sleep period of 4 time slots	CH4 (500) to CH3 (400)	CH3 (400) to CH2 (300)	CH2 (300) to CH1 (200)	CH1 (200) to BS (100)
CN (301) to CH2 (300)	CN (302) to CH2 (300)	...	CN (309) to CH2 (300)	CN (310) to CH2 (300)	CN (311) to CH2 (300)	Sleep period of 4 time slots				
CN (401) to CH3 (400)	CN (402) to CH3 (400)	...	CN (409) to CH3 (400)	CN (410) to CH3 (400)	Sleep period of 5 time slots					
CN (501) to CH4 (500)	CN (502) to CH4 (500)	...	CN (509) to CH4 (500)	CN (510) to CH4 (500)	Sleep period of 5 time slots					

Table 6.12. Actual time slots of Data cycle phase of second Master Cycle in simulation of DCB-TDMA MAC protocol. (Values of S.T. and R.T. are in seconds.)

of data packet. Then, these nodes transmit the data in their respective time slot of the data cycle.

Data cycle phase has been illustrated in Table 6.12. From the table, it can be seen that, CN(201) has sent a data packet to CH1(200) in time slot 20 at time 965 s. which has been received by CH1 at 973.8 s. In the same time slot, CN(301) has sent data to CH2(300), CN(401) has sent data to CH3(400) and CN(501) has sent data to CH4(500). Using subsequent time slots, data has been collected from all the CN nodes (including new nodes) by respective CH nodes at every cluster. After collecting data from CN nodes, the data has been sent upwards by CH nodes to BS using reverse backbone link (CH4-CH3-CH2-CH1-BS). In this way, in time slot 38, BS has the data collected from all the CN nodes of the network. BS then transmits the data to the on-shore control station using an RF link.

Parallel intra-cluster communication has been possible in the network because of power level management by the CH nodes. In the simulation, using power levels $P_H = 0.01$ W and $P_L = 10^{-6}$ W, has given SNR as shown in Table 6.13.

Transmission Power P_t	SL(dB)	RL(dB)	SNR(dB)
$P_H = 0.01$ W (Range = 500 m)	150.77	100.5	21.91
$P_L = 10^{-6}$ W (Range = 10 m)	110.77	93.20	14.6

Table 6.13. Results of power levels and SNR from the simulation of DCB-TDMA MAC protocol.

Similarly, the power level of interfering signal obtained from the simulation is zero, resulting in the SINR value as 14.6 dB. Results obtained on simulation tool are slightly higher than the values obtained theoretically in Table 6.10. The reason for this needs to be investigated by studying the implementation of channel model of the simulator in detail.

In Section 6.5.4, details of total number of packets sent and received by each node in the network have been provided along with delay and energy calculations.

6.5.4 Number of packets transmitted and received

Based on the simulation results, it has been observed that the number of packets sent and received by various nodes are as follows:

In Master cycle 1,

No. of packets sent = no. of packets received by BS(100) = 1

No. of packets sent = no. of packets received by CH1(200) = 12

No. of packets sent = no. of packets received by CH2(300) = 12

No. of packets sent = no. of packets received by CH3(400) = 12

No. of packets sent = no. of packets received by CH4(500) = 11

No. of packets sent = no. of packets received by all CN nodes = 1

In Master cycle 2,

No. of packets sent = no. of packets received by BS(100) = 1

No. of packets sent = no. of packets received by CH1(200) = 13

No. of packets sent = no. of packets received by CH2(300) = 13

No. of packets sent = no. of packets received by CH3(400) = 12

No. of packets sent = no. of packets received by CH4(500) = 11

No. of packets sent = no. of packets received by all CN nodes = 1

In Master cycle 3,

No. of packets sent = no. of packets received by BS(100) = 1

No. of packets sent = no. of packets received by CH1(200) = 13

No. of packets sent = no. of packets received by CH2(300) = 12

No. of packets sent = no. of packets received by CH3(400) = 11

No. of packets sent = no. of packets received by CH4(500) = 11

No. of packets sent = no. of packets received by all CN nodes = 1

In Master cycle 4,

No. of packets sent = no. of packets received by BS(100) = 1

No. of packets sent = no. of packets received by CH1(201) = 13

No. of packets sent = no. of packets received by CH2(301) = 12

No. of packets sent = no. of packets received by CH3(401) = 11

No. of packets sent = no. of packets received by CH4(501) = 11

No. of packets sent = no. of packets received by all CN nodes = 1

Here, the example of Master cycle 2 can be taken to analyze the above results of simulation. In cycle 2, there have been 11 CN nodes in cluster 1 and 2 and 10 CN nodes each in cluster 3 and 4. Hence, in Control cycle phase of Master cycle 2, BS node has transmitted one packet, CH1 and CH2 nodes have transmitted 12 packets each, CH3 has transmitted 11 packets and CH4 has transmitted 10 packets. All CH and CN nodes have received one packet each.

In the Data cycle phase of second Master cycle, CH1 and CH2 have received 11 packets each, since there are 11 CN nodes in cluster 1 and 2. CH3 and CH4 have received 10 data packets each from 10 CN nodes. After receiving data packets on horizontal plane, data has been transmitted on vertical backbone link of network. This transfer has been initiated by CH4 node by transmitting data to CH3 node. CH3 has then forwarded the data to CH2 after aggregating the information with

the data received at its own cluster. Similarly, further, data has been transmitted upwards by CH2 to CH1 and from CH1 to BS. In this way, nodes BS, CH1, CH2, CH3 have received one packet each on vertical link. All CN and CH nodes have transmitted one data packet each in Data cycle phase.

Overall, the number of packets transmitted and received by every node in the network in Master cycle 2 matches correctly with the results obtained in the simulation. Similarly, number of packets transmitted and received by each node in the network in different Master Cycles can be analyzed easily. Based on these results, the calculation of overall delay in the link as well as energy consumption of nodes have been provided in the next section.

6.5.5 Calculations of delay and energy

Packet Delivery Time (\hat{D}) of a packet can be calculated from time-stamps R.T. and S.T. of Table 6.11 and Table 6.12. For example, using values from time slot 1 of Control cycle phase, \hat{D} for packet delivery over vertical link can be obtained as 9.13 s. Similarly, \hat{D} for packet delivery on horizontal link can be obtained using time slot 5 of Control cycle phase as 8.8 s. Similar results can be obtained using other time stamp values as well. In the simulation, size of data and control packet is 110 bytes including the header size. Data rate is 100 bps. As given by (4.1), (4.2) and (4.3), \hat{D} can be obtained as 9.13 s over a vertical link of 500 m, and 8.8 s over a horizontal link of 10 m.

Energy consumption of any node in the network can be obtained by calculating the energy consumption in transmission and reception by a node. For CH node, energy consumption in transmission on vertical link and horizontal link can be calculated separately using P_H and P_L values. Energy consumed in transmission using transmission power level P_H denoted by $E_{TX(PH)}$ is given by,

$$E_{TX(PH)} = \hat{T} \times P_H \quad (6.10)$$

Energy consumed in transmission using transmission power level P_L denoted as $E_{TX(PL)}$ is given as follows:

$$E_{TX(PL)} = \hat{T} \times P_L \quad (6.11)$$

All CN nodes use power level P_L for transmission of data packets. Hence, energy consumed by CN nodes can also be calculated using (6.11). Similarly, energy consumed in reception (denoted by E_{RX}) by any node in the network is given by,

$$E_{RX} = \hat{T} \times P_{RX} \quad (6.12)$$

wherein P_{RX} is constant term of reception and dependent on receiver. P_{RX} in simulation has been assumed as 10^{-5} W.

Energy consumption of nodes has been calculated using (6.10), (6.11) and (6.12). Information of number of packet transmitted and received by BS(100), CH1(200), CH2(300), CH3(400), CH4(500) in cycle 1 has been tabulated in Table 6.14. Also energy consumed in transmission and reception (in millijoules) by these nodes in cycle 1 has been presented in this table. Number of packets transmitted, and received by CN(201), CN(301), CN(401) and CN(501) in cycle 1 has been tabulated in Table 6.15 along with calculations of energy consumption. Since the functioning of all CN nodes remain same, one representative node has been used from each cluster. Similarly, the information of cycle 2 and cycle 3 have been tabulated for CH nodes in Table 6.16 and Table 6.17 respectively.

	BS(100)	CH1(200)	CH2(300)	CH3(400)	CH4(500)
Number of packets transmitted	1	12	12	12	11
Number of packets received	1	12	12	12	11
Energy consumed in transmission (mJ)	88	176.088	176.088	176.088	88.088
Energy consumed in reception (mJ)	0.088	1.056	1.056	1.056	0.968
Total energy consumed (mJ)	88.088	177.144	177.144	177.144	89.056

Table 6.14. Energy consumption by BS and CH nodes in cycle 1 in DCB-TDMA MAC protocol.

From Tables 6.14, Table 6.15, Table 6.16 and Table 6.17, it can be observed that energy consumed by BS in one cycle is 88.088 mJ since it transmits and receives one packet only during each cycle over vertical link of 500 m distance using power level P_H . Similarly, all the CN nodes transmit and receive only one packet in one cycle on a horizontal link of 10 m using power level P_L , consuming only 0.097 mJ energy. But CH nodes consume more energy since they need to communicate with

	CN(201)	CN(301)	CN(401)	CN(501)
Number of packets transmitted	1	1	1	1
Number of packets received	1	1	1	1
Energy consumed in transmission (mJ)	0.009	0.009	0.009	0.009
Energy consumed in reception (mJ)	0.088	0.088	0.088	0.088
Total energy consumed (mJ)	0.097	0.097	0.097	0.097

Table 6.15. Energy consumption by CN nodes in cycle 1.

	BS(100)	CH1(200)	CH2(300)	CH3(400)	CH4(500)
Number of packets transmitted	1	13	13	12	11
Number of packets received	1	13	13	12	11
Energy consumed in transmission (mJ)	88	176.097	176.097	176.088	88.088
Energy consumed in reception (mJ)	0.088	1.144	1.144	1.056	0.968
Total energy consumed (mJ)	88.088	177.241	177.241	177.144	89.056

Table 6.16. Energy consumption by BS and CH nodes in cycle 2 in DCB-TDMA MAC protocol.

	BS(100)	CH1(200)	CH2(300)	CH3(400)	CH4(500)
Number of packets transmitted	1	13	12	11	11
Number of packets received	1	13	12	11	11
Energy consumed in transmission (mJ)	88	176.097	176.088	176.079	88.088
Energy consumed in reception (mJ)	0.088	1.144	1.056	0.968	0.968
Total energy consumed (mJ)	88.088	177.241	177.144	177.047	89.056

Table 6.17. Energy consumption by BS and CH nodes in cycle 3 in DCB-TDMA MAC protocol.

all CN nodes using power level P_L as well as with adjacent CH nodes using power level P_H . Hence, a CH node would be depleted of battery power earlier than the other nodes in the network. For this reason, CH nodes need to be rotated based on their energy consumption. In the protocol, based on the energy threshold and energy consumption of the CH nodes, BS selects one of the

CN node (having highest residual energy and according to sequential numbering) as CH node and communicate this decision in the subsequent Control cycle. The new CH node is appointed in subsequent cycle and takes control of its cluster. Rotation of a CH is achieved using following major steps:

1. **Exchange of time slots of old and new CH nodes** -As the new CH has to become integral part of the backbone link (i.e. BS-CH1-CH2-CH3-CH4), it has to be given the requisite time slots for transmission and reception of control as well as data packets. Old CH will function as a CN node and so will have one time slot for receiving control packet from new CH and one time slot for transmitting data packet to new CH.
2. **Utilizing hybrid power level** - New CH has to communicate on vertical as well as horizontal links. It needs to adopt power levels P_L and P_H accordingly.
3. **Re-routing** - All information (whether control packets or data packets) has to be re-routed via new CH node.

For the illustration purpose, the energy information have been collected after 3 cycles in the simulation and have considered the threshold range of 500 mJ of energy consumption to initiate the CH rotation. In Table 6.18, overall energy consumption of BS and CH nodes as well as representative CN nodes from every cluster have been tabulated. Values of energy consumption of a node after cycle 1, cycle 2 and cycle 3 have been tabulated in column 2, 3 and 4 respectively. These values has also been plotted in Figure 6.3 for illustration.

Nodes	Cycle 1 (mJ)	Cycle 2 (mJ)	Cycle 3 (mJ)
BS(100)	88.088	176.176	264.264
CH1(200)	174.144	351.385	528.626
CH2(300)	174.144	351.385	528.529
CH3(400)	174.144	351.288	528.335
CH4(500)	89.056	178.112	267.168
CN(201)	0.097	0.194	0.291
CN(211)	0.0	0.097	0.194
CN(301)	0.097	0.194	0.291
CN(401)	0.097	0.194	0.291
CN(501)	0.097	0.194	0.291

Table 6.18. Overall energy consumption of nodes after cycle 1, 2 and 3 in DCB-TDMA MAC protocol.

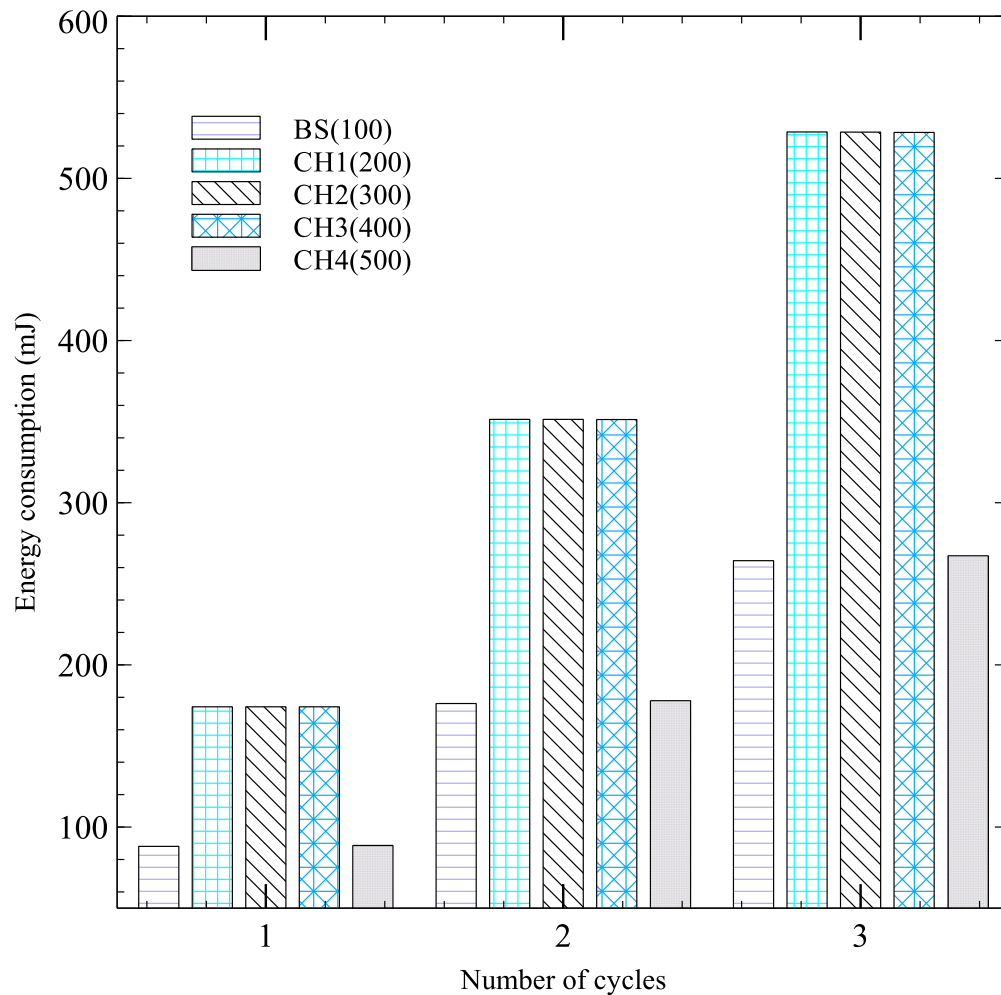


Figure 6.3. Overall energy consumption of nodes after cycle 1, 2 and 3 in DCB-TDMA MAC protocol.

From the results, it can be observed that, CH rotation would be required in clusters 1, 2 and 3 in subsequent (fourth) cycle. Based on the energy consumption of CN nodes, it can be noted that the nodes 211, 301, and 401 would be new CH nodes in these clusters. Accordingly, the BS has appointed these new CH nodes in the fourth cycle of simulation. Time slots of these nodes have been exchanged with old CH nodes as illustrated in Table 6.19 and Table 6.20. Table 6.19 shows the Control cycle phase of fourth Master cycle and Table 6.20 shows Data cycle phase of fourth Master cycle. It can be observed that the control and data information has been routed via new CH nodes.

In Table 6.21, the number of packets transmitted and received by the BS and new CH nodes in the subsequent (fourth) cycle have been tabulated along with the energy consumption in mJ. It is to be noted that here, cluster 1 has 11 nodes, cluster 2 and 4 has 10 nodes each, while cluster 3

Time slot 1	Time slot 2	Time slot 3	Time slot 4	Time slot 5	Time slot 6	Time slot 7	Time slot 13	Time slot 14	Five time slot period (time slots 15 to 19)	
BS (100) to CH1 (211)	CH1 (211) to CH2 (301)	CH2 (301) to CH3 (401)	CH3 (401) to CH4 (500)	CH1 (211) to CN (201)	CH1 (211) to CN (202)	CH1 (211) to CN (203)	CH1 (211) to CN (209)	CH1 (211) to CN (210)	CH1 (211) to CN (200)	Sleep period of 4 time slots
				CH2 (301) to CN (300)	CH2 (301) to CN (302)	CH2 (301) to CN (303)	CH2 (301) to CN (309)	CH2 (301) to CN (310)		Sleep period of 5 time slots
				CH3 (401) to CN (400)	CH3 (401) to CN (402)	CH3 (401) to CN (403)	CH3 (401) to CN (409)			Sleep period of 6 time slots
				CH4 (500) to CN (501)	CH4 (500) to CN (502)	CH4 (500) to CN (503)	CH4 (500) to CN (509)	CH4 (500) to CN (510)		Sleep period of 5 time slots

Table 6.19. Control cycle phase of fourth Master cycle in simulation illustrating CH rotation.

has only 9 nodes. Table 6.22 shows the overall (cumulative) energy consumption of BS, new CH nodes and old CH nodes at the end of fourth cycle. It shows that the CH rotation is very effective in balancing the energy consumption among all the nodes in the network.

In the implementation of the DCB-TDMA MAC protocol on SUNSET simulation platform, Control cycle phase has only unidirectional communication links. Exchange of messages can be incorporated by allowing a bidirectional communication between two nodes during Control Cycle phase. For message exchanges, more than one time slot will be required for a pair of node. One of the essential network management protocol which can be implemented using this concept is time synchronization protocol. This protocol is also important for the proper functioning of TDMA based MAC protocol. In this work, DCB-TDMA MAC protocol have been implemented along with multi-hop Tri-message time synchronization on UnetStack simulation tool [175, 176] as well as on hardware set-up. Details of these implementations have been presented in Chapter 7.

Time slot 20	Time slot 21	Time slot 22	Time slot 28	Time slot 29	Five time slot period (time slots 30 to 34)		Time slot 35	Time slot 36	Time slot 37	Time slot 38
CN (201) to CH1 (211)	CN (202) to CH1 (211)	CN (203) to CH1 (211)	CN (209) to CH1 (211)	CN (210) to CH1 (211)	CN (200) to CH1 (211)	Sleep period of 4 time slots	CH4 (500) to CH3 (401)	CH3 (401) to CH2 (301)	CH2 (301) to CH1 (211)	CH1 (211) to BS (100)
CN (300) to CH2 (301)	CN (302) to CH2 (301)	CN (303) to CH2 (301)	CN (309) to CH2 (301)	CN (310) to CH2 (301)	Sleep period of 5 time slots					
CN (400) to CH3 (401)	CN (402) to CH3 (401)	CN (403) to CH3 (401)	CN (409) to CH3 (401)	Sleep period of 6 time slots						
CN (501) to CH4 (500)	CN (502) to CH4 (500)	CN (503) to CH4 (500)	CN (509) to CH4 (500)	CN (510) to CH4 (500)	Sleep period of 5 time slots					

Table 6.20. Data cycle phase of fourth Master cycle in simulation illustrating CH rotation.

	BS(100)	CH1(211)	CH2(301)	CH3(401)	CH4(500)
Number of packets transmitted	1	13	12	11	11
Number of packets received	1	13	12	11	11
Energy consumed in transmission (mJ)	88	176.097	176.088	176.079	88.088
Energy consumed in reception (mJ)	0.088	1.144	1.056	0.968	0.968
Total energy consumed (mJ)	88.088	177.241	177.144	177.047	89.056

Table 6.21. Energy consumption by BS and new CH nodes in cycle 4 in DCB-TDMA MAC protocol.

Performance comparison of DCB-TDMA MAC protocol has been done with the Basic TDMA MAC protocol for the same three-dimensional network architecture. It is assumed that for the Basic TDMA MAC protocol also, the network is arranged in the similar structure with clusters and Cluster-Head nodes. Master cycle consists of Control cycle phase and Data cycle phase. Initially

Node	Energy consumed in 4 cycles (mJ)
BS(100)	352.352
CH1(211)	177.435
CH2(301)	177.435
CH3(401)	177.338
CH4(500)	356.224
CN(200)	528.723
CN(300)	528.626
CN(400)	528.432

Table 6.22. Energy consumption of BS, new CH nodes and old CH nodes at the end of fourth cycle in DCB-TDMA MAC protocol.

every cluster has 11 nodes including one CH node. Five extra time slots have been provided for adding new nodes as part of future expansion. Node addition and deletion in cycle 2 and cycle 3 has been done in the same manner of DCB-TDMA MAC protocol. Novel features of DCB-TDMA MAC such as power level adaptation by CH node, and CH rotation is not available. Since the power level adaptation is not available, the transmission of information happens sequentially. For example, in control cycle, information is transmitted on backbone link (BS-CH1-CH2-CH3-CH4), and then one-by-one to the CN nodes starting from 201 to 510. Performance evaluation of Basic TDMA MAC protocol in comparison to proposed protocol has been provided in Section 6.6.

6.6 Comparison of DCB-TDMA MAC protocol with Basic TDMA MAC protocol

Comparative performance of proposed protocol with Basic TDMA based protocol can be observed in terms of various network parameters such as end-to-end delay, throughput, channel utilization and energy consumption. These parameters have been calculated in the following subsections.

6.6.1 End-to-end delay

Minimum value of end-to-end delay for the Basic TDMA MAC protocol occurs for the transmission from BS to CN(201), whereas maximum delay value occurs for the BS to CN(510)

transmission on the forward link. Functioning of Basic TDMA MAC is symmetrical in Control cycle and Data cycle phases, hence delay values on these links can be obtained using following equations:

$$\text{Minimum Delay}_{FL} = (L_T - 1) \times T_S + T_S \quad (6.13)$$

$$\text{Minimum Delay}_{RL} = (L_T - 1) \times T_S + T_S \quad (6.14)$$

$$\text{Maximum Delay}_{FL} = [T_{MAX} - (L_T - 1) + 1] \times (L_T - 1) \times T_S \quad (6.15)$$

$$\text{Maximum Delay}_{RL} = [T_{MAX} - (L_T - 1) + 1] \times (L_T - 1) \times T_S \quad (6.16)$$

From (6.13), (6.14), (6.15), and (6.16), it can be observed that, in case of basic TDMA MAC,

$$\text{Minimum Delay}_{FL} = \text{Minimum Delay}_{RL} \quad (6.17)$$

$$\text{Maximum Delay}_{FL} = \text{Maximum Delay}_{RL} \quad (6.18)$$

Considering $MP = 100$ bytes, $MH = 10$ bytes, $GT = 0.5$ s, $FEC_{CR} = 0.96$, $\hat{P} = 0.3333$ s, and $L_T = 5$, parameters of TDMA MAC protocol such as time slot duration T_S , minimum and maximum delay values have been calculated for various R values. Average delay has been calculated from these minimum and maximum values of delay. Results have been tabulated in Table 6.23. End-to-end delay of Basic TDMA MAC has been plotted along with delay values of DCB-TDMA MAC protocol (from Table 6.6) in Figure 6.4. It can be observed that delay in Basic TDMA MAC protocol are significantly higher than the delay in DCB-TDMA MAC protocol. In Basic TDMA MAC, since the adaptive power levels are not available, transmission of information happens sequentially from first to last node of the network. Depending on the number of nodes and number of slots reserved for future use, the delay increases in the Basic TDMA MAC protocol.

Bit Rate R (bps)	Time slot T_S (s)	Min. Delay _{FL} = Min. Delay _{RL} (s)	Max. Delay _{FL} = Max. Delay _{RL} (s)	Average Delay (s)
100	10	50	640	345
500	2.66	13.3	170.24	91.77
1000	1.75	8.75	112	60.38
1500	1.44	7.2	92.16	49.68
2000	1.29	6.45	82.56	44.51

Table 6.23. Average end-to-end delay in Basic TDMA MAC protocol for various bit rates.

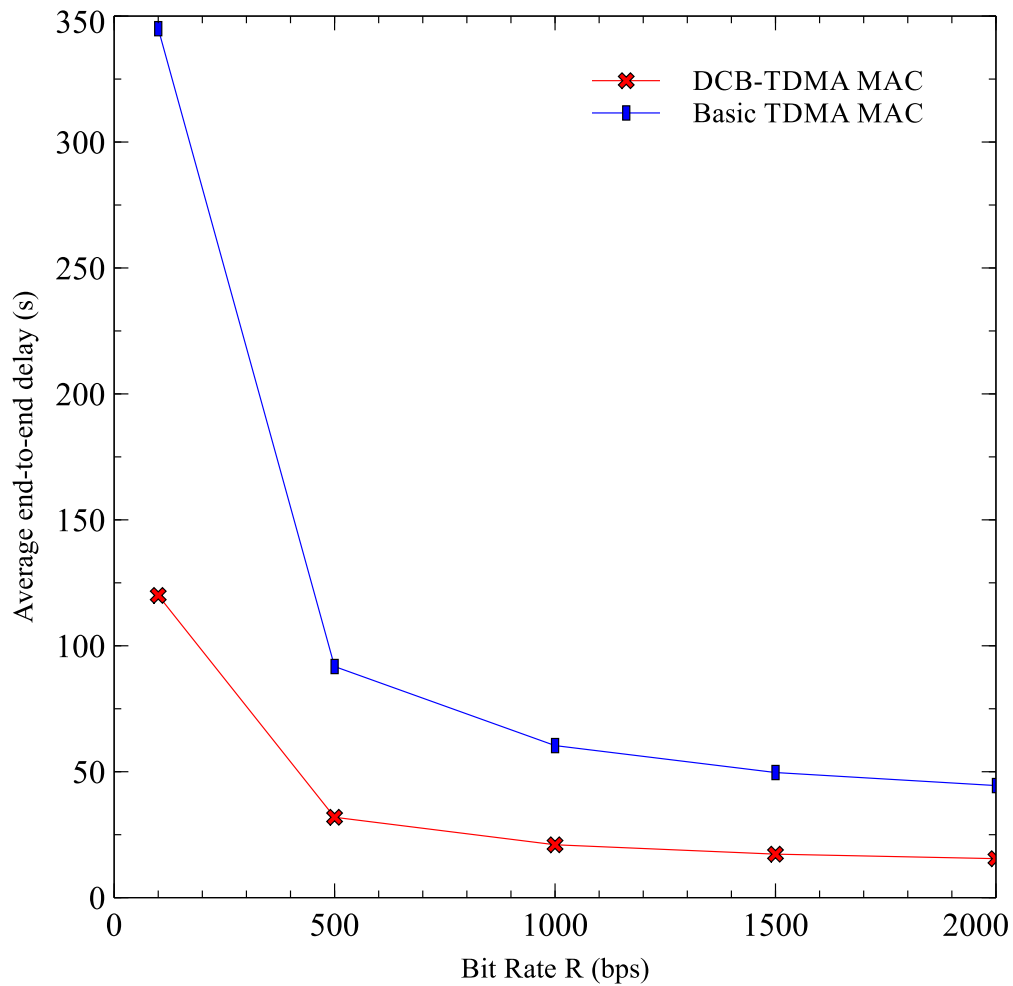


Figure 6.4. Comparison of average end-to-end delay in DCB-TDMA and Basic TDMA MAC protocols.

6.6.2 Throughput

Throughput has been calculated using Equation (6.7) and average delay values from Table 6.23. Throughput for various R values have been tabulated in Table 6.24. Throughput of Basic TDMA MAC has been plotted along with throughput of DCB-TDMA MAC protocol (from Table 6.7) for various R values in Figure 6.5. It can be observed that, since the delay in Basic TDMA MAC is higher than in DCB-TDMA MAC, the throughput in Basic TDMA MAC is lower as compared to DCB-TDMA MAC protocol.

Sr. No.	Bit Rate R (bps)	Throughput (bps)
1	100	2.32
2	500	8.72
3	1000	13.25
4	1500	16.10
5	2000	17.97

Table 6.24. Throughput in Basic TDMA MAC protocol for various bit rates.

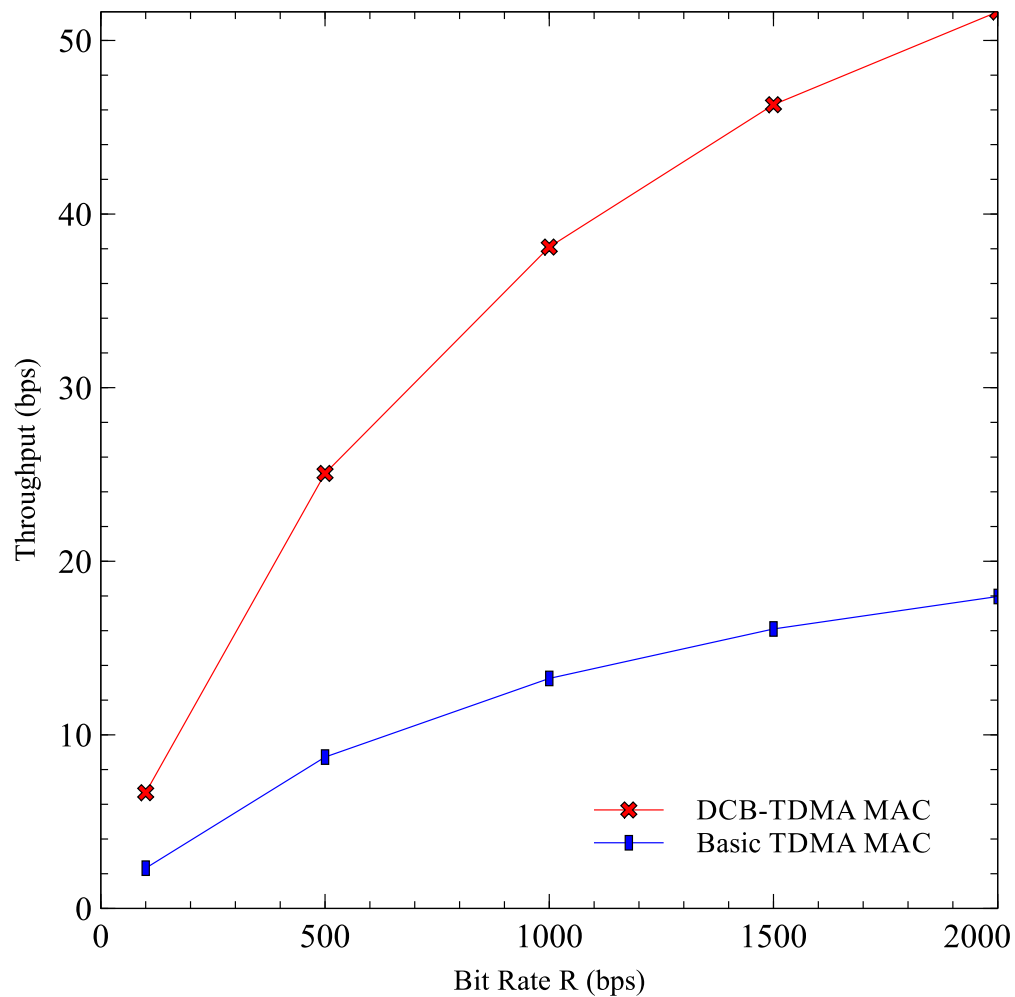


Figure 6.5. Comparison of throughput in DCB-TDMA and Basic TDMA MAC protocols.

6.6.3 Channel utilization

Channel utilization has been calculated for different R values using Equation (6.8) and time slot T_S values from Table 6.23. Channel reuse factor for Basic TDMA MAC protocol is 1. Results of channel utilization have been illustrated in Table 6.25. For comparison, channel utilization of DCB-TDMA MAC and Basic TDMA MAC are plotted in Figure 6.6 for various R values. Results

indicate that the channel utilization in Basic TDMA MAC protocol is smaller by the factor of 4 as compared to DCB-TDMA MAC protocol.

Sr. No.	Bit Rate R (bps)	Channel Utilization (%)
1	100	4.21
2	500	3.17
3	1000	2.41
4	1500	1.95
5	2000	1.63

Table 6.25. Channel utilization in Basic TDMA MAC protocol for various bit rates.

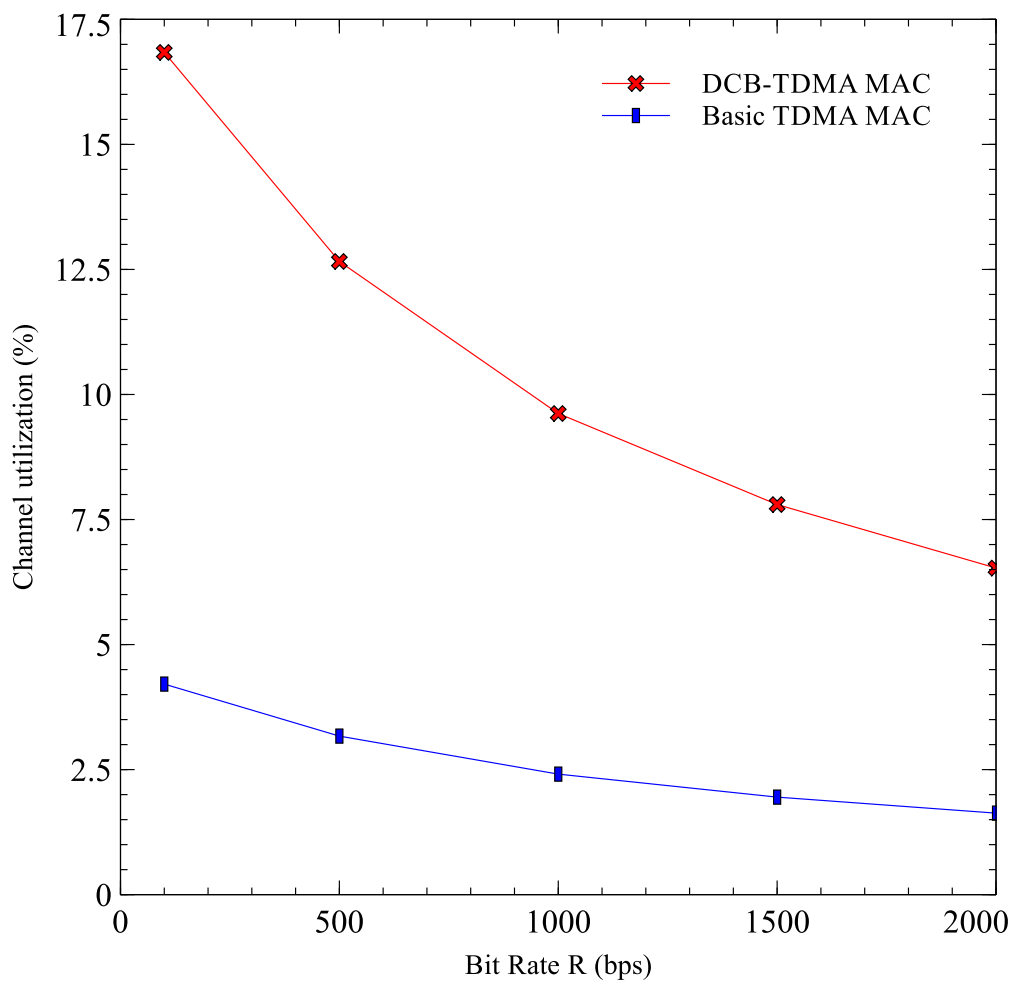


Figure 6.6. Comparison of channel utilization in DCB-TDMA and Basic TDMA MAC protocols.

6.6.4 Energy consumption

Number of packets sent and received by network nodes in the Basic TDMA MAC protocol in cycle 1, 2 and 3 is same as the numbers presented in Subsection 6.5.4. Without the power level

adaptation at CH nodes, it uses same power level P_H for communication at vertical as well as horizontal levels. Energy consumed in transmission and reception of packets by any node is given by 6.10 and 6.12 respectively. Overall energy consumed by BS, CH nodes and representative CN nodes after cycle 1, 2 and 3 have been tabulated in Table 6.26. Results have also been plotted in Figure 6.7. It can be easily observed that the energy consumed in Basic TDMA MAC is significantly higher than the energy consumption in DCB-TDMA MAC protocol.

Nodes	Cycle 1 (mJ)	Cycle 2 (mJ)	Cycle 3 (mJ)
BS(100)	88.088	176.176	264.264
CH1(200)	1057.056	2202.200	3347.344
CH2(300)	1057.056	2202.200	3259.256
CH3(400)	1057.056	2114.112	3083.080
CH4(500)	968.968	1937.936	2906.904
CN(201)	88.088	176.176	264.264
CN(211)	0.0	88.088	176.176
CN(301)	88.088	176.176	264.264
CN(401)	88.088	176.176	264.264
CN(501)	88.088	176.176	264.264

Table 6.26. Overall energy consumption of nodes after cycle 1, 2 and 3 in Basic TDMA MAC protocol.

From the above comparison, it is evident that the effective power level management has improved network performance of DCB-TDMA MAC protocol in terms of end-to-end delay, throughput, channel utilization and energy consumption. Moreover, since the power levels in DCB-TDMA MAC have been calculated based on distance of communication, the energy consumed in transmission of information is optimum. Further, CH rotation scheme based on energy consumption balances energy consumption among various nodes in the cluster. Hence, the DCB-TDMA MAC protocol has fairness not just in terms of data transmission of node, but also in energy consumption. This policy of CH rotation and utilization of sleep-wake cycle makes the DCB-TDMA MAC protocol as energy-efficient protocol, helping in prolonging the life-time of network.

In the Section 6.7, summary of this chapter has been provided.

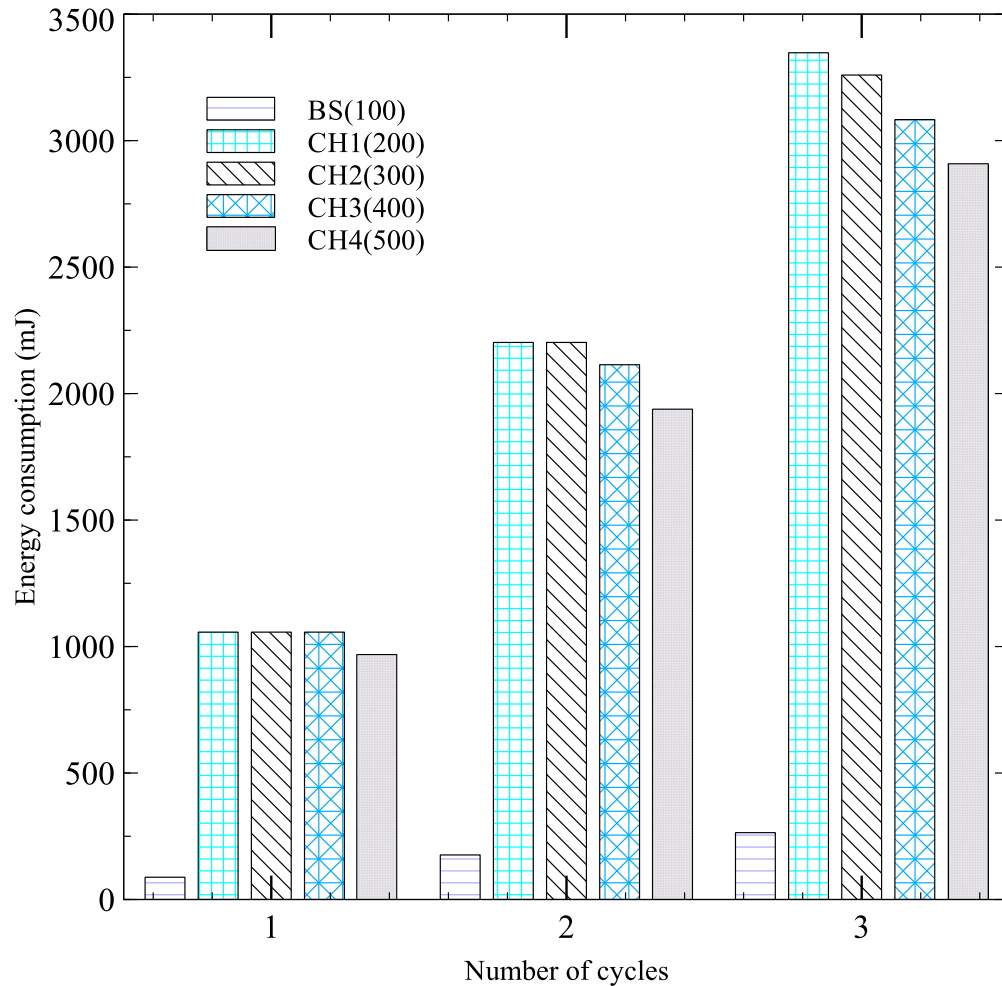


Figure 6.7. Overall energy consumption of nodes after cycle 1, 2 and 3 in Basic TDMA MAC protocol.

6.7 Summary

This chapter has provided the detail description, implementation details, simulation results and analysis of proposed DCB-TDMA MAC protocol. Novel features of this protocol has been summarized as follows:

1. **Energy awareness:-** The proposed protocol is energy aware network protocol, since it has implemented following schemes:
 - (a) This protocol has sleep-wake duration of 50% for energy conservation. The sleep duration can be tuned based on requirement of application.
 - (b) Time slots for every node has been pre-scheduled. Hence, a node can sleep during the remaining period of cycle, conserving energy.

- (c) Energy consumption has been balanced between network nodes with the help of CH rotation, which results in increasing the life-span of network.
 - (d) Power levels used by nodes have been decided based on distance of transmission, resulting in optimum use of transmission power.
2. **Cross-layer implementation:-** The proposed protocol is inherently cross-layer in nature, since it has features such as:
- (a) Cluster-Head rotation based on energy consumption.
 - (b) Power level management of transmission power.
 - (c) Re-route formation dynamically in case of CH rotation.
3. **Scalability:-** The proposed protocol addresses scalability issue of TDMA MAC protocol by using reservation of time slots for future expansion of network.
4. **Improving channel utilization:-** Effective channel utilization of bandwidth constrained underwater acoustic channel is very important. This protocol improves channel utilization with the help of dynamic adaptation of power levels by CH nodes.

Chapter 7

Cross-layer Protocol Stack

Development for Three-Dimensional UASN

7.1 Introduction

Among the research community of wireless networks, it is accepted opinion that the network performance can be improved using cross-layer design approach. As stated in Section 2.2, lack of information sharing across protocol layers is the major shortcoming of traditional layered design approach. Because of this reason, wireless networks operate in suboptimal mode when a traditional design approach is used. In this chapter, a cross-layer implementation of protocol stack has been presented for three-dimensional UASN, involving a joint design of network functionalities across various layers and network management planes as well. Improving energy efficiency of network, and thus prolonging the life-time is the main focus of this cross-layer design.

The organization of Chapter is as follows: In Section 7.2, the proposed protocol stack has been described for three-dimensional UASN. This description has been provided layer-wise for the sake of simplicity. Complete protocol stack has been implemented on the UnetSim simulator. UnetSim simulator uses the UnetStack implementation to simulate an underwater network on a

single computer [233]. Details of simulation parameters, results and analysis of results have been provided in Section 7.3. Implementation of the proposed protocol stack on hardware testbed set-up along with results and analysis have been presented in Section 7.4. A brief summary of Chapter has been provided in Section 7.5.

7.2 Description of proposed protocol stack for three-dimensional UASN

In this section, cross-layer implementation of protocol stack for three-dimensional UASN has been discussed in detail. This description has been provided layer-wise as follows:

7.2.1 Network architecture of UASN

The network architecture as described in Section 3.2 has been considered for long-term ocean column monitoring application. This three-dimensional network architecture is an example of structured and static deployment with a multi-tiered, multi-hop communication links. It has been assumed that the network is sparsely deployed and is homogeneous in nature. As shown in Figure 3.1, the network consists of clusters deployed at multiple levels from top to bottom.

7.2.2 Physical layer

A channel model for deep water as described in Section 3.4 has been considered for this three-dimensional network. Various parameters of channel such as TL, NL, Propagation speed has been calculated using this model. Based on these calculations, the optimum frequency for operation, modulation technique, data rates and power levels of transmission and reception have been determined as per description in Section 3.5.

7.2.3 Data Link Layer

At DLL, DCB-TDMA MAC as proposed in Chapter 6 has been implemented on the network. The network repeats the Master cycle, consisting of sleep phase and wake phase. A wake phase further consists of Control cycle and Data cycle phase. In the Control cycle phase of DCB-TDMA MAC protocol, implementation of multi-hop Tri-message time synchronization as described in Section 5.4 has been provided. Every node has certain number of time slots allotted for the transmission and reception of information. Though a data aggregation scheme has not been implemented as of now, a larger time slot can be provided to encompass a MAC layer payload.

7.2.4 Network layer

In the proposed three-dimensional network, a forward link from BS to CN nodes has been provided for transfer of control information. Similarly, for the collection of data at the BS, a reverse link from CN nodes to BS has been made available. All these links have been routed through a corresponding CH nodes of a cluster. Similarly, the routing of information from the top to the bottom of network has been done using the backbone link. These routes have been formed by using initial deployment of nodes and appointment of CH nodes. When the CH nodes are rotated, the route has been recalculated dynamically via the newly selected CH nodes.

7.2.5 Transport layer

Implementation of scheduled based MAC protocol guarantees near-zero probability of congestion and collisions in the network. End-to-end reliability is ensured on both forward and reverse links. Additionally, an FEC based schemes can be implemented while accommodating data in the payload for transmission. Retransmissions has not been scheduled on the network, since the network has sufficient spatial redundancy as well as repetitive Master cycles for information transmission.

7.2.6 Application layer

The collection of data from ocean column in regular intervals has been the intended application of this network. This has been ensured by a Data cycle phase repeating in every Master cycle. A query based data collection can also be done using feature described in Subsection 6.3.1.

7.2.7 Management planes of UASN

Various protocols of network management planes such as time synchronization, clustering, and power level management have been incorporated in the proposed protocol stack. It has been assumed that the location information of nodes is known a-priori. The information of location, level of deployment and time slots has been pre-programmed into the node before deployment in the network. This information has also been made available to the BS. When a new node is added, the information in BS has been updated accordingly.

7.2.7.1 Clustering

As described in Section 3.2, the network consists of multiple levels such as Level 1, Level 2, and so on, deployed at varying depths from sea-surface. At sea-surface, only a single node (BS) has been deployed. At every level below the sea-surface, a cluster of nodes has been deployed in horizontal plane. Every cluster includes a CH node. Rotation of CH has been performed based on energy consumption of nodes.

7.2.7.2 Time synchronization

In this network, a proposed multi-hop Tri-message time synchronization protocol as described in Subsection 5.4 has been implemented. It has been demonstrated in Chapter 5 that this protocol consumes the least amount of energy as compared to other time synchronization protocols while providing comparable accuracy. Hence, this protocol has been chosen for implementation on the network.

7.2.7.3 Power level management

CH node uses optimal power levels for communication over vertical and horizontal links. It has been assumed that CH nodes can dynamically change the power levels while switching from communication over vertical link to horizontal link. These power levels at physical layer of network protocol stack ensures the optimal utilization of transmission powers by a node. At MAC layer, the nodes have been scheduled to sleep mode for the entire duration other than their assigned time slots for transmission and reception. The network follows a sleep-wake pattern with a duty cycle of 50%. CH nodes have been rotated based on energy information, in order to have balance of energy consumption among network nodes. In totality, there has been a strong power level management at various layers to improve energy efficiency of network.

This proposed protocol stack has been implemented on the UnetSim simulator introduced in Subsection 2.3.1.5. Details of implementation, results and analysis of results have been provided in the Section 7.3.

7.3 Implementation of proposed cross-layer protocol stack of UASN on UnetSim simulator

UnetSim simulator has been used for the implementation of proposed cross-layer protocol stack. UnetSim simulator is developed by Acoustic research laboratory of National University of Singapore under Unet Project. Main component of UnetSim is the agent-based network stack termed as UnetStack. Brief introduction of UnetStack has been provided in Subsection 7.3.1.

7.3.1 UnetStack

UnetSim uses UnetStack implementation, which is a collection of software agents providing well-defined services. It has a service oriented architecture which allows information to be shared, services to be provided, and behaviors to be negotiated between different agents. In UnetStack, user can describe a code in Domain Specific Language (DSL) developed using Groovy. This same code can be run in simulation and can also be deployed on underwater modems. The

functionality of porting the compiled binary of the same code on modems saves huge amount of time and effort. It also removes the uncertainties of any differences during porting phase. Once a protocol is developed and tested in simulation, it is ready to be deployed and tested at sea in any UnetStack-compatible modem. Currently, various modems such as ARL UNET-II modem [183], Subnero modem [234], Evologics WiSE-edition modem [235] have UnetStack compatibility. Numerous field experiments have been conducted using these modems in past 5-6 years.

The main feature of UnetStack is that, in the UnetStack, though the agents play the role that 'layers' play in traditional network stacks they are not organized in any enforced hierarchy and are free to interact in any way suitable to meet application needs. This promotes low-overhead protocols and cross-layer information sharing [175]. Agents are defined with their services, messages, capabilities and parameters. A list of services in UnetStack is available on [175]. Developers can implement additional services as well as extend UnetStack by developing new agents.

Figure 7.1 shows a UnetStack architecture. A complete underwater networking solution is provided using various agents shown in the figure. The stack runs on a Java virtual machine and the fjage open- source agent framework. Access to physical layer services of modem is provided using physical (driver) agent in the UnetStack.

Details of simulation parameters used for the implementation of proposed protocol stack at various layers have been discussed in following subsections.

7.3.2 Deployment of three-dimensional network

A three-dimensional network as proposed in Section 3.2 has been deployed using UnetSim. Terminologies and numbering of nodes, levels, and clusters has also been taken from Section 3.2. In this particular implementation, total number of levels (L_T) has been taken as 4. At sea-surface, BS node has been deployed. There are three clusters below sea-surface with a vertical distance of $d_u = 500$ m in-between them. Total depth of the deployed network is $D_u = 1500$ m. Nodes have been deployed in the form of cluster with a maximum distance of $R_u = 10$ m from the center of column. Initially, every cluster has four nodes, wherein one node is CH node and three nodes are

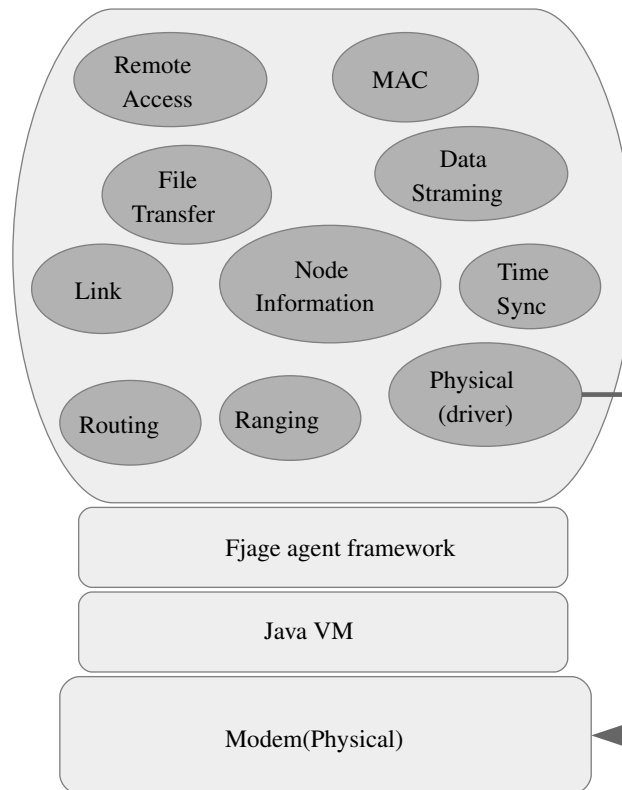


Figure 7.1. The UnetStack architecture [233].

CN nodes. There is provision of 6 more slots in every cluster for the future expansion of network. In cycle 2, three nodes have been added in every cluster. Similarly, in cycle 3, three more nodes have been added in every cluster. In cycle 5, CH rotation has been showcased based on energy consumption. The behavior of the cross-layered protocol stack has been tracked on the simulator for multiple cycles.

Locations of nodes in the simulation have been chosen randomly in the peripheral range of 10 m from center of column. Node 100 is the BS node located at $(x = 0, y = 0, \text{ and } z = 0)$. Certain randomization has been assumed in the location since it would be highly difficult to deploy and retain the node at the exact desired location in underwater network. Results obtained in one of the simulation run has been provided in the Table 7.1. From the table, it can be observed that, the nodes are not positioned in exact horizontal plane, but rather are located in the vertical range of 4 to 5 m from a given depth level.

Cluster 1		Cluster 2		Cluster 3	
Node	Location (x,y,z) (m)	Node	Location (x,y,z) (m)	Node	Location (x,y,z) (m)
200	(1.57, -1.18, 500.00)	300	(-7.52, -4.72, 997.00)	400	(4.53, -0.24, 1504.00)
201	(4.32, 3.50, 503.00)	301	(-1.01, 4.20, 1002.20)	401	(-3.92, 8.35, 1502.02)
202	(-1.49, 9.36, 500.50)	302	(-0.21, -8.80, 1003.50)	402	(-2.65, -5.28, 1498.00)
203	(-4.39, 7.36, 502.80)	303	(-4.07, 0.57, 999.80)	403	(-1.62, 0.91, 1499.50)
204	(7.95, -8.22, 501.60)	304	(3.91, 1.13, 998.50)	404	(-0.72, -7.02, 1501.50)
205	(-2.71, -2.45, 500)	305	(6.60, -6.40, 1001.00)	405	(-2.97, -8.99, 1500.50)
206	(3.06, -1.16, 498.50)	306	(8.17, -5.03, 1001.40)	406	(1.22, -5.88, 1498.00)
207	(1.20, -4.50, 499.00)	307	(-4.18, 3.59, 1002.10)	407	(-2.19, -7.83, 1502.00)
208	(8.38, 0.50, 504.20)	308	(-7.91, -8.89, 1003.00)	408	(6.31, -5.25, 1501.30)
209	(-6.33, -1.46, 502.10)	309	(2.06, -5.26, 1000.00)	409	(-9.36, 3.93, 1503.60)

Table 7.1. Locations of nodes in deployment of three-dimensional UASN.

7.3.3 Physical layer parameters

In the simulation set-up, basic acoustic model available on UnetSim [236] has been utilized. This model comprises of Urick acoustic model and communication model. For Urick acoustic model, absorption coefficient has been calculated using Schulkin and Marsh equation [211]. In the communication model, BPSK modulation scheme has been implemented for Rician or Rayleigh fading channel. The parameters as tabulated in Table 7.2 have been used in simulation.

Sr. No.	Parameter	Value
1	Frequency of operation (f)	25 kHz
2	Bandwidth (B)	4096 Hz
3	Bit Rate (R)	2400 bps
4	Temperature of sea-surface	25 °C
5	Salinity (S)	35 ppt
6	Spreading factor (k)	2
7	Noise Level (NL)	40 dB
8	Higher power level P_H	0.01 W
9	Lower power level P_L	10^{-6} W

Table 7.2. Parameters for simulation of proposed protocol stack on UnetSim.

TL for communication range can be calculated using (3.1). Absorption coefficient α_t can be obtained by Schulkin and Marsh equation [211] described as follows:

$$\alpha_t = \left(0.0186 \frac{S f_T f^2}{f_T^2 + f^2} + 0.0268 \frac{f^2}{f_T} \right) \times 1.0936 \quad (7.1)$$

wherein relaxation frequency f_T (in kHz) is given by,

$$f_T = 21.9 \times 10^{6 - 1520/T + 273} \quad (7.2)$$

Here, f is frequency of operation in kHz, T is the temperature in $^{\circ}C$ and S is salinity in ppt.

Using power levels $P_H = 0.01$ W and $P_L = 10^{-6}$ W, and using (3.1), (7.1), (3.26), and (3.27), SNR values can be calculated for vertical link of 500 m and horizontal link of 10 m. SNR values for the given scenario has been tabulated in Table 7.3.

Transmission Power P_t	SL(dB)	RL(dB)	SNR(dB)
$P_H = 0.01$ W (Range = 500 m)	150.77	95.49	19.37
$P_L = 10^{-6}$ W (Range = 10 m)	110.77	90.74	14.62

Table 7.3. Calculations of power levels and SNR values in proposed protocol stack.

SINR value on the horizontal link due to parallel intra-cluster communications happening in adjacent clusters can be obtained using (3.29). Value obtained in the given scenario is SINR = 14.55 dB. Further, using (3.28) and Table 3.2, it can be observed that the probability of bit error in BPSK modulation technique is negligibly smaller.

7.3.4 Description of proposed protocol stack in terms of Master cycles

In the proposed cross-layer protocol stack, DCB-TDMA MAC protocol has been implemented which has been described in Chapter 6.2. Structure of Master cycle is same as shown in Figure 6.1. In a Control cycle phase of DCB-TDMA MAC protocol, unidirectional transmission of control information has been replaced with the multi-hop Tri-message time synchronization protocol described in Section 5.4. To fetch the results of clock skew and offset, a fourth message has

been added in Tri-message time synchronization protocol, wherein a synchronized node sends the values of skew and offset to the synchronizing reference node in a packet payload. Hence, every slot of control cycle has been replaced by 4 time slots for each pair of nodes. Rest of the functionality of DCB-TDMA MAC protocol is similar to the description provided in Section 6.2.

In Table 7.4, an illustration of Control cycle phase of first Master cycle has been provided. As shown in the table, slot numbers 1 to 4 are used for synchronization of CH1(200) by the BS(100). Next four slots (5 to 8) are used by CH1(200) to synchronize CH2(300). Similarly slots 9 to 12 are used by CH2(300) to synchronize CH3(400). After the synchronization of CH nodes on backbone link, the synchronization of CN nodes is initiated by respective CH nodes of clusters. This communication at horizontal level is performed in parallel across various clusters. For example, from the table, it can be observed that the slots 13 to 16 are used for synchronization of CN(201), CN(301), CN(401) by CH1(200), CH2(300) and CH3(400) respectively. All CN nodes are synchronized by the end of Control cycle phase. For future expansion, 24 slots are kept reserved for possible addition of 6 nodes in the network. Data Cycle starts after these time slots.

Time slot 1 to 4	Time slot 5 to 8	Time slot 9 to 12	Time slot 13 to 16	Time slot 17 to 20	Time slot 21 to 24	24 time slot period (time slots 25 to 48)
BS (100) to CH1 (200)	CH1 (200) to CH2 (300)	CH2 (300) to CH3 (400)	CH1 (200) to CN (201)	CH1 (200) to CN (202)	CH1 (200) to CN (203)	Sleep period
			CH2 (300) to CN (301)	CH2 (300) to CN (302)	CH2 (300) to CN (303)	Sleep period
			CH3 (400) to CN (401)	CH3 (400) to CN (402)	CH3 (400) to CN (403)	Sleep period

Table 7.4. Illustration of Control cycle phase of first Master cycle for simulation in UnetSim simulator.

Data cycle phase for the first Master cycle is as shown in Table 7.5. As shown, the Data cycle starts with slot number 49. In this slot, CN(201), CN(301) and CN(401) transmit the data information to

their corresponding CH nodes CH1(200), CH2(300) and CH3(400) respectively. Time slot 50 is used by CN(202), CN(302) and CN(402) to transmit the data information to CH nodes CH1(200), CH2(300) and CH3(400), whereas in time slot 51, nodes CN(203), CN(303) and CN(403) transmit data to these CH nodes. Next 6 time slots are used for future expansion. After this time period, the data is transmitted upwards using reverse backbone link CH3-CH2-CH1-BS. Every hop takes one time slot each. Once data is collected at BS, it can transmit this data to the on-shore control station using RF link.

Time slot 49	Time slot 50	Time slot 51	Six time slot period (time slots 52 to 57)	Time slot 58	Time slot 59	Time slot 60
CN (201) to CH1 (200)	CN (202) to CH1 (200)	CN (203) to CH1 (200)	Sleep period	CH3 (400) to CH2 (300)	CH2 (300) to CH1 (200)	CH1 (200) to BS (100)
CN (301) to CH2 (300)	CN (302) to CH2 (300)	CN (303) to CH2 (300)	Sleep period			
CN (401) to CH3 (400)	CN (402) to CH3 (400)	CN (403) to CH3 (400)	Sleep period			

Table 7.5. Illustration of Data cycle phase of first Master cycle for simulation in UnetSim simulator.

7.3.4.1 Setting the duration of time slot

For network simulation, following parameters have been considered:

1. Packet Size (P_S) = 11 bytes.
2. Average propagation Speed of acoustic signal in water (c) has been calculated using Mackenzie formula (Section 3.4) for respective depth values.
3. Bit Rate (R) = 2400 bps.

Using (4.1), (4.2), and (4.3), parameters such as Packet Transmission Time (\hat{T}), Propagation Time (\hat{P}) and Packet Delivery Time \hat{D} can be obtained.

In this scenario,

$$\hat{T} = 0.037 \text{ s.}$$

$$\hat{P} \text{ (for horizontal link of 10m) is } 0.0066 \text{ s.}$$

$$\hat{P} \text{ (for vertical link of 500m) } = 0.3333 \text{ s.}$$

$$\hat{D} \text{ (Horizontal link) } = 0.043 \text{ s.}$$

$$\hat{D} \text{ (Vertical link) } = 0.3703 \text{ s.}$$

For carrying the time stamps in the packet, only 11 bytes are required. In the simulation, MAC frame size has also been taken as 11 bytes for both data as well as control packets for the sake of simplicity. But, the time slots are assigned assuming much higher payload capacity of order of 500 bytes. Hence, time slot duration of 2 s have been assigned for all the nodes in the network.

In Subsection 6.5.3, the simulation results have been illustrated in terms of actual time stamps while transmitting and receiving the packets in network.

7.3.5 Simulation results

Simulation of cross-layer protocol stack has been performed multiple times and for multiple cycles on the simulator. In this subsection, analysis of the representative examples of the simulation results has been provided from log files and trace files. Detail results have been provided in the Appendix B.

7.3.5.1 Results of transmission delay

It has been observed in the simulation that the Control cycle of the first Master cycle starts at 4.00 s, wherein BS(100) sends control information (first message of Tri-message time synchronization) to the CH1(200). Time stamps obtained from trace file, (in a standard NS2 format), are as follows:

$$+ \text{ time (Enqueue time of packet) } = 4.000000 \text{ s.}$$

$$- \text{ time (Dequeue time of packet) } = 4.050000 \text{ s.}$$

r time (Time of receive event at receiver) = 4.436000 s.

and Propagation speed is 1538.38 m/s.

Total delay of transmission can be calculated using as follows:

$$\text{Delay} = (\text{r time}) - (\text{-time}) \quad (7.3)$$

From the above example, delay value is 0.386 s. This delay is equal to \hat{D} (vertical link) along with additional factor of 0.025 as the preamble duration in the simulator.

Similarly, for the horizontal link of 10 m, the time stamps obtained are as follows:

+ time (Enqueue time of packet) = 28.000000 s.

- time (Dequeue time of packet) = 28.050000 s.

r time (Time of receive event at receiver) = 28.115000 s.

Delay value in this example is 0.065 s. This delay is equal to \hat{D} (horizontal link) along with preamble duration of 0.025 s.

7.3.5.2 Results of time synchronization

To illustrate the example of time synchronization results, two pairs of nodes have been chosen. First pair is BS(100) synchronizing CH1(200) (in time slots 1 to 4) and the other pair is CH1(200) synchronizing CN(201) in time slots 13 to 16. Results have been tabulated in the Table 7.6 and Table 7.7. In Table 7.6, time stamps of message exchanges between BS(100) and CH1(200) have been shown, along with results of clock skew and offset. Similarly, in Table 7.7, time stamps of message exchanges between CH1(200) and CN(201) have been shown, along with results of clock skew and offset. These results have been validated using (5.31) and (5.32).

BS(100)		CH1(200)	
Message	Time stamps (s)	Message	Time stamps (s)
A1	4.000000	B1	4.375020
A2	6.375020	B2	6.000000
A3	8.000000	B3	8.375020
Skew = 1.0 and Offset = 0.0 s			

Table 7.6. Result of time synchronization for BS(100) and CH1(200) nodes.

CH1(200)		CN(201)	
Message	Time stamp (s)	Message	Time stamp (s)
A1	28.000000	B1	28.054032
A2	30.054032	B2	30.000000
A3	32.000000	B3	32.054032
Skew = 1.0 and Offset = 0.0 s			

Table 7.7. Result of time synchronization for CH1(200) and CN(201) nodes.

7.3.5.3 Illustration of Data cycle phase

To illustrate the functioning of Data cycle in simulation, the time stamps of data transmissions of network nodes have been tabulated in Table 7.8. It can be observed that, Data cycle starts with time slot 49 at time 100.0 s. In a time slot of 2 s, one node of each cluster has sent data to the CH nodes. After collecting data from all the CN nodes, data has been transmitted to BS using backbone link. In the table, S.T. represent time stamp of transmission by a node and R.T. represent time stamp of receive event at intended receiver node. All timing values are represented in seconds.

7.3.5.4 Results of energy consumption

By considering $P_H = 0.01$ W, $P_L = 10^{-6}$ W and $P_{RX} = 10^{-5}$ W and using (6.10), (6.11) and (6.12), energy consumption of nodes can be calculated. Energy values of first Master cycle have been tabulated in the Table 7.9 along with number of packets transmitted and received by the node.

As mentioned in Subsection 7.3.2, in cycle 2, three nodes have been added in each cluster. These nodes are numbered as L04 to L06, and they occupy time slots reserved for future expansion in Control cycle as well as Data cycle phase. In cycle 3, nodes L07 to L09 have been added in each cluster. Cycle 4 continues with all 10 nodes of each cluster, wherein BS can collect energy information from the nodes. In cycle 5, Cluster-Head nodes have been rotated based on energy consumption. Since new nodes L07 to L09 consume least energy, one among these nodes can be made as new CH nodes. L07 node of each cluster have been considered for the CH nodes based on ID number of the node. Energy consumption values of BS and CH nodes in cycle 2 and 3 have been provided in Table 7.10, Table 7.11. For cycle 4, energy consumption values are same as that of cycle 3. In Figure 7.2, cumulative consumption of energy after cycle 1, 2, 3 and 4 by these

Time slot 49	Time slot 50	Time slot 51	Six time slot period (time slots 52 to 57)	Time slot 58	Time slot 59	Time slot 60
S.T. 100.00	S.T. 102.00	S.T. 104.00	Sleep period from 106.00 to 118.00	S.T. 118.00	S.T. 120.00	S.T. 122.00
R.T. 100.12	R.T. 102.12	R.T. 104.12		R.T. 118.44	R.T. 120.44	R.T. 122.44
CN (201) to CH1 (200)	CN (202) to CH1 (200)	CN (203) to CH1 (200)	Sleep period	CH3 (400) to CH2 (300)	CH2 (300) to CH1 (200)	CH1 (200) to BS (100)
CN (301) to CH2 (300)	CN (302) to CH2 (300)	CN (303) to CH2 (300)		Sleep period		
CN (401) to CH3 (400)	CN (402) to CH3 (400)	CN (403) to CH3 (400)	Sleep period			

Table 7.8. Results of Data cycle phase of first Master cycle on UnetSim simulator. (Values of S.T. and R.T. are in seconds.)

	BS(100)	CH1(200)	CH2(300)	CH3(400)
Number of packets transmitted	2	11	11	9
Number of packets received	3	14	14	11
Energy consumed in transmission (mJ)	0.733	1.837	1.837	1.103
Energy consumed in reception (mJ)	0.001	0.005	0.005	0.004
Total energy consumed (mJ)	0.734	1.842	1.842	1.107

Table 7.9. Results of energy consumption of BS and CH nodes in first Master cycle of proposed protocol stack.

nodes has been plotted. In Table 7.12, energy consumption of CN node L01, L04, and L07 after 4 cycles has been provided.

Table 7.13 and Table 7.14 illustrates the Control cycle phase and Data cycle phase of CH rotation

	BS(100)	CH1(200)	CH2(300)	CH3(400)
Number of packets transmitted	2	17	17	15
Number of packets received	3	23	23	20
Energy consumed in transmission (mJ)	0.733	1.840	1.840	1.107
Energy consumed in reception (mJ)	0.001	0.008	0.008	0.007
Total energy consumed (mJ)	0.734	1.849	1.849	1.114

Table 7.10. Results of energy consumption of BS and CH nodes in second Master cycle of proposed protocol stack.

	BS(100)	CH1(200)	CH2(300)	CH3(400)
Number of packets transmitted	2	23	23	21
Number of packets received	3	32	32	29
Energy consumed in transmission (mJ)	0.733	1.844	1.844	1.110
Energy consumed in reception (mJ)	0.001	0.012	0.012	0.011
Total energy consumed (mJ)	0.734	1.856	1.856	1.121

Table 7.11. Results of energy consumption of BS and CH nodes in third Master cycle of proposed protocol stack.

	CN(L01)	CN(L04)	CN(L07)
Total energy consumed (mJ)	0.010	0.008	0.005

Table 7.12. Results of total energy consumption of CN nodes in four cycles of proposed protocol stack.

in cycle 5 respectively. These tables indicate the exchange of time slots between new CH and old CH nodes. Also, routing of information is performed via new CH nodes representing a dynamic change of route in the network. It should be noted that, new CH nodes follow the dynamic power level changes depending on communication over horizontal and vertical links.

To indicate the balance of energy consumption after CH rotation, energy consumption of new CH nodes CH1(207), CH2(307) and CH3(407) have been tabulated along with old CH nodes CN(200),

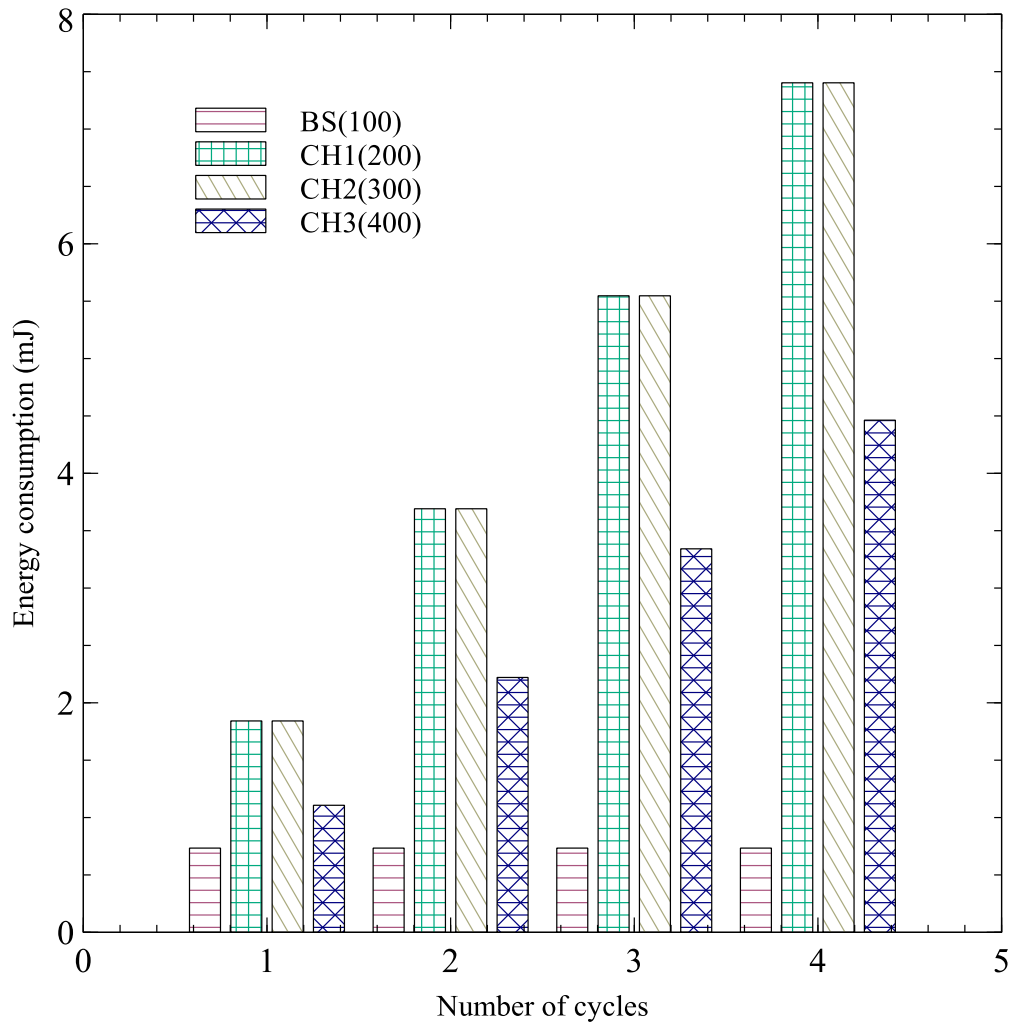


Figure 7.2. Overall energy consumption of BS and CH nodes after cycle 1, 2, 3 and 4.

CN(300), CN(400) after fifth and sixth cycles in Table 7.15. It can be observed that new CH nodes consume higher energy per cycle as compared to CN nodes as expected.

7.3.5.5 Evaluating network performance

In this section, the evaluation of network performance has been provided using parameters such as end-to-end delay, throughput and channel utilization.

Equation (6.5) can be modified for calculation of end-to-end delay on forward link as follows:

$$\text{Delay}_{\text{FL}} = 4 \times (L_T - 1) \times T_S + 4 \times (XX - 1) \times T_S \quad (7.4)$$

Time slot 1 to 4	Time slot 5 to 8	Time slot 9 to 12	Time slot 13 to 16	Time slot 17 to 20	Time slot 21 to 24	Time slot 37 to 40	Time slot 41 to 44	Time slot 45 to 48
BS (100) to CH1 (207)	CH1 (207) to CH2 (307)	CH2 (307) to CH3 (407)	CH1 (207) to CN (201)	CH1 (207) to CN (202)	CH1 (207) to CN (203)	CH1 (207) to CN (200)	CH1 (207) to CN (208)	CH1 (207) to CN (209)
			CH2 (307) to CN (301)	CH2 (307) to CN (302)	CH2 (307) to CN (303)	CH2 (307) to CN (300)	CH2 (307) to CN (308)	CH2 (307) to CN (309)
			CH3 (407) to CN (401)	CH3 (407) to CN (402)	CH3 (407) to CN (403)	CH3 (407) to CN (400)	CH3 (407) to CN (408)	CH3 (407) to CN (409)

Table 7.13. Illustration of Control cycle phase of fifth Master cycle in the implementation of proposed protocol stack in UnetSim.

Time slot 49	Time slot 50	Time slot 51	Time slot 55	Time slot 56	Time slot 57	Time slot 58	Time slot 59	Time slot 60
CN (201) to CH1 (207)	CN (202) to CH1 (207)	CN (203) to CH1 (207)	CN (200) to CH1 (207)	CN (208) to CH1 (207)	CN (209) to CH1 (207)	CH3 (407) to CH2 (307)	CH2 (307) to CH1 (207)	CH1 (207) to BS (100)
CN (301) to CH2 (307)	CN (302) to CH2 (307)	CN (303) to CH2 (307)	CN (300) to CH2 (307)	CN (308) to CH2 (307)	CN (309) to CH2 (307)			
CN (401) to CH3 (407)	CN (402) to CH3 (407)	CN (403) to CH3 (407)	CN (400) to CH3 (407)	CN (408) to CH3 (407)	CN (409) to CH3 (407)			

Table 7.14. Illustration of Data cycle phase of fifth Master cycle in the implementation of proposed protocol stack in UnetSim.

Equation (6.6) can be used for calculation of end-to-end delay on reverse link. These delay values have been tabulated in Table 7.16.

Node	Energy consumption after 5 cycles (mJ)	Energy consumption after 6 cycles (mJ)
CH1(207)	1.861	3.717
CH2(307)	1.861	3.717
CH3(407)	1.126	2.247
CN(200)	7.406	7.409
CN(300)	7.406	7.409
CN(400)	4.466	4.469

Table 7.15. Comparison of energy consumption of new CH and old CH nodes after CH rotation.

Parameter	Value (s)
Delay _{FL} for Node L01	24
Delay _{FL} for Node L09	88
Average Delay _{FL}	56
Delay _{RL} for Node L01	24
Delay _{RL} for Node L09	8
Average Delay _{RL}	16
Average end-to-end Delay	36

Table 7.16. Results of average end-to-end delay in proposed protocol stack implementation on UnetSim.

Throughput value for the proposed protocol stack implementation using (6.7) is 2.44 bps. Similarly, Channel utilization parameter can be calculated using (6.8). In a current scenario, channel reuse factor is 3. For the payload of 11 bytes, the channel utilization value is 0.006%. Reason for this lower value is the underutilization of time slot of 2 s by the payload of 11 bytes. Maximum payload of 500 bytes can be accommodated in this time slot, resulting in $\hat{T} = 1.67$ s and channel utilization as 0.27%.

The proposed protocol stack has been implemented on hardware testbed set-up described in Chapter 4. The details of this implementation have been provided in the next section.

7.4 Hardware testbed implementation of proposed protocol stack

Figure 4.4 shows the testbed set-up installed in the laboratory. For the testing of proposed protocol stack on this testbed set-up four underwater acoustic nodes have been built using Simple Acoustic

Modems (SAMs) [180], TelosB motes [218], interfacing units, and power supplies as described in Section 4.2. Different topologies used for implementing the cross-layer protocol stack has been described in the next subsection.

7.4.1 Network deployment

For the network deployment, two different implementation cases have been used as shown in Figure 7.3 and Figure 7.4. In Figure 7.3, working of complete vertical link has been illustrated whereas in Figure 7.4, functionality of one complete cluster has been studied. These deployments have been denoted as case 1 and case 2 respectively.

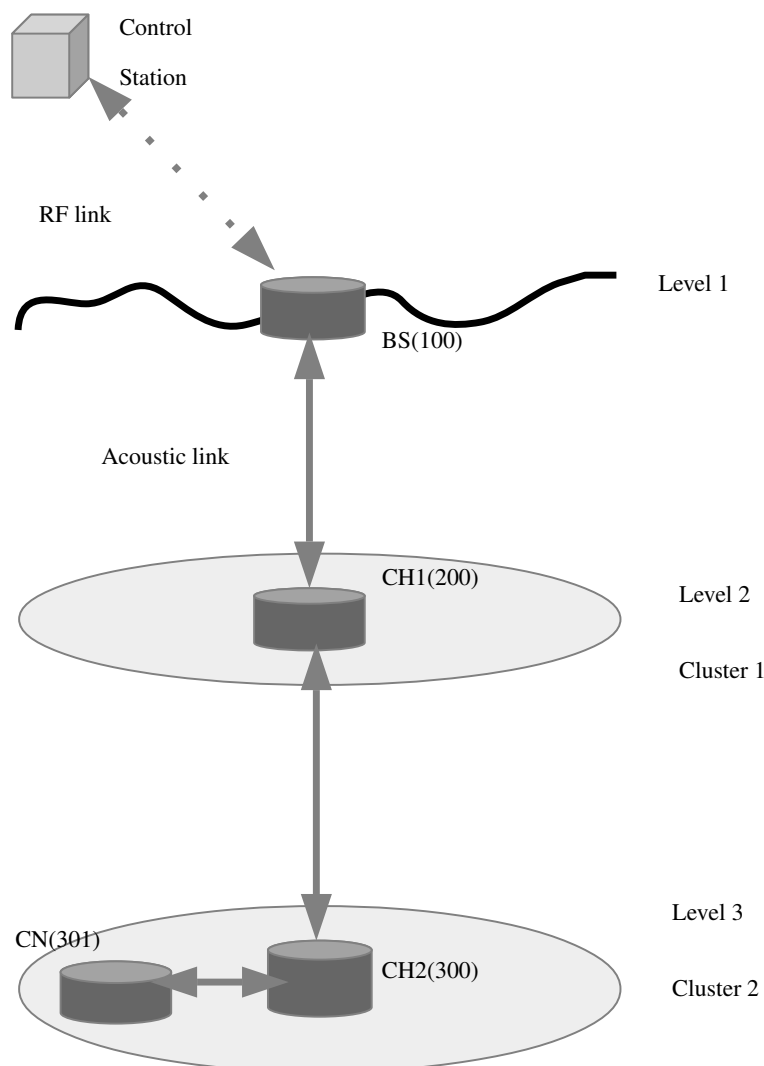


Figure 7.3. Node deployment for exploring functionality of vertical link (case 1).

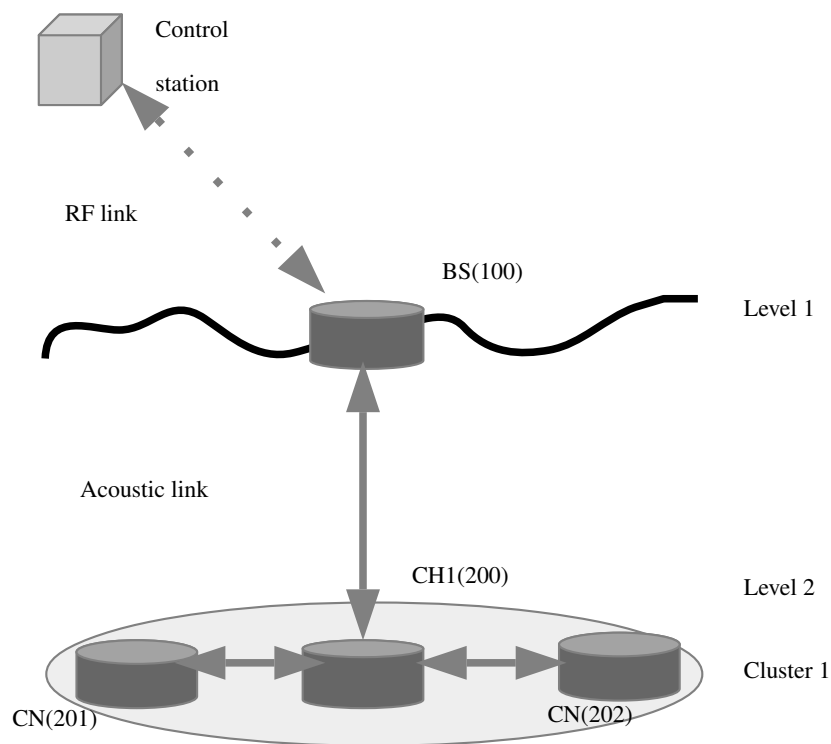


Figure 7.4. Node deployment for exploring functionality of horizontal link (case 2).

7.4.2 Physical layer parameters

A modem has been configured for the default available parameters as follows:

- Operating frequency = 33.8 kHz
- Transmission power = 183 dB
- Modulation technique = Pulse Position Modulation
- $R = 8$ bps

There are certain limitations imposed by the nature of testbed and functionalities of modem in the implementation of proposed protocol stack listed as follows:

1. The testbed set-up does not provide the depth dimension to implement actual three-dimensional network.

2. The modem does not have multiple power level settings. Hence, the parallel intra-cluster communication could not be tested. Rather, the transmissions are done sequentially on every possible link in the network. CH rotation has also not been attempted on the network because of limited resources.
3. Packet drops on the network deployed on this testbed is very high because of the reverberations in the water tank.

Working of vertical link has been tested in one case whereas cluster functionality has been tested in another case of network deployment.

7.4.3 Working of protocol stack

The proposed protocol stack has been implemented on the hardware testbed with limitations of the hardware as specified earlier. The setting of parameters such as data rate and packet size for time synchronization is similar to the setting explained in Section 5.5.

As shown in Figure 7.3, BS(100) receives the control information from control station on RF link. Control cycle of DCB-TDMA MAC protocol starts with BS(100) time synchronizing CH1(200). Four message exchanges (including the fourth message containing clock skew and offset) take place between each pair of synchronizing nodes. As shown in the figure, synchronization has been performed on the links namely BS(100)-CH1(200), CH1(200)-CH2(300), and then CH2(300)-CN(301). Using data rate of 8 bps, packet size of 22 bytes, and time interval of 10 s between messages, the entire duration of time synchronization on hardware testbed set-up is 6.4 minutes. Total duration has been kept as 19.2 minutes for this time synchronization phase. In case of guarantee of 100% PDR on the network, the extra duration can be used for addition of nodes in future expansion of network. Since the PDR of 75% has been observed on the network, this extra duration has been used for the provision of retransmission of time synchronization information. Two retransmission attempts have been provided on the network since achieving time synchronization is essential for proper functioning of network. If the synchronization is successful in the first attempt then the corresponding transmitter node has been programmed to sleep in order to save energy.

After 19.2 minutes, the data cycle starts. Data is sent only once without retransmission attempt or acknowledgment. Every node prepares a data packet of 22 bytes and has a time slot duration of 30 s. Data cycle starts with CN(301) sending the data packet to CH2(300). The data here has a time stamp of transmission included as a payload in HDLC frame. CH2(300) transmits a data packet to CH1(200), replacing the earlier time stamp by its own time stamp of transmission event. Similarly, CH1(200) transmits data packet to BS(100). Finally, BS(100) transmits the data packet using RF link to control station. Results of one run of Master cycle has been tabulated in Table 7.17. As shown in the table, the clock skew and offset values have been calculated for every synchronizing node in control cycle phase. Node implements offset correction after calculating these values. Data cycle starts after 19.2 minutes with each node sending the time stamp in the message. These time stamp values have been tabulated in the table.

Control Cycle (Time synchronization results)				Data Cycle		
Link	Time stamps (s)		Skew	Offset (s)	Link	Time stamps (s)
BS(100) to CH1(200)	A ₁ = 40.010 A ₂ = 94.080 A ₃ = 104.080	B ₁ = 62.020 B ₂ = 72.020 B ₃ = 126.128	1.000	-0.025	CN(301) to CH(300)	1152.384
CH1(200) to CH2(300)	B ₄ = 168.134 B ₅ = 222.152 B ₆ = 232.152	C ₁ = 190.146 C ₂ = 200.146 C ₃ = 254.168	1.000	0.003	CH2(300) to CH1(200)	1182.516
CH2(300) to CN(301)	C ₄ = 296.175 C ₅ = 350.214 C ₆ = 360.232	D ₁ = 318.198 D ₂ = 328.198 D ₃ = 382.227	0.9995	0.1650	CH1(200) to BS(100)	1212.448

Table 7.17. Result of protocol stack implementation on the hardware testbed set-up (case 1).

After the data cycle, all the nodes sleep for duration of 30 minutes, providing duty cycle of around 40%. At the end of sleep period, again the same sequence of time synchronization (19.2 minutes duration), data cycle (1.5 minutes duration), and sleep period (30 minutes duration) has been followed.

In Figure 7.4, network has been deployed to showcase the working on one complete cluster. The time synchronization in this case has been performed as follows: Initially, BS(100) synchronizes CH1(200). CH1(200) then synchronizes CN(201). Later CH1(200) synchronizes another cluster node CN(202). In data cycle, CH1(200) collects the data information from CN(201) and CN(202). After collecting data from the CN nodes, CH1(200) transmits data to BS(100). BS(100) then transmits data to control station using an RF link. Results of one run of Master cycle on this

deployment has been tabulated in Table 7.18. Clock skew and offset values have been shown in the table for every synchronizing node. The time gap between synchronization of CN(201) and CN(202) is because of packet drop and retransmission event. Data cycle starts after around 19.2 minutes. BS(100) has collected all the data information at the end of data cycle.

Control Cycle (Time synchronization results)				Data Cycle		
Link	Time stamps (s)		Skew	Offset (s)	Link	Time stamps (s)
BS(100) to CH1(200)	A ₁ = 34.032 A ₂ = 88.222 A ₃ = 98.222	B ₁ = 56.193 B ₂ = 66.193 B ₃ = 120.276	0.9983	0.1699	CN(201) to CH1(200)	1152.112
CH1(200) to CN(201)	B ₄ = 162.288 B ₅ = 216.312 B ₆ = 226.312	C ₁ = 184.297 C ₂ = 194.298 C ₃ = 248.338	1.000	-0.0025	CN(202) to CH1(200)	1182.285
CH1(200) to CN(202)	C ₄ = 756.135 C ₅ = 810.231 C ₆ = 820.231	D ₁ = 778.066 D ₂ = 788.067 D ₃ = 842.159	0.9999	-0.038	CH1(200) to BS(100)	1212.455

Table 7.18. Result of protocol stack implementation on the hardware testbed set-up (case 2).

In Section 7.5, a brief summary of the chapter has been provided.

7.5 Summary

In this chapter, a multi-hop Tri-message time synchronization protocol has been incorporated in the control cycle phase of DCB-TDMA MAC protocol, resulting in a single unified framework for the UASN. It has been showcased that the resultant framework is the cross-layer module since it includes important features of physical layer, data link layer, network layer, transport layer and application layer. Protocols of network management planes such as time synchronization, Cluster-Head selection and power level management has also been incorporated in the design. This cross-layer protocol stack provides energy efficient operation making it suitable for long duration UASN application.

The three-dimensional network architecture has been deployed using UnetSim simulation platform. This simulator has been chosen since it's UnetStack architecture has important feature of cross-layer information sharing. Using this simulator, number of clusters of nodes has been deployed at varying depths in the ocean. A cross-layer protocol stack has been simulated on this

network. Results of transmission delay, time synchronization, and energy consumption has been provided in detail. Network performance has been evaluated in terms of throughput and channel utilization.

This protocol stack has also been implemented on hardware testbed set-up of UASN. Two different topologies have been used to showcase the working of vertical link and horizontal link respectively. Results of this implementation has been showcased for Control cycle and Data cycle phases of DCB-TDMA MAC protocol. In Control cycle phase, multi-hop Tri-message time synchronization has been performed. In Data cycle, data is collected from sensor nodes of network at the Base-Station node. In simulations, ideal results has been obtained for time synchronization and PDR because of absence of jitter and multipath phenomenon. On the hardware testbed, effect of jitter and reverberation has resulted in coarse-grained time synchronization. Also, certain retransmission attempts has to be provided in the deployed network since the PDR is around 75% on testbed.

Chapter 8

Conclusions and Future Scope

8.1 Conclusions

In a traditional approach of ocean floor or ocean column monitoring, various sensor nodes are deployed to record the data during the monitoring mission. This data is available to end user only after recovering the sensor nodes from the sensor field. There are several issues with this approach such as 1) real time monitoring is not possible, 2) online system reconfiguration is not available, 3) failure of system can not be identified before recovering the sensor nodes (or instruments), and 4) amount of data is limited by the capacity of on-board memory devices.

These issues are overcome by utilizing an Underwater Acoustic Sensor Network (UASN). UASN is the network of sensor nodes deployed underwater to perform synoptic, co-operative and adaptive sampling of three-dimensional ocean environment. These nodes are networked via wireless links that rely on acoustic communication. A broad range of applications include i) ocean sampling networks, ii) environmental monitoring, iii) undersea explorations, iv) seismic monitoring, v) equipment monitoring, vi) assisted navigation and vii) distributed tactical surveillance operations.

Each of these applications might require completely different approach of design and development of UASN. Most important feature of sensor network development is the provision of flexibility it offers to the developer in the design of network architecture and protocol stack to suit the requirements of application. Correct choice of network architecture and optimal design of protocol

stack has a critical impact on parameters of network performance. Although numerous protocols have been proposed and developed by researchers for the terrestrial wireless sensor networks, these protocols can not be readily applied to UASN because of following peculiar properties of UASN:

- Extremely limited bandwidth of underwater acoustic communication.
- Large and variable propagation delay of acoustic signals in an underwater channel.
- Spatially and temporarily varying channel impulse response.
- Higher probability of bit error and temporary loss of connectivity (shadow zones).
- Harsh and hostile underwater environment damaging nodes due to fouling and corrosion.
- Higher cost of acoustic modems, deployment and recovery operations.
- Severe resource constraints including available energy.

To address these challenges and resource constraints of UASN, it is advocated to use a cross-layer protocol stack implementation. In cross-layer design, functionalities of two or more layers can be combined to form a single coherent framework. Recent studies on WSNs reveal that cross-layer integration and design techniques result in significant improvement in terms of energy conservation. Such an approach is now being tested for the UASN as well.

In this work, a problem of designing three-dimensional UASN has been considered. Such a network can be used for long-term ocean column monitoring application. For this network, a static and structured three-dimensional network architecture has been proposed. To achieve a longer operational life-time of the network, energy efficiency has been the main focus of the design.

The proposed network has a Base-Station node deployed on the sea-surface. BS communicates with the on-shore control station using the RF link. Multiple clusters have been deployed below each other at various depths from the sea-surface. These clusters consist of Cluster-Head node along with certain number of cluster member nodes. The functionalities and addresses of nodes have been configured at the time of deployment of network. The optimum frequency, data rate, modulation technique and power levels have been chosen for nodes based on the characteristics of deep water acoustic channel model.

For the proposed cluster-based three-dimensional network, a cross-layer protocol stack has been developed encompassing important functionalities of physical layer, data link layer, network layer, transport layer, and application layer. Various protocols of network management planes such as time synchronization, clustering, and power level management have also been incorporated in the proposed cross-layer protocol stack. Novel features of the protocol stack can be briefly summarized layer-wise as follows:

1. Physical layer - The design and modeling of physical layer has been done taking into consideration the role of a node in the network as well as the location of deployment. A multi-hop communication link is preferred over the single direct communication link for improving energy as well as bandwidth utilization. Cluster-Head node communicates over long range vertical link as well as small range horizontal link. For these communications, Cluster-Head node adapts an optimum power level for transmission, making effective use of energy. Also, the power levels have been chosen in such a way to allow multiple parallel intra-cluster communication without requirement of multiple frequencies or codes. The provision of multiple parallel communication increases the channel utilization in a severely bandwidth constrained UASN.
2. Data link layer - At the Data Link Layer, a new variant of TDMA MAC has been proposed. TDMA based MAC has been considered, since it offers fairness among network nodes and simplicity of design while avoiding packet collisions, overhearing, and idle listening issues. Proposed variant of TDMA is termed as “Dynamic Cluster-Based TDMA (DCB-TDMA)” MAC protocol, since it utilizes dynamic power level setting of the nodes, and also allows the nodes to be dynamically added or removed from the network. It has been shown that the protocol consists of route formation, Cluster-Head selection and effective power level management schemes. The protocol has been implemented on SUNSET, an open source simulation platform developed by SENSES lab, Sapienza University, Rome. The features of protocol such as node addition, node deletion, Cluster-Head selection, route formation has been showcased using the simulation results. It has been demonstrated that the performance of protocol is better as compared to Basic TDMA based MAC protocol implemented on the same network architecture. Also, the energy awareness feature has been evident from the fact that the protocol has around 50% duty cycle and it balances the energy consumption

of nodes by rotating Cluster-Head nodes based on energy consumption information. Nodes are programmed to sleep for the duration other than the pre-assigned time slots leading to effective energy utilization.

3. Network layer - A multi-hop route has been provided for the forward and reverse link on the network. Route re-establishment in case of Cluster-Head selection has also been showcased using simulation tools.
4. Transport layer - An end-to-end link between BS and every cluster member node has been provided in the network. The reliability and fairness of data transmission on these links has been demonstrated using software tools.
5. Application layer - The application involving collection of data from sensor nodes at regular intervals has been showcased in the work.
6. Time synchronization - In this work, a multi-hop time synchronization protocol has been proposed and developed. This protocol is an extension of Tri-message time synchronization which achieves the time synchronization using only three message exchanges. It has been demonstrated that the protocol provides better performance in terms of accuracy under various situations such as i) delay variations, ii) jitter distribution, iii) time gap from last synchronization. It has also been shown that the protocol consumes very less energy making it energy efficient protocol. Computational cost and time required for achieving synchronization is also lesser as compared to other regression based techniques of time synchronization.
7. Cluster-Head selection - In a proposed protocol stack, energy consumption information of all the network nodes has been regularly collected by the BS. Depending on the energy threshold, a Cluster-Head node has been selected from among the cluster member nodes. This feature has been demonstrated using the simulation tools. The Cluster-Head selection ensures the balancing of energy consumption among all nodes, improving life-span of network. Any failure of network because of energy depleted nodes is also prevented. A chain of Cluster-Head nodes act as a backbone link. In case of a non-responsive Cluster-Head node, a new Cluster-Head node can be selected by the network.

8. Power level management - A Cluster-Head node communicates with other Cluster-Head nodes on vertical link of 500 m using a higher power level of transmission. On the other hand, it communicates with cluster member nodes located in radius of 10 m using a lower power level. This power levels are dynamically adjusted by the Cluster-Head node. Similarly, every node turns off the radio and enters into sleep mode in sleep cycle as well as for duration other than pre-assigned time slots to conserve energy. These features of power level management has been showcased using analytical modeling and simulation tools.

Different software simulation tools have been explored to demonstrate various functionalities of the proposed protocol stack and to validate the results of theoretical modeling. MATLAB simulation has been used to demonstrate time synchronization protocol and to perform comparative analysis with other protocols of time synchronization. SUNSET simulation platform has been used to successfully implement the DCB-TDMA MAC protocol. A three-dimensional cluster-based network has been used for testing the MAC protocol. Around 45 nodes have been included in the network to showcase the scalability. Various features of the protocol such as node addition, deletion, Cluster-Head selection, energy consumption has been successfully demonstrated. The complete protocol stack including multi-hop Tri-message time synchronization has been implemented on a similar three-dimensional network using UnetSim simulator. Analysis of various network parameters has been performed using this simulation tool.

It is a common understanding in the research community that “Simulations are doomed to succeed.” Actual testing of protocol stack on hardware testbed is very important to understand the effects of practical environment. For this purpose, a laboratory based testbed set-up has been built to identify issues related to implementation of the network protocol stack on actual hardware set-up. Proposed cross-layer protocol stack has been deployed on this testbed set-up using various possible topologies. The effect of limitations of hardware on the protocol stack have also been tested. In this way, a testbed set-up has served as a platform for validation of theoretical and simulation results of protocols developed for UASN.

Overall, this work can be taken as a reference guide for design and development of three-dimensional UASN applications. Knowing the requirements of application such as i) types of sensors, ii) time interval of data collection, and iii) dimensions of column can help in deciding the required networking parameters using this work.

8.2 Future scope

Work undertaken in this thesis can be extended in future in following directions :

1. In the proposed network architecture, it has been assumed that the nodes are placed at their respective levels and positions at the time of initial deployment. Hence, the nodes are assumed to be aware of their locations. Such a network architecture is an example of structured deployment. In sensor networks, nodes are usually dropped in a sensing field in a random manner with no fine control over the placements. In such a random deployment scenario, nodes are supposed to have self-organization capability to form an easily manageable network architecture by performing neighbor discovery and positioning algorithms. To obtain position information of the node, a localization algorithm needs to be developed. Localization algorithm provides many features such as correlation of collected data, geographical routing, and node addressing. Also, for the efficient utilization of resources in sensor networks, the sensor nodes are often grouped into disjoint, non-overlapping clusters. For organizing sensor nodes into clusters, a distributed clustering algorithm is necessary. Various such important network services needs to be developed for the self organization of the network, if random deployment is considered.
2. Depending on the interval of sensing and data collection, a huge amount of data transfer takes place in the network. Typical characteristic of the sensor network is the redundant low-rate data and many-to-one flows. An effective data aggregation algorithm can be designed in the sensor network to combine data from several sensors to eliminate redundant transmissions. Such a data-centric approach can minimize the energy consumption prolonging the life-time of the network. In the proposed network architecture, data fusion or aggregation scheme can be applied at Cluster-Head nodes. Such data aggregation scheme may require provision of variable packet size and varying time slot period.

Appendix A

Simple Acoustic Modem (SAM)

Simple Acoustic Modem (SAM) is a miniature acoustic modem. It is developed by Desert Star Systems LLC, an underwater technology manufacturer located in Monterey Bay, California. These modems have been chosen for the indoor testbed development described in Chapter 4. Details of SAM modems has been provided in this appendix.

SAM is a simple, effective and low-cost acoustic modem for variety of applications that demand reliable data exchange at slow to medium speeds. It can work in a variety of underwater environments, from shallow areas or harbors all the way to deep oceanic waters. SAM has following features:

- Small size: 135 mm length x 40 mm diameter
- Energy efficient communication and low standby power consumption
- Range: 250 m (typical), 1000 m (ideal, deep ocean)
- Acoustic data exchange in shallow, confined and deep waters alike
- Works in high multi-path and noisy environments
- Robust data exchange format, including checksum
- Instant operation (no configuration) for use with “dumb devices”

- Configuration through serial commands for use with “smart devices”

Certain possible applications of SAM modules are as follows:

- Retrieval of data from underwater instruments
- Control of underwater instruments and equipment
- Control of Autonomous Underwater Vehicles (AUV)
- An acoustic teletype for communication with submarines or among submarines
- Cost saving replacement of wired data links
- Marine mammal telemetry
- Acoustic diver or vehicle identification
- Remote control of lift bags
- Acoustic releases
- Communication with stealthy instruments that are buried into the sea floor

Specifications of SAM are as follows:

- Size: 135 mm L x 40 mm D (5.3” L x 1.6” D)
- Weight: 230 g in air and 50 g in water
- Depth Rating: 300 m max. working depth
- Operating temperature: 0-70 degrees Celsius
- Data I/O: RS-232 serial data link
- Serial Buffer: The size of the serial buffer is 32 bytes
- Sonar transceiver: TX power 183 dB at 8 V and 189 dB at 16 V

- 4-channel frequency hopping transmitter, 33.8 kHz- 42 kHz standard, 65 kHz - 75 kHz optional
- RX sensitivity: ≤ 95 dB re 1 μ Pa
- Single channel receiver, 33.8 kHz
- Sonar transducers: 33 kHz - 42 kHz omni-directional transducer standard
- Sonar range: 100 - 1000 m communication range
- Sonar modulation: Single channel receiver, pulse position modulation
- Sonar data format: 16 bit data + 4 bit checksum per word
- Sonar bit rate (receive) : 3-23 bps actual user data (5 - 35 bps raw)
- Sonar bit rate (transmit) : 3-96 bps actual user data (5 - 150 bps raw)
- Power consumption: 8 V to 16 V supply voltage
 - Up to 21 mA in sonar receive mode
 - Up to 2 Ampere peak in high-power sonar transmit mode
- Data transmit power:
 - Around 30 mAh per 1000 user data bytes with 16 V supply
 - Around 15 mAh per 1000 user data bytes with 8 V supply

Appendix B

Results of Implementation of Cross-layer Protocol Stack

We have used UnetSim simulator for the implementation of proposed cross-layer protocol stack as described in Section 7.3. Results of simulation has been provided in this appendix. These results are part of log file and trace file outputs generated by UnetSim after running a simulation.

The node numbering and corresponding address configuration has been provided in Table B.1. Further, along with these nodes, a BS node has been configured with address 100 and deployed at the sea-surface.

Cluster 1		Cluster 2		Cluster 3	
Node No.	Node Address	Node No.	Node Address	Node No.	Node Address
10	200	20	300	30	400
11	201	21	301	31	401
12	202	22	302	32	402
13	203	23	303	33	403
14	204	24	304	34	404
15	205	25	305	35	405
16	206	26	306	36	406
17	207	27	307	37	407
18	208	28	308	38	408
19	209	29	309	39	409

Table B.1. Node numbers and addresses in UnetSim simulator.

Results of time synchronization (for cycle 1) are as follows:

```
1
2 8436|INFO|cbtdma@20:println|cycle : 1 node : 10 A1[0] = 4000000
3 A2[0] = 4375020 A3[0] = 6000000 A4[0] = 6375020 A5[0] = 8000000
4 A6[0] = 8375020
5
6 8436|INFO|cbtdma@20:println|cycle : 1 node : 10 skew : 1.0, offset : 0.0
7
8 16434|INFO|cbtdma@60:println|cycle : 1 node : 20 A1[0] = 12000000
9 A2[0] = 12373130 A3[0] = 14000000 A4[0] = 14373130 A5[0] = 16000000
10 A6[0] = 16373130
11
12 16434|INFO|cbtdma@60:println|cycle : 1 node : 20 skew : 1.0, offset : 0.0
13
14 24441|INFO|cbtdma@100:println|cycle : 1 node : 30 A1[0] = 20000000
15 A2[0] = 20379674 A3[0] = 22000000 A4[0] = 22379674 A5[0] = 24000000
16 A6[0] = 24379674
17
18 24441|INFO|cbtdma@100:println|cycle : 1 node : 30 skew : 1.0, offset : 0.0
19
20 32115|INFO|cbtdma@24:println|cycle : 1 node : 11 A1[0] = 28000000
21 A2[0] = 28054032 A3[0] = 30000000 A4[0] = 30054032 A5[0] = 32000000
22 A6[0] = 32054032
23
24
25 32115|INFO|cbtdma@24:println|cycle : 1 node : 11 skew : 1.0, offset : 0.0
26
27 32119|INFO|cbtdma@104:println|cycle : 1 node : 31 A1[0] = 28000000
28 A2[0] = 28057919 A3[0] = 30000000 A4[0] = 30057919 A5[0] = 32000000
29 A6[0] = 32057919
30
31
32 32119|INFO|cbtdma@104:println|cycle : 1 node : 31 skew : 1.0, offset : 0.0
33
34 32119|INFO|cbtdma@64:println|cycle : 1 node : 21 A1[0] = 28000000
35 A2[0] = 28057934 A3[0] = 30000000 A4[0] = 30057934 A5[0] = 32000000
36 A6[0] = 32057934
37
```

```
38 32119|INFO|cbtdma@64:println|cycle : 1 node : 21 skew : 1.0, offset : 0.0
39
40 40118|INFO|cbtdma@28:println|cycle : 1 node : 12 A1[0] = 36000000
41 A2[0] = 36057140 A3[0] = 38000000 A4[0] = 38057140 A5[0] = 40000000
42 A6[0] = 40057140
43
44 40118|INFO|cbtdma@108:println|cycle : 1 node : 32 A1[0] = 36000000
45 A2[0] = 36056908 A3[0] = 38000000 A4[0] = 38056908 A5[0] = 40000000
46 A6[0] = 40056908
47
48 40118|INFO|cbtdma@28:println|cycle : 1 node : 12 skew : 1.0, offset : 0.0
49
50 40118|INFO|cbtdma@68:println|cycle : 1 node : 22 A1[0] = 36000000
51 A2[0] = 36056891 A3[0] = 38000000 A4[0] = 38056891 A5[0] = 40000000
52 A6[0] = 40056891
53
54 40118|INFO|cbtdma@108:println|cycle : 1 node : 32 skew : 1.0, offset : 0.0
55
56 40118|INFO|cbtdma@68:println|cycle : 1 node : 22 skew : 1.0, offset : 0.0
57
58 48115|INFO|cbtdma@72:println|cycle : 1 node : 23 A1[0] = 44000000
59 A2[0] = 44054496 A3[0] = 46000000 A4[0] = 46054496 A5[0] = 48000000
60 A6[0] = 48054496
61
62 48115|INFO|cbtdma@72:println|cycle : 1 node : 23 skew : 1.0, offset : 0.0
63
64 48116|INFO|cbtdma@112:println|cycle : 1 node : 33 A1[0] = 44000000
65 A2[0] = 44055010 A3[0] = 46000000 A4[0] = 46055010 A5[0] = 48000000
66 A6[0] = 48055010
67
68 48116|INFO|cbtdma@112:println|cycle : 1 node : 33 skew : 1.0, offset : 0.0
69
70 48118|INFO|cbtdma@32:println|cycle : 1 node : 13 A1[0] = 44000000
71 A2[0] = 44057010 A3[0] = 46000000 A4[0] = 46057010 A5[0] = 48000000
72 A6[0] = 48057010
73
74 48118|INFO|cbtdma@32:println|cycle : 1 node : 13 skew : 1.0, offset : 0.0
```

Time-stamps of various events in cycle 1 are as follows:

```

1 BEGIN SIMULATION 1
2
3 + -t 4.000000 -s 1 -d 10 -i 1570367547 -p 1 -x {1.0 10.0 -1 ----- null}
4 - -t 4.050000 -s 1 -d 10 -i 1570367547 -p 1 -x {1.0 10.0 -1 ----- null}
5 r -t 4.436000 -s 1 -d 10 -i 1570367547 -p 1 -x {1.0 10.0 -1 ----- null}
6
7 + -t 6.000000 -s 10 -d 1 -i 1384669905 -p 1 -x {10.0 1.0 -1 ----- null}
8 - -t 6.050000 -s 10 -d 1 -i 1384669905 -p 1 -x {10.0 1.0 -1 ----- null}
9 r -t 6.436000 -s 10 -d 1 -i 1384669905 -p 1 -x {10.0 1.0 -1 ----- null}
10
11 + -t 8.000000 -s 1 -d 10 -i 1696443173 -p 1 -x {1.0 10.0 -1 ----- null}
12 - -t 8.050000 -s 1 -d 10 -i 1696443173 -p 1 -x {1.0 10.0 -1 ----- null}
13 r -t 8.436000 -s 1 -d 10 -i 1696443173 -p 1 -x {1.0 10.0 -1 ----- null}
14
15 + -t 10.000000 -s 10 -d 1 -i 1001924083 -p 1 -x {10.0 1.0 -1 ----- null}
16 - -t 10.050000 -s 10 -d 1 -i 1001924083 -p 1 -x {10.0 1.0 -1 ----- null}
17 r -t 10.436000 -s 10 -d 1 -i 1001924083 -p 1 -x {10.0 1.0 -1 ----- null}
18
19
20 + -t 12.000000 -s 10 -d 20 -i 827089104 -p 1 -x {10.0 20.0 -1 ----- null}
21 - -t 12.050000 -s 10 -d 20 -i 827089104 -p 1 -x {10.0 20.0 -1 ----- null}
22 r -t 12.434000 -s 10 -d 20 -i 827089104 -p 1 -x {10.0 20.0 -1 ----- null}
23
24 + -t 14.000000 -s 20 -d 10 -i 1318022842 -p 1 -x {20.0 10.0 -1 ----- null}
25 - -t 14.050000 -s 20 -d 10 -i 1318022842 -p 1 -x {20.0 10.0 -1 ----- null}
26 r -t 14.434000 -s 20 -d 10 -i 1318022842 -p 1 -x {20.0 10.0 -1 ----- null}
27
28 + -t 16.000000 -s 10 -d 20 -i -525077670 -p 1 -x {10.0 20.0 -1 ----- null}
29 - -t 16.050000 -s 10 -d 20 -i -525077670 -p 1 -x {10.0 20.0 -1 ----- null}
30 r -t 16.434000 -s 10 -d 20 -i -525077670 -p 1 -x {10.0 20.0 -1 ----- null}
31
32 + -t 18.000000 -s 20 -d 10 -i 415749518 -p 1 -x {20.0 10.0 -1 ----- null}
33 - -t 18.050000 -s 20 -d 10 -i 415749518 -p 1 -x {20.0 10.0 -1 ----- null}
34 r -t 18.434000 -s 20 -d 10 -i 415749518 -p 1 -x {20.0 10.0 -1 ----- null}
35
36 + -t 20.000000 -s 20 -d 30 -i -61795458 -p 1 -x {20.0 30.0 -1 ----- null}
37 - -t 20.050000 -s 20 -d 30 -i -61795458 -p 1 -x {20.0 30.0 -1 ----- null}

```



```
38 r -t 20.441000 -s 20 -d 30 -i -61795458 -p 1 -x {20.0 30.0 -1 ----- null}
39
40 + -t 22.000000 -s 30 -d 20 -i 842211326 -p 1 -x {30.0 20.0 -1 ----- null}
41 - -t 22.050000 -s 30 -d 20 -i 842211326 -p 1 -x {30.0 20.0 -1 ----- null}
42 r -t 22.441000 -s 30 -d 20 -i 842211326 -p 1 -x {30.0 20.0 -1 ----- null}
43
44 + -t 24.000000 -s 20 -d 30 -i 1620209237 -p 1 -x {20.0 30.0 -1 ----- null}
45 - -t 24.050000 -s 20 -d 30 -i 1620209237 -p 1 -x {20.0 30.0 -1 ----- null}
46 r -t 24.441000 -s 20 -d 30 -i 1620209237 -p 1 -x {20.0 30.0 -1 ----- null}
47
48 + -t 26.000000 -s 30 -d 20 -i -1304499182 -p 1 -x {30.0 20.0 -1 ----- null}
49 - -t 26.050000 -s 30 -d 20 -i -1304499182 -p 1 -x {30.0 20.0 -1 ----- null}
50 r -t 26.441000 -s 30 -d 20 -i -1304499182 -p 1 -x {30.0 20.0 -1 ----- null}
51
52 + -t 28.000000 -s 10 -d 11 -i 223865875 -p 1 -x {10.0 11.0 -1 ----- null}
53 + -t 28.000000 -s 30 -d 31 -i 1988512791 -p 1 -x {30.0 31.0 -1 ----- null}
54 + -t 28.000000 -s 20 -d 21 -i 1685286455 -p 1 -x {20.0 21.0 -1 ----- null}
55 - -t 28.050000 -s 10 -d 11 -i 223865875 -p 1 -x {10.0 11.0 -1 ----- null}
56 - -t 28.050000 -s 20 -d 21 -i 1685286455 -p 1 -x {20.0 21.0 -1 ----- null}
57 - -t 28.050000 -s 30 -d 31 -i 1988512791 -p 1 -x {30.0 31.0 -1 ----- null}
58 r -t 28.115000 -s 10 -d 11 -i 223865875 -p 1 -x {10.0 11.0 -1 ----- null}
59 r -t 28.119000 -s 20 -d 21 -i 1685286455 -p 1 -x {20.0 21.0 -1 ----- null}
60 r -t 28.119000 -s 30 -d 31 -i 1988512791 -p 1 -x {30.0 31.0 -1 ----- null}
61
62
63 + -t 30.000000 -s 11 -d 10 -i 1424156368 -p 1 -x {11.0 10.0 -1 ----- null}
64 + -t 30.000000 -s 31 -d 30 -i 1461853052 -p 1 -x {31.0 30.0 -1 ----- null}
65 + -t 30.000000 -s 21 -d 20 -i -299538998 -p 1 -x {21.0 20.0 -1 ----- null}
66 - -t 30.050000 -s 11 -d 10 -i 1424156368 -p 1 -x {11.0 10.0 -1 ----- null}
67 - -t 30.050000 -s 31 -d 30 -i 1461853052 -p 1 -x {31.0 30.0 -1 ----- null}
68 - -t 30.050000 -s 21 -d 20 -i -299538998 -p 1 -x {21.0 20.0 -1 ----- null}
69 r -t 30.115000 -s 11 -d 10 -i 1424156368 -p 1 -x {11.0 10.0 -1 ----- null}
70 r -t 30.119000 -s 21 -d 20 -i -299538998 -p 1 -x {21.0 20.0 -1 ----- null}
71 r -t 30.119000 -s 31 -d 30 -i 1461853052 -p 1 -x {31.0 30.0 -1 ----- null}
72
73
74 + -t 32.000000 -s 10 -d 11 -i 54861045 -p 1 -x {10.0 11.0 -1 ----- null}
75 + -t 32.000000 -s 20 -d 21 -i 369595108 -p 1 -x {20.0 21.0 -1 ----- null}
```

```
76 + -t 32.000000 -s 30 -d 31 -i 99125135 -p 1 -x {30.0 31.0 -1 ----- null}
77 - -t 32.050000 -s 10 -d 11 -i 54861045 -p 1 -x {10.0 11.0 -1 ----- null}
78 - -t 32.050000 -s 20 -d 21 -i 369595108 -p 1 -x {20.0 21.0 -1 ----- null}
79 - -t 32.050000 -s 30 -d 31 -i 99125135 -p 1 -x {30.0 31.0 -1 ----- null}
80 r -t 32.115000 -s 10 -d 11 -i 54861045 -p 1 -x {10.0 11.0 -1 ----- null}
81 r -t 32.119000 -s 30 -d 31 -i 99125135 -p 1 -x {30.0 31.0 -1 ----- null}
82 r -t 32.119000 -s 20 -d 21 -i 369595108 -p 1 -x {20.0 21.0 -1 ----- null}
83
84
85 + -t 34.000000 -s 31 -d 30 -i -289901623 -p 1 -x {31.0 30.0 -1 ----- null}
86 + -t 34.000000 -s 21 -d 20 -i -1650092310 -p 1 -x {21.0 20.0 -1 ----- null}
87 + -t 34.000000 -s 11 -d 10 -i 1221661861 -p 1 -x {11.0 10.0 -1 ----- null}
88 - -t 34.050000 -s 11 -d 10 -i 1221661861 -p 1 -x {11.0 10.0 -1 ----- null}
89 - -t 34.050000 -s 21 -d 20 -i -1650092310 -p 1 -x {21.0 20.0 -1 ----- null}
90 - -t 34.050000 -s 31 -d 30 -i -289901623 -p 1 -x {31.0 30.0 -1 ----- null}
91 r -t 34.115000 -s 11 -d 10 -i 1221661861 -p 1 -x {11.0 10.0 -1 ----- null}
92 r -t 34.119000 -s 31 -d 30 -i -289901623 -p 1 -x {31.0 30.0 -1 ----- null}
93 r -t 34.119000 -s 21 -d 20 -i -1650092310 -p 1 -x {21.0 20.0 -1 ----- null}
94
95 + -t 36.000000 -s 10 -d 12 -i 397418770 -p 1 -x {10.0 12.0 -1 ----- null}
96 + -t 36.000000 -s 20 -d 22 -i -662948331 -p 1 -x {20.0 22.0 -1 ----- null}
97 + -t 36.000000 -s 30 -d 32 -i -485929922 -p 1 -x {30.0 32.0 -1 ----- null}
98 - -t 36.050000 -s 10 -d 12 -i 397418770 -p 1 -x {10.0 12.0 -1 ----- null}
99 - -t 36.050000 -s 20 -d 22 -i -662948331 -p 1 -x {20.0 22.0 -1 ----- null}
100 - -t 36.050000 -s 30 -d 32 -i -485929922 -p 1 -x {30.0 32.0 -1 ----- null}
101 r -t 36.118000 -s 10 -d 12 -i 397418770 -p 1 -x {10.0 12.0 -1 ----- null}
102 r -t 36.118000 -s 20 -d 22 -i -662948331 -p 1 -x {20.0 22.0 -1 ----- null}
103 r -t 36.118000 -s 30 -d 32 -i -485929922 -p 1 -x {30.0 32.0 -1 ----- null}
104
105 + -t 38.000000 -s 32 -d 30 -i -1617285483 -p 1 -x {32.0 30.0 -1 ----- null}
106 + -t 38.000000 -s 22 -d 20 -i -635486533 -p 1 -x {22.0 20.0 -1 ----- null}
107 + -t 38.000000 -s 12 -d 10 -i 1435829436 -p 1 -x {12.0 10.0 -1 ----- null}
108 - -t 38.050000 -s 12 -d 10 -i 1435829436 -p 1 -x {12.0 10.0 -1 ----- null}
109 - -t 38.050000 -s 22 -d 20 -i -635486533 -p 1 -x {22.0 20.0 -1 ----- null}
110 - -t 38.050000 -s 32 -d 30 -i -1617285483 -p 1 -x {32.0 30.0 -1 ----- null}
111 r -t 38.118000 -s 12 -d 10 -i 1435829436 -p 1 -x {12.0 10.0 -1 ----- null}
112 r -t 38.118000 -s 22 -d 20 -i -635486533 -p 1 -x {22.0 20.0 -1 ----- null}
113 r -t 38.118000 -s 32 -d 30 -i -1617285483 -p 1 -x {32.0 30.0 -1 ----- null}
```

```
114
115 + -t 40.000000 -s 20 -d 22 -i -1198982007 -p 1 -x {20.0 22.0 -1 ----- null}
116 + -t 40.000000 -s 10 -d 12 -i 495803522 -p 1 -x {10.0 12.0 -1 ----- null}
117 + -t 40.000000 -s 30 -d 32 -i -1568826737 -p 1 -x {30.0 32.0 -1 ----- null}
118 - -t 40.050000 -s 10 -d 12 -i 495803522 -p 1 -x {10.0 12.0 -1 ----- null}
119 - -t 40.050000 -s 30 -d 32 -i -1568826737 -p 1 -x {30.0 32.0 -1 ----- null}
120 - -t 40.050000 -s 20 -d 22 -i -1198982007 -p 1 -x {20.0 22.0 -1 ----- null}
121 r -t 40.118000 -s 10 -d 12 -i 495803522 -p 1 -x {10.0 12.0 -1 ----- null}
122 r -t 40.118000 -s 20 -d 22 -i -1198982007 -p 1 -x {20.0 22.0 -1 ----- null}
123 r -t 40.118000 -s 30 -d 32 -i -1568826737 -p 1 -x {30.0 32.0 -1 ----- null}
124
125
126 + -t 42.000000 -s 12 -d 10 -i 1650451679 -p 1 -x {12.0 10.0 -1 ----- null}
127 + -t 42.000000 -s 22 -d 20 -i 1842552936 -p 1 -x {22.0 20.0 -1 ----- null}
128 + -t 42.000000 -s 32 -d 30 -i 1979483249 -p 1 -x {32.0 30.0 -1 ----- null}
129 - -t 42.050000 -s 12 -d 10 -i 1650451679 -p 1 -x {12.0 10.0 -1 ----- null}
130 - -t 42.050000 -s 22 -d 20 -i 1842552936 -p 1 -x {22.0 20.0 -1 ----- null}
131 - -t 42.050000 -s 32 -d 30 -i 1979483249 -p 1 -x {32.0 30.0 -1 ----- null}
132 r -t 42.118000 -s 12 -d 10 -i 1650451679 -p 1 -x {12.0 10.0 -1 ----- null}
133 r -t 42.118000 -s 22 -d 20 -i 1842552936 -p 1 -x {22.0 20.0 -1 ----- null}
134 r -t 42.118000 -s 32 -d 30 -i 1979483249 -p 1 -x {32.0 30.0 -1 ----- null}
135
136
137 + -t 44.000000 -s 10 -d 13 -i 729613270 -p 1 -x {10.0 13.0 -1 ----- null}
138 + -t 44.000000 -s 30 -d 33 -i -1559998173 -p 1 -x {30.0 33.0 -1 ----- null}
139 + -t 44.000000 -s 20 -d 23 -i 1877131614 -p 1 -x {20.0 23.0 -1 ----- null}
140 - -t 44.050000 -s 10 -d 13 -i 729613270 -p 1 -x {10.0 13.0 -1 ----- null}
141 - -t 44.050000 -s 20 -d 23 -i 1877131614 -p 1 -x {20.0 23.0 -1 ----- null}
142 - -t 44.050000 -s 30 -d 33 -i -1559998173 -p 1 -x {30.0 33.0 -1 ----- null}
143 r -t 44.115000 -s 20 -d 23 -i 1877131614 -p 1 -x {20.0 23.0 -1 ----- null}
144 r -t 44.116000 -s 30 -d 33 -i -1559998173 -p 1 -x {30.0 33.0 -1 ----- null}
145 r -t 44.118000 -s 10 -d 13 -i 729613270 -p 1 -x {10.0 13.0 -1 ----- null}
146
147
148 + -t 46.000000 -s 13 -d 10 -i -527244576 -p 1 -x {13.0 10.0 -1 ----- null}
149 + -t 46.000000 -s 33 -d 30 -i 26561732 -p 1 -x {33.0 30.0 -1 ----- null}
150 + -t 46.000000 -s 23 -d 20 -i -1802584457 -p 1 -x {23.0 20.0 -1 ----- null}
151 - -t 46.050000 -s 13 -d 10 -i -527244576 -p 1 -x {13.0 10.0 -1 ----- null}
```

```
152 - -t 46.050000 -s 23 -d 20 -i -1802584457 -p 1 -x {23.0 20.0 -1 ----- null}
153 - -t 46.050000 -s 33 -d 30 -i 26561732 -p 1 -x {33.0 30.0 -1 ----- null}
154 r -t 46.115000 -s 23 -d 20 -i -1802584457 -p 1 -x {23.0 20.0 -1 ----- null}
155 r -t 46.116000 -s 33 -d 30 -i 26561732 -p 1 -x {33.0 30.0 -1 ----- null}
156 r -t 46.118000 -s 13 -d 10 -i -527244576 -p 1 -x {13.0 10.0 -1 ----- null}
157
158
159 + -t 48.000000 -s 10 -d 13 -i 1919659219 -p 1 -x {10.0 13.0 -1 ----- null}
160 + -t 48.000000 -s 30 -d 33 -i 1675202558 -p 1 -x {30.0 33.0 -1 ----- null}
161 + -t 48.000000 -s 20 -d 23 -i -148677836 -p 1 -x {20.0 23.0 -1 ----- null}
162 - -t 48.050000 -s 10 -d 13 -i 1919659219 -p 1 -x {10.0 13.0 -1 ----- null}
163 - -t 48.050000 -s 30 -d 33 -i 1675202558 -p 1 -x {30.0 33.0 -1 ----- null}
164 - -t 48.050000 -s 20 -d 23 -i -148677836 -p 1 -x {20.0 23.0 -1 ----- null}
165 r -t 48.115000 -s 20 -d 23 -i -148677836 -p 1 -x {20.0 23.0 -1 ----- null}
166 r -t 48.116000 -s 30 -d 33 -i 1675202558 -p 1 -x {30.0 33.0 -1 ----- null}
167 r -t 48.118000 -s 10 -d 13 -i 1919659219 -p 1 -x {10.0 13.0 -1 ----- null}
168
169
170 + -t 50.000000 -s 13 -d 10 -i 2043174756 -p 1 -x {13.0 10.0 -1 ----- null}
171 + -t 50.000000 -s 33 -d 30 -i 293715606 -p 1 -x {33.0 30.0 -1 ----- null}
172 + -t 50.000000 -s 23 -d 20 -i 340513313 -p 1 -x {23.0 20.0 -1 ----- null}
173 - -t 50.050000 -s 13 -d 10 -i 2043174756 -p 1 -x {13.0 10.0 -1 ----- null}
174 - -t 50.050000 -s 23 -d 20 -i 340513313 -p 1 -x {23.0 20.0 -1 ----- null}
175 - -t 50.050000 -s 33 -d 30 -i 293715606 -p 1 -x {33.0 30.0 -1 ----- null}
176 r -t 50.115000 -s 23 -d 20 -i 340513313 -p 1 -x {23.0 20.0 -1 ----- null}
177 r -t 50.116000 -s 33 -d 30 -i 293715606 -p 1 -x {33.0 30.0 -1 ----- null}
178 r -t 50.118000 -s 13 -d 10 -i 2043174756 -p 1 -x {13.0 10.0 -1 ----- null}
179
180 + -t 100.000000 -s 11 -d 10 -i -1876651867 -p 1 -x {11.0 10.0 -1 ----- null}
181 + -t 100.000000 -s 31 -d 30 -i 311196498 -p 1 -x {31.0 30.0 -1 ----- null}
182 + -t 100.000000 -s 21 -d 20 -i 1638291177 -p 1 -x {21.0 20.0 -1 ----- null}
183 - -t 100.050000 -s 11 -d 10 -i -1876651867 -p 1 -x {11.0 10.0 -1 ----- null}
184 - -t 100.050000 -s 31 -d 30 -i 311196498 -p 1 -x {31.0 30.0 -1 ----- null}
185 - -t 100.050000 -s 21 -d 20 -i 1638291177 -p 1 -x {21.0 20.0 -1 ----- null}
186 r -t 100.115000 -s 11 -d 10 -i -1876651867 -p 1 -x {11.0 10.0 -1 ----- null}
187 r -t 100.119000 -s 31 -d 30 -i 311196498 -p 1 -x {31.0 30.0 -1 ----- null}
188 r -t 100.119000 -s 21 -d 20 -i 1638291177 -p 1 -x {21.0 20.0 -1 ----- null}
189
```

```

190
191 + -t 102.000000 -s 12 -d 10 -i -2004910960 -p 1 -x {12.0 10.0 -1 ----- null}
192 + -t 102.000000 -s 22 -d 20 -i 2145653276 -p 1 -x {22.0 20.0 -1 ----- null}
193 + -t 102.000000 -s 32 -d 30 -i 1902302409 -p 1 -x {32.0 30.0 -1 ----- null}
194 - -t 102.050000 -s 12 -d 10 -i -2004910960 -p 1 -x {12.0 10.0 -1 ----- null}
195 - -t 102.050000 -s 22 -d 20 -i 2145653276 -p 1 -x {22.0 20.0 -1 ----- null}
196 - -t 102.050000 -s 32 -d 30 -i 1902302409 -p 1 -x {32.0 30.0 -1 ----- null}
197 r -t 102.118000 -s 12 -d 10 -i -2004910960 -p 1 -x {12.0 10.0 -1 ----- null}
198 r -t 102.118000 -s 22 -d 20 -i 2145653276 -p 1 -x {22.0 20.0 -1 ----- null}
199 r -t 102.118000 -s 32 -d 30 -i 1902302409 -p 1 -x {32.0 30.0 -1 ----- null}
200
201 + -t 104.000000 -s 33 -d 30 -i -741809198 -p 1 -x {33.0 30.0 -1 ----- null}
202 + -t 104.000000 -s 23 -d 20 -i 410210238 -p 1 -x {23.0 20.0 -1 ----- null}
203 + -t 104.000000 -s 13 -d 10 -i 1990266780 -p 1 -x {13.0 10.0 -1 ----- null}
204 - -t 104.050000 -s 13 -d 10 -i 1990266780 -p 1 -x {13.0 10.0 -1 ----- null}
205 - -t 104.050000 -s 33 -d 30 -i -741809198 -p 1 -x {33.0 30.0 -1 ----- null}
206 - -t 104.050000 -s 23 -d 20 -i 410210238 -p 1 -x {23.0 20.0 -1 ----- null}
207 r -t 104.115000 -s 23 -d 20 -i 410210238 -p 1 -x {23.0 20.0 -1 ----- null}
208 r -t 104.116000 -s 33 -d 30 -i -741809198 -p 1 -x {33.0 30.0 -1 ----- null}
209 r -t 104.118000 -s 13 -d 10 -i 1990266780 -p 1 -x {13.0 10.0 -1 ----- null}
210
211
212 + -t 118.000000 -s 30 -d 20 -i -383660597 -p 1 -x {30.0 20.0 -1 ----- null}
213 - -t 118.050000 -s 30 -d 20 -i -383660597 -p 1 -x {30.0 20.0 -1 ----- null}
214 r -t 118.441000 -s 30 -d 20 -i -383660597 -p 1 -x {30.0 20.0 -1 ----- null}
215
216 + -t 120.000000 -s 20 -d 10 -i 1657531287 -p 1 -x {20.0 10.0 -1 ----- null}
217 - -t 120.050000 -s 20 -d 10 -i 1657531287 -p 1 -x {20.0 10.0 -1 ----- null}
218 r -t 120.438000 -s 20 -d 10 -i 1657531287 -p 1 -x {20.0 10.0 -1 ----- null}
219
220 + -t 122.000000 -s 10 -d 1 -i 1258999467 -p 1 -x {10.0 1.0 -1 ----- null}
221 - -t 122.050000 -s 10 -d 1 -i 1258999467 -p 1 -x {10.0 1.0 -1 ----- null}
222 r -t 122.436000 -s 10 -d 1 -i 1258999467 -p 1 -x {10.0 1.0 -1 ----- null}

```

Results of energy consumption of nodes (in cycle 1) are as follows:

```

1 122000|INFO|cbtdma@72:println|node : 23
2 cycle : 1 -----> send_bytes_head : 0,

```

```
3 send_bytes_normal : 33, recv_bytes : 22,
4 send_energy_normal : 1.7435001E-6,
5 send_energy_head : 0.0,
6 send_energy : 1.7435001E-6,
7 receive_energy : 7.333333E-8
8
9 122000|INFO|cbtdma@64: println | node : 21
10 cycle : 1 ——> send_bytes_head : 0,
11 send_bytes_normal : 33, recv_bytes : 22,
12 send_energy_normal : 1.7435001E-6,
13 send_energy_head : 0.0,
14 send_energy : 1.7435001E-6,
15 receive_energy : 7.333333E-8
16
17 122000|INFO|cbtdma@60: println | node : 20
18 cycle : 1 ——> send_bytes_head : 55,
19 send_bytes_normal : 66, recv_bytes : 154,
20 send_energy_normal : 3.4870002E-6,
21 send_energy_head : 0.0018333333,
22 send_energy : 0.0018368203,
23 receive_energy : 5.133333E-7
24
25 122000|INFO|cbtdma@68: println | node : 22
26 cycle : 1 ——> send_bytes_head : 0,
27 send_bytes_normal : 33, recv_bytes : 22,
28 send_energy_normal : 1.7435001E-6,
29 send_energy_head : 0.0,
30 send_energy : 1.7435001E-6,
31 receive_energy : 7.333333E-8
32
33 122000|INFO|cbtdma@20: println | node : 10
34 cycle : 1 ——> send_bytes_head : 55,
35 send_bytes_normal : 66, recv_bytes : 154,
36 send_energy_normal : 3.4870002E-6,
37 send_energy_head : 0.0018333333,
38 send_energy : 0.0018368203,
39 receive_energy : 5.133333E-7
40
```

```
41 122000|INFO|cbtdma@32:println|node : 13
42 cycle : 1 ——> send_bytes_head : 0,
43 send_bytes_normal : 33, recv_bytes : 22,
44 send_energy_normal : 1.7435001E-6,
45 send_energy_head : 0.0,
46 send_energy : 1.7435001E-6,
47 receive_energy : 7.3333333E-8
48
49 122000|INFO|cbtdma@100:println|node : 30
50 cycle : 1 ——> send_bytes_head : 33,
51 send_bytes_normal : 66, recv_bytes : 121,
52 send_energy_normal : 3.4870002E-6,
53 send_energy_head : 0.0011,
54 send_energy : 0.001103487,
55 receive_energy : 4.0333333E-7
56
57 122000|INFO|cbtdma@24:println|node : 11
58 cycle : 1 ——> send_bytes_head : 0,
59 send_bytes_normal : 33, recv_bytes : 22,
60 send_energy_normal : 1.7435001E-6,
61 send_energy_head : 0.0,
62 send_energy : 1.7435001E-6,
63 receive_energy : 7.3333333E-8
64
65 122000|INFO|cbtdma@108:println|node : 32
66 cycle : 1 ——> send_bytes_head : 0,
67 send_bytes_normal : 33, recv_bytes : 22,
68 send_energy_normal : 1.7435001E-6,
69 send_energy_head : 0.0,
70 send_energy : 1.7435001E-6,
71 receive_energy : 7.3333333E-8
72
73 122000|INFO|cbtdma@28:println|node : 12
74 cycle : 1 ——> send_bytes_head : 0,
75 send_bytes_normal : 33, recv_bytes : 22,
76 send_energy_normal : 1.7435001E-6,
77 send_energy_head : 0.0,
78 send_energy : 1.7435001E-6,
```

```
79 receive_energy : 7.333333E-8
80
81 122000|INFO|cbtdma@112:println|node : 33
82 cycle : 1 ——> send_bytes_head : 0,
83 send_bytes_normal : 33, recv_bytes : 22,
84 send_energy_normal : 1.7435001E-6,
85 send_energy_head : 0.0,
86 send_energy : 1.7435001E-6,
87 receive_energy : 7.333333E-8
88
89 122000|INFO|cbtdma@104:println|node : 31
90 cycle : 1 ——> send_bytes_head : 0,
91 send_bytes_normal : 33, recv_bytes : 22,
92 send_energy_normal : 1.7435001E-6,
93 send_energy_head : 0.0,
94 send_energy : 1.7435001E-6,
95 receive_energy : 7.333333E-8
```


Bibliography

- [1] “National Oceanic and Atmospheric Administration,” <http://oceanexplorer.noaa.gov/facts/facts.html>, accessed: 2016-05-09.
- [2] E. M. Sozer, M. Stojanovic, and J. G. Proakis, “Underwater Acoustic Networks,” *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 72–83, 2000.
- [3] J. G. Proakis, E. M. Sozer, J. A. Rice, and M. Stojanovic, “Shallow Water Acoustic Networks,” *IEEE Communications Magazine*, vol. 39, no. 11, pp. 114–119, 2001.
- [4] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater Acoustic Sensor Networks: Research Challenges,” *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.
- [5] M. Chitre, S. Shahabudeen, and M. Stojanovic, “Underwater Acoustic Communications and Networking: Recent Advances and Future Challenges,” *Marine technology society journal*, vol. 42, no. 1, pp. 103–116, 2008.
- [6] M. Chitre, S. Shahabudeen, L. Freitag, and M. Stojanovic, “Recent Advances in Underwater Acoustic Communications & Networking,” in *IEEE OCEANS*, 2008, pp. 1–10.
- [7] I. F. Akyildiz, D. Pompili, and T. Melodia, “State-of-the-art in Protocol Research for Underwater Acoustic Sensor Networks,” in *Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 7–16.
- [8] J. Heidemann, M. Stojanovic, and M. Zorzi, “Underwater Sensor Networks: Applications, Advances and Challenges,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 158–175, 2012.

- [9] J. L. Hill, "System Architecture for Wireless Sensor Networks," Ph.D. dissertation, University of California, Berkeley, 2003.
- [10] T. Arampatzis, J. Lygeros, and S. Manesis, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," in *Proceedings of the IEEE International Symposium on Intelligent Control and Mediterrean Conference on Control and Automation*, 2005, pp. 719–724.
- [11] C.-Y. Chong and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.
- [13] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov, "System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring," *Journal of mobile multimedia*, vol. 1, no. 4, pp. 307–326, 2006.
- [14] A. Tagarakis, V. Liakos, L. Perlepes, S. Fountas, and T. Gemtos, "Wireless Sensor Network for Precision Agriculture," in *IEEE 15th Panhellenic Conference on Informatics (PCI)*, 2011, pp. 397–402.
- [15] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application," in *ACM Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 214–226.
- [16] M. Castillo-Effer, D. H. Quintela, W. Moreno, R. Jordan, and W. Westhoff, "Wireless Sensor Networks for Flash-flood Alerting," in *Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems*, 2004, pp. 142–146.
- [17] S. Rajasegarar, J. Gubbi, O. Bondarenko, S. Kininmonth, S. Marusic, S. Bainbridge, I. Atkinson, and M. Palaniswami, "Sensor Network Implementation Challenges in the Great Barrier Reef Marine Environment," in *Proceedings of the ICT-Mobile Summit Conference*, 2008, pp. 1–9.

- [18] K. Liu, Z. Yang, M. Li, Z. Guo, Y. Guo, F. Hong, X. Yang, Y. He, Y. Feng, and Y. Liu, "OceanSense: Monitoring the Sea with Wireless Sensor Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, no. 2, pp. 7–9, 2010.
- [19] L. Lanbo, Z. Shengli, and C. Jun-Hong, "Prospects and Problems of Wireless Communication for Underwater Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 8, pp. 977–994, 2008.
- [20] Y. Wang, Y. Liu, and Z. Guo, "Three-Dimensional Ocean Sensor Networks: a Survey," *Journal of Ocean University of China*, vol. 11, no. 4, pp. 436–450, 2012.
- [21] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb, "Autonomous Oceanographic Sampling Networks," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.
- [22] R. H. Weisberg, A. Barth, A. Alvera-Azcárate, and L. Zheng, "A Coordinated Coastal Ocean Observing and Modeling System for the West Florida Continental Shelf," *Harmful Algae*, vol. 8, no. 4, pp. 585–597, 2009.
- [23] H. Manum and M. Schmid, "Monitoring in a Harsh Environment," *Control & Automation*, vol. 18, no. 5, pp. 22–27, 2007.
- [24] N. Mohamed, I. Jawhar, J. Al-Jaroodi, and L. Zhang, "Sensor Network Architectures for Monitoring Underwater Pipelines," *Sensors*, vol. 11, no. 11, pp. 10 738–10 764, 2011.
- [25] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless Sensor Networks for Underwater Surveillance Systems," *Ad Hoc Networks*, vol. 4, no. 4, pp. 431–446, 2006.
- [26] S. J. Barr, J. Wang, and B. Liu, "An Efficient Method for Constructing Underwater Sensor Barriers," *Journal of Communications*, vol. 6, no. 5, pp. 370–383, 2011.
- [27] J. A. Rice, "US Navy Seaweb Development," in *ACM Proceedings of the second workshop on Underwater networks*, 2007, pp. 3–4.
- [28] R. Urick, *Principles Of Underwater Sound*. McGraw-Hill Ryerson Limited, 1983.
- [29] M. C. Domingo, "Overview of Channel Models for Underwater Wireless Communication Networks," *Phys. Commun.*, vol. 1, no. 3, pp. 163–182, 2008.

- [30] D. Pompili and I. F. Akyildiz, "Overview of Networking Protocols for Underwater Wireless Communications," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 97–102, 2009.
- [31] M. Stojanovic, "On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, 2007.
- [32] —, "Acoustic (Underwater) Communications," *Encyclopedia of Telecommunications*, 2003.
- [33] J. Preisig, "Acoustic Propagation Considerations for Underwater Acoustic Communications Network Development," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 2–10, 2007.
- [34] D. Brady and J. C. Preisig, "Underwater Acoustic Communications," *Wireless Communications: Signal Processing Perspectives*, vol. 8, pp. 330–379, 1998.
- [35] M. Stojanovic, "Underwater Acoustic Communications: Design Considerations on the Physical Layer," in *Fifth Annual Conference on Wireless on Demand Network Systems and Services*, 2008, pp. 1–10.
- [36] M. Neugebauer, J. Ploennigs, and K. Kabitzsch, "Evaluation of Energy costs for Single Hop vs. Multi Hop with Respect to Topology Parameters," in *IEEE International Workshop on Factory Communication Systems*, 2006, pp. 175–182.
- [37] A. A. Syed, W. Ye, J. Heidemann, and B. Krishnamachari, "Understanding Spatio-Temporal Uncertainty in Medium Access with ALOHA Protocols," in *ACM Proceedings of the second workshop on Underwater networks*, 2007, pp. 41–48.
- [38] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: a Reliable Transport Protocol for Wireless Sensor Networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002, pp. 1–11.
- [39] J. Bachrach and C. Taylor, "Localization in Sensor Networks," *Handbook of sensor networks: Algorithms and Architectures*, vol. 1, 2005.

- [40] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync Protocol for Sensor Networks," in *ACM Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 138–149.
- [41] T. Kwon and M. Gerla, "Clustering with Power Control," in *MILCOM*, 1999, pp. 1424–1428.
- [42] I. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, Inc., 2010.
- [43] D. Pompili, T. Melodia, and I. F. Akyildiz, "Three-dimensional and Two-dimensional Deployment Analysis for Underwater Acoustic Sensor Networks," *Ad Hoc Networks*, vol. 7, no. 4, pp. 778–790, 2009.
- [44] S. C. Dhongdi, K. Anupama, and L. J. Gudino, "Review of Protocol Stack Development of Underwater Acoustic Sensor Network (UASN)," in *IEEE Underwater Technology (UT)*, 2015, pp. 1–17.
- [45] J. Llor and M. P. Malumbres, "Underwater Wireless Sensor Networks: How Do Acoustic Propagation Models Impact the Performance of Higher-Level Protocols?" *Sensors*, vol. 12, no. 2, pp. 1312–1335, 2012.
- [46] S. Singh, S. E. Webster, L. Freitag, L. L. Whitcomb, K. Ball, J. Bailey, and C. Taylor, "Acoustic Communication Performance of the WHOI Micro-modem in Sea Trials of the Nereus Vehicle to 11,000 m Depth," in *IEEE OCEANS*, 2009, pp. 1–6.
- [47] D. Green, "Acoustic Modems, Navigation Aids, and Networks for Undersea Operations," in *IEEE OCEANS*, 2010, pp. 1–6.
- [48] M. Stojanovic, J. Catipovic, and J. G. Proakis, "Adaptive Multichannel Combining and Equalization for Underwater Acoustic Communications," *The Journal of the Acoustical Society of America*, vol. 94, no. 3, pp. 1621–1631, 1993.
- [49] M. Stojanovic, "Low Complexity OFDM Detector for Underwater Acoustic Channels," in *IEEE OCEANS*, 2006, pp. 1–6.
- [50] A. C. Singer, J. K. Nelson, and S. S. Kozat, "Signal Processing for Underwater Acoustic Communications," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 90–96, 2009.

- [51] P. C. Carrascosa and M. Stojanovic, "Adaptive Channel Estimation and Data Detection for Underwater Acoustic MIMO-OFDM Systems," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 3, pp. 635–646, 2010.
- [52] "LinkQuest, Inc., "Underwater Acoustic Modems";" <http://www.link-quest.com/html/models1.htm>, accessed: 2014-04-08.
- [53] "DSPCOMM, "Underwater Wireless Modem";" http://www.dspcomm.com/products_aquacomm.html, accessed: 2014-04-08.
- [54] "Tritech International, "Micron Data Modem";" <http://www.tritech.co.uk/product/micron-data-modem>, accessed: 2014-04-08.
- [55] R. Jurdak, C. V. Lopes, and P. Baldi, "A Survey, Classification and Comparative Analysis of Medium Access Control Protocols for Ad Hoc Networks," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 1, pp. 2–16, 2004.
- [56] S. Kumar, V. S. Raghavan, and J. Deng, "Medium Access Control Protocols for Ad Hoc Wireless Networks: A Survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326–358, 2006.
- [57] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: a Survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.
- [58] K. Kredo II and P. Mohapatra, "Medium Access Control in Wireless Sensor Networks," *Computer Networks*, vol. 51, no. 4, pp. 961–994, 2007.
- [59] K. Kredo, P. Djukic, and P. Mohapatra, "STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol," in *IEEE INFOCOM*, 2009, pp. 2961–2965.
- [60] Y.-D. Chen, C.-Y. Lien, S.-W. Chuang, and K.-P. Shih, "DSSS: a TDMA-based MAC Protocol with Dynamic Slot Scheduling Strategy for Underwater Acoustic Sensor Networks," in *IEEE OCEANS*, 2011, pp. 1–6.
- [61] Y. Ma, Z. Guo, Y. Feng, M. Jiang, and G. Feng, "C-MAC: A TDMA-based MAC Protocol for Underwater Acoustic Sensor Networks," in *IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing, (NSWCTC)*, 2009, pp. 728–731.

- [62] G. Zhongwen, L. Zhengbao, and H. Feng, "USS-TDMA: Self-stabilizing TDMA Algorithm for Underwater Wireless Sensor Network," in *IEEE International Conference on Computer Engineering and Technology*, 2009, pp. 578–582.
- [63] Y. Zhong, J. Huang, and J. Han, "A TDMA MAC Protocol for Underwater Acoustic Sensor Networks," in *IEEE Youth Conference on Information, Computing and Telecommunication, (YC-ICT)*, 2009, pp. 534–537.
- [64] P. Anjani and M. Chitre, "Design and Implementation of Super-TDMA: A MAC Protocol Exploiting Large Propagation Delays for Underwater Acoustic Networks," in *ACM Proceedings of the 10th International Conference on Underwater Networks & Systems*, 2015, pp. 1–8.
- [65] —, "Experimental Demonstration of Super-TDMA: A MAC Protocol Exploiting Large Propagation Delays in Underwater Acoustic Networks," in *Underwater Communications Networking*, 2016, pp. 1–5.
- [66] H.-X. Tan and W. K. G. Seah, "Distributed CDMA-Based MAC Protocol for Underwater Sensor Networks," in *32nd IEEE Conference on Local Computer Networks*, 2007, pp. 26–36.
- [67] D. Pompili, T. Melodia, and I. F. Akyildiz, "A Distributed CDMA Medium Access Control for Underwater Acoustic Sensor Networks," in *Proc. of Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2007, pp. 63–70.
- [68] —, "A CDMA-Based Medium Access Control for Underwater Acoustic Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1899–1909, 2009.
- [69] X. Guo, M. R. Frater, and M. J. Ryan, "A Propagation-Delay-Tolerant Collision Avoidance Protocol for Underwater Acoustic Sensor Networks," in *IEEE OCEANS*, 2007, pp. 1–6.
- [70] —, "Design of a Propagation-Delay-Tolerant MAC Protocol for Underwater Acoustic Sensor Networks," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 2, pp. 170–180, 2009.

- [71] B. Peleato and M. Stojanovic, "Distance Aware Collision Avoidance Protocol for Ad-Hoc Underwater Acoustic Sensor Networks," *IEEE Communications Letters*, vol. 11, no. 12, pp. 1025–1027, 2007.
- [72] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC protocol for Underwater Acoustic Networks," in *IEEE OCEANS*, 2007, pp. 1–7.
- [73] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "MACA-MN: A MACA-based MAC Protocol for Underwater Acoustic Networks with Packet Train for Multiple Neighbors," in *IEEE Vehicular Technology Conference*, 2008, pp. 46–50.
- [74] H.-H. Ng, W.-S. Soh, and M. Motani, "MACA-U: A Media Access Protocol for Underwater Acoustic Networks," in *IEEE Global Telecommunications Conference*, 2008, pp. 1–5.
- [75] W. Lin, E. Cheng, and F. Yuan, "A MACA-Based MAC Protocol for Underwater Acoustic Sensor Networks," *Journal of Communications*, vol. 6, no. 2, pp. 179–184, 2011.
- [76] W.-H. Liao and C.-C. Huang, "SF-MAC: A Spatially Fair MAC Protocol for Underwater Acoustic Sensor Networks," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1686–1694, 2012.
- [77] H.-H. Ng, W.-S. Soh, and M. Motani, "An Underwater Acoustic MAC Protocol Using Reverse Opportunistic Packet Appending," *Computer Networks*, vol. 57, no. 14, pp. 2733–2751, 2013.
- [78] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "ALOHA-based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks," in *26th IEEE International Conference on Computer Communications*, 2007, pp. 2271–2275.
- [79] Y. Zhou, K. Chen, J. He, and H. Guan, "Enhanced Slotted ALOHA Protocols for Underwater Sensor Networks with Large Propagation Delay," in *73rd IEEE Vehicular Technology Conference*, 2011, pp. 1–5.
- [80] J. Ahn and B. Krishnamachari, "Performance of a Propagation Delay Tolerant ALOHA Protocol for Underwater Wireless Networks," in *Distributed Computing in Sensor Systems*, 2008, pp. 1–16.

- [81] S. De, P. Mandal, and S. S. Chakraborty, "On the Characterization of ALOHA in Underwater Wireless Networks," *Mathematical and Computer Modelling*, vol. 53, no. 11, pp. 2093–2107, 2011.
- [82] N. Yao, Z. Peng, M. Zuba, and J.-H. Cui, "Improving ALOHA via Backoff Tuning in Underwater Sensor Networks," in *6th IEEE International ICST Conference on Communications and Networking in China (CHINACOM)*, 2011, pp. 1038–1043.
- [83] P. Xie and J.-H. Cui, "R-MAC: An Energy-Efficient MAC protocol for Underwater Sensor Networks," in *IEEE International Conference on Wireless Algorithms, Systems and Applications*, 2007, pp. 187–198.
- [84] L. T. Tracy and S. Roy, "A Reservation MAC Protocol for Ad-Hoc Underwater Acoustic Sensor Networks," in *Proceedings of the third ACM international workshop on Underwater Networks*, 2008, pp. 95–98.
- [85] G. Fan, H. Chen, L. Xie, and K. Wang, "A Hybrid Reservation-based MAC Protocol for Underwater Acoustic Sensor Networks," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1178–1192, 2013.
- [86] Y. Zhu, R. Z. Zhou, J. P. Zheng, and J.-H. Cui, "An Efficient Geo-Routing Aware MAC protocol for Underwater Acoustic Networks," in *International Conference on Ad Hoc Networks*, 2010, pp. 185–200.
- [87] M. K. Park and V. Rodoplu, "UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 3, pp. 710–720, 2007.
- [88] A. A. Syed, W. Ye, and J. Heidemann, "T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks," in *27th IEEE Conference on Computer Communications*, 2008, pp. 789–797.
- [89] N. Chirdchoo, W.-s. Soh, and K. Chua, "RIPT: A Receiver-Initiated Reservation-based Protocol for Underwater Acoustic Networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1744–1753, 2008.

- [90] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A Review of Routing Protocols for Mobile Ad Hoc Networks," *Ad hoc networks*, vol. 2, no. 1, pp. 1–22, 2004.
- [91] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad hoc networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [92] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *ACM SIGCOMM Computer Communication Review*, 1994, pp. 234–244.
- [93] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," in *Proceedings of IEEE International Multi Topic Conference: Technology for the 21st Century*, 2001, pp. 62–68.
- [94] C. Perkins and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," Internet Draft, draft-ietf-manet-aodv-00.txt, Tech. Rep., 1997.
- [95] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile computing*, 1996, pp. 153–181.
- [96] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Wireless networks*, vol. 7, no. 6, pp. 609–616, 2001.
- [97] T. Melodia, D. Pompili, and I. F. Akyildiz, "On the Interdependence of Distributed Topology Control and Geographical Routing in Ad Hoc and Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 520–532, 2005.
- [98] P. Xie, J.-H. Cui, and L. Lao, "VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks," in *Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems, (NETWORKING)*, 2006, pp. 1216–1221.
- [99] N. Nicolaou, A. See, P. Xie, J.-H. Cui, and D. Maggiorini, "Improving the Robustness of Location-Based Routing for Underwater Sensor Networks," in *IEEE OCEANS*, 2007, pp. 1–6.

- [100] J. M. Jornet, M. Stojanovic, and M. Zorzi, “Focused Beam Routing Protocol for Underwater Acoustic Networks,” in *Proceedings of the third ACM international workshop on Underwater Networks*, 2008, pp. 75–82.
- [101] D. Pompili, T. Melodia, and I. F. Akyildiz, “Routing Algorithms for Delay-Insensitive and Delay-Sensitive Applications in Underwater Sensor Networks,” in *Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006, pp. 298–309.
- [102] D. Hwang and D. Kim, “DFR: Directional Flooding-based Routing Protocol for Underwater Sensor Networks,” in *IEEE OCEANS*, 2008, pp. 1–7.
- [103] U. Lee, J. Kong, M. Gerla, J.-S. Park, and E. Magistretti, “Time-Critical Underwater Sensor Diffusion with No Proactive Exchanges and Negligible Reactive Floods,” *Ad Hoc Networks*, vol. 5, no. 6, pp. 943–958, 2007.
- [104] M. C. Domingo and R. Prior, “Energy Analysis of Routing Protocols for Underwater Wireless Sensor Networks,” *Computer Communications*, vol. 31, no. 6, pp. 1227–1238, 2008.
- [105] ———, “Design and Analysis of a GPS-free Routing Protocol for Underwater Wireless Sensor Networks in Deep Water,” in *IEEE International Conference on Sensor Technologies and Applications*, 2007, pp. 215–220.
- [106] N. Li, J.-F. Martínez, J. M. Meneses Chaus, and M. Eckert, “A Survey on Underwater Acoustic Sensor Network Routing Protocols,” *Sensors*, vol. 16, no. 3, p. 414, 2016.
- [107] V. Chandrasekhar, W. K. Seah, Y. S. Choo, and H. V. Ee, “Localization in Underwater Sensor Networks: Survey and Challenges,” in *Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 33–40.
- [108] S. Gopi, G. Kannan, D. Chander, U. B. Desai, and S. Merchant, “PULRP: Path Unaware Layered Routing Protocol for Underwater Sensor Networks,” in *IEEE International Conference on Communications*, 2008, pp. 3141–3145.
- [109] S. Gopi, K. Govindan, D. Chander, U. B. Desai, and S. Merchant, “E-PULRP: Energy Optimized Path Unaware Layered Routing Protocol for Underwater Sensor Networks,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3391–3401, 2010.

- [110] U. Lee, P. Wang, Y. Noh, F. Vieira, M. Gerla, and J.-H. Cui, "Pressure Routing for Underwater Sensor Networks," in *IEEE INFOCOM*, 2010, pp. 1–9.
- [111] M. Al-Bzoor, Y. Zhu, J. Liu, A. Reda, J.-H. Cui, and S. Rajasekaran, "Adaptive Power Controlled Routing for Underwater Sensor Networks," in *Wireless Algorithms, Systems, and Applications*, 2012, pp. 549–560.
- [112] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, "VAPR: Void-Aware Pressure Routing for Underwater Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 895–908, 2013.
- [113] Y.-S. Chen and Y.-W. Lin, "Mobicast Routing Protocol for Underwater Sensor Networks," *IEEE Sensors Journal*, vol. 13, no. 2, pp. 737–749, 2013.
- [114] S. Climent, A. Sanchez, J. V. Capella, N. Meratnia, and J. J. Serrano, "Underwater Acoustic Wireless Sensor Networks: Advances and Future Trends in Physical, MAC and Routing Layers," *Sensors*, vol. 14, no. 1, pp. 795–833, 2014.
- [115] J. W. Lee, J. Y. Cheon, and H.-S. Cho, "A Cooperative ARQ Scheme in Underwater Acoustic Sensor Networks," in *IEEE OCEANS*, 2010, pp. 1–5.
- [116] H. Tan, W.-G. Seah, and L. Doyle, "A Multi-hop ARQ Protocol for Underwater Acoustic Networks," in *IEEE OCEANS*, 2007, pp. 1–6.
- [117] J.-W. Lee, J.-P. Kim, J.-H. Lee, Y. S. Jang, K. C. Dho, K. Son, and H.-S. Cho, "An Improved ARQ Scheme in Underwater Acoustic Sensor Networks," in *IEEE OCEANS*, 2008, pp. 1–5.
- [118] B. Liu, F. Garcin, F. Ren, and C. Lin, "A Study of Forward Error Correction Schemes for Reliable Transport in Underwater Sensor Networks," in *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2008, pp. 197–205.
- [119] S. Cai, Z. Gao, D. Yang, J. Zhao, and Y. Zhao, "IPool-ADELIN: An Extended ADELIN Based on IPool Node for Reliable Transport of Underwater Acoustic Sensor Networks," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1435–1442, 2013.

- [120] D. Shin and D. Kim, "FRT: Fast and Reliable Transport Protocol for Underwater Wireless Sensor Networks," in *IEEE Asia-Pacific Services Computing Conference*, 2009, pp. 402–407.
- [121] B. Liu, H. Chen, X. Lei, F. Ren, and K. Sezaki, "Internode Distance-Based Redundancy Reliable Transport in Underwater Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 1–16, 2010.
- [122] S. Cai, Z. Gao, D. Yang, and N. Yao, "A Network Coding Based Protocol for Reliable Data Transfer in Underwater Acoustic Sensor," *Ad Hoc Networks*, vol. 11, no. 5, pp. 1603–1609, 2013.
- [123] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [124] P. Xie, Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, "SDRT: a Reliable Data Transport Protocol for Underwater Sensor Networks," *Ad Hoc Networks*, vol. 8, no. 7, pp. 708–722, 2010.
- [125] Z. Zhou, H. Mo, Y. Zhu, Z. Peng, J. Huang, and J.-H. Cui, "Fountain Code Based Adaptive Multi-hop Reliable Data Transfer for Underwater Acoustic Networks," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 6396–6400.
- [126] H. Mo, A. C. Mingir, H. Alhumyani, Y. Albayram, and J.-H. Cui, "UW-HARQ: An Underwater Hybrid ARQ Scheme: Design, Implementation and Initial Test," in *IEEE OCEANS*, 2012, pp. 1–5.
- [127] M. C. Domingo, "Marine Communities Based Congestion Control in Underwater Wireless Sensor Networks," *Information Sciences*, vol. 228, pp. 203–221, 2013.
- [128] Z. Jiang, "Underwater Acoustic Networks-Issues and Solutions," *International journal of intelligent control and systems*, vol. 13, no. 3, pp. 152–161, 2008.
- [129] M. Erol, L. F. Vieira, and M. Gerla, "Localization with Dive'N' Rise (DNR) Beacons for Underwater Acoustic Sensor Networks," in *ACM Proceedings of the second workshop on Underwater networks*, 2007, pp. 97–100.
- [130] Z. Zhou, J.-H. Cui, and S. Zhou, "Efficient Localization for Large-Scale Underwater Sensor Networks," *Ad Hoc Networks*, vol. 8, no. 3, pp. 267–279, 2010.

- [131] H. Luo, Z. Guo, W. Dong, F. Hong, and Y. Zhao, "LDB: Localization with Directional Beacons for Sparse 3D Underwater Acoustic Sensor Networks," *Journal of networks*, vol. 5, no. 1, pp. 28–38, 2010.
- [132] M. Erol-Kantarci, H. T. Mouftah, and S. Oktug, "A Survey of Architectures and Localization Techniques for Underwater Acoustic Sensor Networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 487–502, 2011.
- [133] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147–163, 2002.
- [134] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The Flooding Time Synchronization Protocol," in *ACM Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 39–49.
- [135] J. Van Greunen and J. Rabaey, "Lightweight Time Synchronization for Sensor Networks," in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003, pp. 11–19.
- [136] A. A. Syed and J. S. Heidemann, "Time Synchronization for High Latency Acoustic Networks," in *IEEE INFOCOM*, 2006, pp. 1–6.
- [137] C. Tian, H. Jiang, X. Liu, X. Wang, W. Liu, and Y. Wang, "Tri-message: a Lightweight Time Synchronization Protocol for High Latency and Resource-Constrained Networks," in *IEEE International Conference on Communications (ICC)*, 2009, pp. 1–5.
- [138] C. Sun, F. Yang, L. Ding, L. Qian, and C. Zhi, "Multi-Hop Time Synchronization for Underwater Acoustic Networks," in *IEEE OCEANS*, 2012, pp. 1–7.
- [139] C. Lu, S. Wang, and M. Tan, "A Time Synchronization Method for Underwater Wireless Sensor Networks," in *IEEE Chinese Control and Decision Conference*, 2009, pp. 4305–4310.
- [140] Z. Tian, C. Chen, Y. Xia, and X. Guan, "TSLA: An Energy Efficient Time Synchronization and Localization Algorithm for Underwater Acoustic Sensor Networks," in *IEEE*

- International Conference on Wireless Communications & Signal Processing (WCSP)*, 2015, pp. 1–6.
- [141] F. Lu, D. Mirza, and C. Schurgers, “D-Sync: Doppler-Based Time Synchronization for Mobile Underwater Sensor Networks,” in *Proceedings of the Fifth ACM International Workshop on UnderWater Networks*, 2010, pp. 1–8.
- [142] J. Liu, Z. Wang, Z. Peng, M. Zuba, J.-H. Cui, and S. Zhou, “TSMU: A Time Synchronization Scheme for Mobile Underwater Sensor Networks,” in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–6.
- [143] J. Liu, Z. Zhou, Z. Peng, J.-H. Cui, M. Zuba, and L. Fiondella, “Mobi-Sync: Efficient Time Synchronization for Mobile Underwater Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 406–416, 2013.
- [144] Z. Li, Z. Guo, F. Hong, and L. Hong, “E2DTS: An Energy Efficiency Distributed Time Synchronization Algorithm for Underwater Acoustic Mobile Sensor Networks,” *Ad Hoc Networks*, vol. 11, no. 4, pp. 1372–1380, 2013.
- [145] P. Wang, C. Li, and J. Zheng, “Distributed Minimum-Cost Clustering Protocol for UnderWater Sensor Networks (UWSNs),” in *IEEE International Conference on Communications (ICC)*, 2007, pp. 3510–3515.
- [146] K. Anupama, A. Sasidharan, and S. Vadlamani, “A Location-Based Clustering Algorithm for Data Gathering in 3D Underwater Wireless Sensor Networks,” in *International Symposium on Telecommunications*, 2008, pp. 343–348.
- [147] Y. Li, Y. Wang, Y. Ju, and R. He, “Energy Efficient Cluster Formulation Protocols in Clustered Underwater Acoustic Sensor Networks,” in *7th IEEE International Conference on Biomedical Engineering and Informatics*, 2014, pp. 923–928.
- [148] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An Application-specific Protocol Architecture for Wireless Microsensor Networks,” *IEEE Transactions on wireless communications*, vol. 1, no. 4, pp. 660–670, 2002.

- [149] D. Pompili and I. F. Akyildiz, "A Cross-Layer Communication Solution for Multimedia Applications in Underwater Acoustic Sensor Networks," in *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 275–284.
- [150] J. M. Jornet, M. Stojanovic, and M. Zorzi, "On Joint Frequency and Power Allocation in a Cross-Layer Protocol for Underwater Acoustic Networks," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 4, pp. 936–947, 2010.
- [151] L.-C. Kuo and T. Melodia, "Tier-Based Underwater Acoustic Routing for Applications with Reliability and Delay Constraints," in *Proceedings of IEEE International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN)*, 2011, pp. 1–6.
- [152] S. Basagni, C. Petrioli, R. Petrocchia, and D. Spaccin, "Channel-Aware Routing for Underwater Wireless Networks," in *IEEE OCEANS*, 2012, pp. 1–9.
- [153] D. Pompili and I. F. Akyildiz, "A Multimedia Cross-Layer Protocol for Underwater Acoustic Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2924–2933, 2010.
- [154] R. Szwedczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a Sensor Network Expedition," in *European Workshop on Wireless Sensor Networks*, 2004, pp. 307–322.
- [155] J. I. Choi, J. W. Lee, M. Wachs, and P. Levis, "Opening the Sensornet Black Box," *ACM SIGBED Review*, vol. 4, no. 3, pp. 13–18, 2007.
- [156] R. Zurawski, *Networked Embedded Systems*. CRC press, 2009.
- [157] "Network Simulator-2," <http://www.isi.edu/nsnam/ns/>, accessed: 2015-08-02.
- [158] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," *UCLA Computer Science Department Technical Report*, vol. 990027, no. 1999, p. 213, 1999.
- [159] "Castalia," <https://castalia.forge.nicta.com.au/index.php/en/>, accessed: 2015-03-09.
- [160] P. Levis and N. Lee, "Tossim: A Simulator for TinyOS Networks," *UC Berkeley*, vol. 24, 2003.

- [161] G. Werner-Allen, P. Swieskowski, and M. Welsh, "Motelab: A Wireless Sensor Network Testbed," in *Proceedings of the 4th IEEE International Symposium on Information Processing in Sensor Networks*, 2005, pp. 1–6.
- [162] M. Dyer, J. Beutel, T. Kalt, P. Oehen, L. Thiele, K. Martin, and P. Blum, "Deployment Support Network," in *European Conference on Wireless Sensor Networks*, 2007, pp. 195–211.
- [163] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, "EmStar: A Software Environment for Developing and Deploying Wireless Sensor Networks," in *USENIX Annual Technical Conference*, 2004, pp. 283–296.
- [164] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Demo Abstract: Cross-level Simulation in COOJA," in *Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications*, 2006.
- [165] "UWSN Simulator - Aqua-sim," <http://obinet.engr.uconn.edu/wiki/index.php/Aqua-Sim>, accessed: 2015-02-12.
- [166] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. Cui, Z. Shi, Y. Fei, and S. Zhou, "Aqua-Sim: an NS-2 Based Simulator for Underwater Sensor Networks," in *IEEE OCEANS*, 2009, pp. 1–7.
- [167] "Underwater Sensor Network Lab, Research Laboratory, UCONN," <http://uwsn.engr.uconn.edu/>, accessed: 2015-05-10.
- [168] "DESERT Underwater," <http://nautilus.dei.unipd.it/desert-underwater>, accessed: 2015-05-10.
- [169] R. Masiero, S. Azad, F. Favaro, M. Petrani, G. Toso, F. Guerra, P. Casari, and M. Zorzi, "DESERT Underwater: an NS-Miracle-based framework to DEsign, Simulate, Emulate and Realize Test-beds for Underwater network protocols," in *IEEE OCEANS*, 2012, pp. 1–10.
- [170] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "NS2-MIRACLE: A Modular Framework for Multi-technology and Cross-layer Support in Network Simulator 2," in *Proceedings of the 2Nd International Conference on Performance Evaluation Methodologies and Tools*, 2007, pp. 1–8.

- [171] “UWSN Group, SENSES lab, SUNSET,” http://reti.dsi.uniroma1.it/UWSN_Group/index.php?page=sunset, accessed: 2014-01-11.
- [172] C. Petrioli, R. Petroccia, and D. Spaccini, “SUNSET Version 2.0: Enhanced Framework for Simulation, Emulation and Real-Life Testing of Underwater Wireless Sensor Networks,” in *Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems*, 2013, pp. 1–8.
- [173] “World Ocean Simulation System (WOSS),” <http://telecom.dei.unipd.it/ns/woss/>, accessed: 2014-04-09.
- [174] F. Guerra, P. Casari, and M. Zorzi, “World Ocean Simulation System (WOSS): a Simulation Tool for Underwater Networks with Realistic Propagation Modeling,” in *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, 2009, pp. 1–8.
- [175] “UNET, The Underwater Networks Project,” <http://www.unetstack.net/doc/html/index.html>, accessed: 2016-02-04.
- [176] S. Shahabudeen, M. Chitre, M. Motani, and A. Siah, “Unified Simulation and Implementation Software Framework for Underwater MAC Protocol Development,” in *IEEE OCEANS*, 2009, pp. 1–9.
- [177] “ARL Laboratory, National University of Singapore,” <http://arl.nus.edu.sg/twiki6/bin/view/ARL>, accessed: 2016-02-04.
- [178] S. S. Viana, L. F. Vieira, M. A. Vieira, J. A. M. Nacif, and A. B. Vieira, “Survey on the Design of Underwater Sensor Nodes,” *Design Automation for Embedded Systems*, vol. 20, no. 3, pp. 171–190, 2015.
- [179] “Aquatec, Aquamodem -1000,” <http://www.aquatecgroup.com/index.php/products-sp-8636/aquamodem/aquamodem-1000>, accessed: 2015-02-10.
- [180] “Desert Star System, SAM-1,” <http://www.desertstar.com/acoustic-modems.html>, accessed: 2014-05-09.
- [181] “EvoLogics, S2CM 48/78,” http://www.evologics.de/en/products/acoustics/s2cm_series.html, accessed: 2014-02-10.

- [182] I. Vasilescu, C. Detweiler, and D. Rus, “AquaNodes: an Underwater Sensor Network,” in *Proceedings of the second workshop on Underwater networks*, 2007, pp. 85–88.
- [183] M. Chitre, I. Topor, and T. Koay, “The UNET-2 Modem—An Extensible Tool for Underwater Networking Research,” in *IEEE OCEANS*, 2012, pp. 1–7.
- [184] “Teledyne Benthos Acoustic modems,” http://teledynebenthos.com/product/acoustic_modems/960-series-atm-965, accessed: 2015-02-09.
- [185] R. Jurdak, C. V. Lopes, and P. Baldi, “Software Acoustic Modems for Short Range Mote-based Underwater Sensor Networks,” in *IEEE OCEANS*, 2007, pp. 1–7.
- [186] D. Torres, J. Friedman, T. Schmid, and M. B. Srivastava, “Software-defined Underwater Acoustic Networking Platform,” in *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, 2009, pp. 1–8.
- [187] H. Yan, S. Zhou, Z. Shi, J.-H. Cui, L. Wan, J. Huang, and H. Zhou, “DSP Implementation of SISO and MIMO OFDM Acoustic Modems,” in *IEEE OCEANS*, 2010, pp. 1–6.
- [188] “UCSB, Aquamodem,” <http://www.ece.ucsb.edu/aquanode/Modem/designspecs.htm>, accessed: 2014-08-20.
- [189] J. Wills, W. Ye, and J. Heidemann, “Low-Power Acoustic Modem for Dense Underwater Sensor Networks,” in *Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 79–85.
- [190] “WHOI,Mico-modem,” <http://acomms.whoi.edu/micro-modem/>, accessed: 2014-03-20.
- [191] “Seaweb, Naval Postgraduate School, US,” http://www.public.navy.mil/subfor/underseawarfaremagazine/issues/archives/issue_45/seaweb.html, accessed: 2016-05-09.
- [192] A. Goodney, Y. H. Cho, J. Heidemann, and J. Wroclawski, “An Underwater Communication and Sensing Testbed in Marina del Rey (poster abstract),” in *Proceedings of the 5th ACM International Workshop on Underwater Networks (WUWNet)*, 2010.
- [193] Z. Peng, J.-H. Cui, B. Wang, K. Ball, and L. Freitag, “An Underwater Network Testbed: Design, Implementation and Measurement,” in *Proceedings of the second workshop on Underwater networks*, 2007, pp. 65–72.

- [194] “Project ORTUN,” <http://www.isi.edu/ilense/ortun/>, accessed: 2016-05-09.
- [195] B. Chen and D. Pompili, “A Testbed for Performance Evaluation of Underwater Vehicle Team Formation and Steering Algorithms.” in *SECON*, 2010, pp. 1–3.
- [196] L. Freitag, K. Ball, J. Partan, E. Gallimore, S. Singh, and P. Koski, “Underwater Acoustic Network Testbed,” in *Proceedings of ACM International Workshop on UnderWater Networks (WUWNet)*, 2011.
- [197] C. Petrioli, R. Petroccia, J. Shusta, and L. Freitag, “From Underwater Simulation to at-sea Testing Using the NS-2 Network Simulator,” in *IEEE OCEANS*, 2011, pp. 1–9.
- [198] “Project DATUNR,” <http://www.isi.edu/ilense/datunr/index.html>, accessed: 2016-05-09.
- [199] “Ocean-TUNE, Community Ocean Testbed,” <http://www.oceantune.org/>, accessed: 2016-05-09.
- [200] B. Wang, “Coverage Problems in Sensor Networks: A Survey,” *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, p. 32, 2011.
- [201] D. Li and H. Liu, “Sensor Coverage in Wireless Sensor Networks,” *Wireless Networks: Research, Technology and Applications*, vol. 2, pp. 3–31, 2009.
- [202] L. Brekhovskikh and I. Lysanov, *Fundamentals of Ocean Acoustics*, ser. Springer series in electrophysics. Springer-Verlag, 1991.
- [203] A. Quazi and W. Konrad, “Underwater Acoustic Communications,” *IEEE Communications Magazine*, vol. 20, no. 2, pp. 24–30, 1982.
- [204] R. Coates, *Underwater Acoustic Systems*, ser. A Halstead Press book. John Wiley & Sons Canada Limited, 1989.
- [205] M. Stojanovic, “Underwater Acoustic communications,” *Encyclopedia of Electrical and Electronics Engineering*, vol. 22, pp. 688–698, 1999.
- [206] I. F. Akyildiz, D. Pompili, and T. Melodia, “Challenges for Efficient Communication in Underwater Acoustic Sensor Networks,” *SIGBED Rev.*, vol. 1, no. 2, pp. 3–8, 2004.

- [207] J. Catipovic, "Performance Limitations in Underwater Acoustic Telemetry," *IEEE journal of Ocean Engineering*, vol. 15, no. 3, pp. 205–216, 1990.
- [208] F. Fisher and V. Simmons, "Sound Absorption in Sea Water," *The Journal of the Acoustical Society of America*, vol. 62, no. 3, pp. 558–564, 1977.
- [209] R. Francois and G. Garrison, "Sound Absorption Based on Ocean Measurements: Part I: Pure Water and Magnesium Sulfate Contributions," *The Journal of the Acoustical Society of America*, vol. 72, no. 3, pp. 896–907, 1982.
- [210] ———, "Sound Absorption Based on Ocean Measurements. Part II: Boric Acid Contribution and Equation for Total Absorption," *The Journal of the Acoustical Society of America*, vol. 72, no. 6, pp. 1879–1890, 1982.
- [211] M. Schulkin and H. Marsh, "Absorption of Sound in Sea-water," *Radio and Electronic Engineer*, vol. 25, no. 6, pp. 493–500, 1963.
- [212] M. A. Ainslie and J. G. McColm, "A Simplified Formula for Viscous and Chemical Absorption in Sea Water," *The Journal of the Acoustical Society of America*, vol. 103, no. 3, pp. 1671–1672, 1998.
- [213] A. B. Coppens, "Simple Equations for the Speed of Sound in Neptunian Waters," *The Journal of the Acoustical Society of America*, vol. 69, no. 3, pp. 862–863, 1981.
- [214] V. A. Del Grosso, "New Equation for the Speed of Sound in Natural Waters (with Comparisons to Other Equations)," *The Journal of the Acoustical Society of America*, vol. 56, no. 4, pp. 1084–1091, 1974.
- [215] C.-T. Chen and F. J. Millero, "Speed of Sound in Seawater at High Pressures," *The Journal of the Acoustical Society of America*, vol. 62, no. 5, pp. 1129–1135, 1977.
- [216] K. V. Mackenzie, "Nine-Term Equation for Sound Speed in the Oceans," *The Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 807–812, 1981.
- [217] S. C. Dhongdi, K. R. Anupama, M. Joneja, and L. Bhatia, "Development of Test-bed Set-up for Underwater Acoustic Communication and Sensor Network," in *National Symposium on Acoustics (NSA)*, 2015.

- [218] “TelosB Mote Platform,” http://www.willow.co.uk/TelosB_Datasheet.pdf, accessed: 2016-05-09.
- [219] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An Operating System for Sensor Networks,” *Ambient intelligence*, vol. 35, pp. 115–148, 2004.
- [220] S. C. Dhongdi, K. R. Anupama, L. J. Gudino, and R. H. Sant, “Deployment Analysis of Underwater Acoustic Sensor Network on Miniature Test-bed,” in *7th International Conference on Advanced Computing and Communication Technology (ICACCT)*, 2013, pp. 18–24.
- [221] S. C. Dhongdi, K. R. Anupama, R. H. Sant, and L. J. Gudino, “Implementation of Multi-hop Time Synchronization on Miniature Test-bed Setup of Underwater Acoustic Sensor Network,” in *Twentieth IEEE National Conference on Communications (NCC)*, 2014, pp. 1–6.
- [222] ———, “Implementation of Multi-hop Bidirectional Communication Link with Time Synchronization on Miniature Test-bed of underwater Acoustic Sensor Network,” *Journal of Engineering Science and Technology Review (JESTR)*, vol. 9, no. 3, pp. 1–8, 2016.
- [223] S. C. Dhongdi, K. R. Anupama, R. Agrawal, and L. J. Gudino, “Simulation and Testbed Implementation of TDMA MAC on Underwater Acoustic Sensor Network,” in *Twenty First IEEE National Conference on Communications (NCC)*, 2015, pp. 1–6.
- [224] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” in *Proceedings of the 33rd IEEE annual Hawaii international conference on System sciences*, 2000, pp. 1–10.
- [225] B. Zeng, J. Wei, and T. Hu, “An Energy-Efficient Data Fusion Protocol for Wireless Sensor Network,” in *10th IEEE International Conference on Information Fusion*, 2007, pp. 1–7.
- [226] S. Singh and C. S. Raghavendra, “PAMAS—Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks,” *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, 1998.

- [227] W. Ye, J. Heidemann, and D. Estrin, “An Energy-Efficient MAC protocol for Wireless Sensor Networks,” in *Twenty-First IEEE Annual Joint Conference of the Computer and Communications Societies, (INFOCOM)*, 2002, pp. 1567–1576.
- [228] Y. Wu, S. Fahmy, and N. B. Shroff, “Energy Efficient Sleep/Wake Scheduling for Multi-Hop Sensor Networks: Non-convexity and Approximation Algorithm,” in *26th IEEE International Conference on Computer Communications (INFOCOM)*, 2007, pp. 1568–1576.
- [229] T. Van Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems*, 2003, pp. 171–180.
- [230] V. Claesson, H. Lönn, and N. Suri, “Efficient TDMA Synchronization for Distributed Embedded Systems,” in *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems*, 2001, pp. 198–201.
- [231] H. Kopetz and W. Ochsenreiter, “Clock Synchronization in Distributed Real-Time Systems,” *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 933–940, 1987.
- [232] J. Elson and D. Estrin, “Time Synchronization for Wireless Sensor Networks,” in *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS)*, 2001, pp. 1965–1970.
- [233] M. Chitre, R. Bhatnagar, and W.-S. Soh, “UnetStack: an Agent-based Software Stack and Simulator for Underwater Networks,” in *IEEE OCEANS*, 2014, pp. 1–10.
- [234] “Subnero,” <http://subnero.com/technology/modem>, accessed: 2016-02-09.
- [235] “Evologics-Wise Modem,” https://www.evologics.de/en/news.html?newsman_news_id=51, accessed: 2014-02-10.
- [236] “Basic Acoustic Model,” <http://www.unetstack.net/doc/html/channels.html>, accessed: 2016-02-04.

Publications Based on Present Work

Journal Publications

- (1) S. C. Dhongdi, K. R. Anupama, R. H. Sant, and L. J. Gudino, "Implementation of Multi-hop Bidirectional Communication Link with Time Synchronization on Miniature Test-bed of Underwater Acoustic Sensor Network," *Journal of Engineering Science and Technology Review (JESTR)*, vol. 9, no. 3, pp. 1-8, 2016.
- (2) S. C. Dhongdi, K. R. Anupama, V. Rege, R. Agrawal, and L. J. Gudino, "Implementation of Dynamic Cluster-Based TDMA MAC Protocol on Three-dimensional Architecture of Underwater Acoustic Sensor Network," *Journal of Networks (JNW)*, - Accepted for publication.
- (3) S. C. Dhongdi, P. Nahar, R. Sethunathan, L. J. Gudino, and K. R. Anupama, "Cross-layer Protocol Stack Development for Three-dimensional Underwater Acoustic Sensor Network." *Journal of Network and Computer Applications (JNCA)*, - Communicated.

Conference Publications

- (1) S. C. Dhongdi, K. R. Anupama, L. J. Gudino, and R. H. Sant, "Deployment Analysis of Underwater Acoustic Sensor Network on Miniature Test-bed," in *7th International Conference on Advanced Computing and Communication Technology (ICACCT)*, 2013, pp. 18– 24.
- (2) S. C. Dhongdi, K. R. Anupama, R. H. Sant, and L. J. Gudino, "Implementation of Multi-hop Time Synchronization on Miniature Test-bed Setup of Underwater Acoustic Sensor Network," in *Twentieth IEEE National Conference on Communications (NCC)*, 2014, pp. 1–6.

- (3) S. C. Dhongdi, K. Anupama, and L. J. Gudino, "Review of Protocol Stack Development of Underwater Acoustic Sensor Network (UASN)," in *IEEE Underwater Technology (UT)*, 2015, pp. 1–17.
- (4) S. C. Dhongdi, K. R. Anupama, R. Agrawal, and L. J. Gudino, "Simulation and Testbed Implementation of TDMA MAC on Underwater Acoustic Sensor Network," in *Twenty First IEEE National Conference on Communications (NCC)*, 2015, pp. 1–6.
- (5) S. C. Dhongdi, K. R. Anupama, M. Joneja, and L. Bhatia, "Development of Test-bed Set-up for Underwater Acoustic Communication and Sensor Network," in *National Symposium on Acoustics (NSA)*, 2015.

**Development of Energy Efficient Network Architecture and
Protocol Stack for Underwater Wireless Sensor Networks
(UWSNs)**

THESIS

Submitted in partial fulfillment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY

by

DHONGDI SARANG CHANDRASHEKHAR

Under the Supervision of

Prof. K. R. Anupama

and

Co-supervision of

Dr. Lucy J. Gudino



BITS Pilani
Pilani|Dubai|Goa|Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

2016

Chapter 8

Conclusions and Future Scope

8.1 Conclusions

In a traditional approach of ocean floor or ocean column monitoring, various sensor nodes are deployed to record the data during the monitoring mission. This data is available to end user only after recovering the sensor nodes from the sensor field. There are several issues with this approach such as 1) real time monitoring is not possible, 2) online system reconfiguration is not available, 3) failure of system can not be identified before recovering the sensor nodes (or instruments), and 4) amount of data is limited by the capacity of on-board memory devices.

These issues are overcome by utilizing an Underwater Acoustic Sensor Network (UASN). UASN is the network of sensor nodes deployed underwater to perform synoptic, co-operative and adaptive sampling of three-dimensional ocean environment. These nodes are networked via wireless links that rely on acoustic communication. A broad range of applications include i) ocean sampling networks, ii) environmental monitoring, iii) undersea explorations, iv) seismic monitoring, v) equipment monitoring, vi) assisted navigation and vii) distributed tactical surveillance operations.

Each of these applications might require completely different approach of design and development of UASN. Most important feature of sensor network development is the provision of flexibility it offers to the developer in the design of network architecture and protocol stack to suit the requirements of application. Correct choice of network architecture and optimal design of protocol

stack has a critical impact on parameters of network performance. Although numerous protocols have been proposed and developed by researchers for the terrestrial wireless sensor networks, these protocols can not be readily applied to UASN because of following peculiar properties of UASN:

- Extremely limited bandwidth of underwater acoustic communication.
- Large and variable propagation delay of acoustic signals in an underwater channel.
- Spatially and temporarily varying channel impulse response.
- Higher probability of bit error and temporary loss of connectivity (shadow zones).
- Harsh and hostile underwater environment damaging nodes due to fouling and corrosion.
- Higher cost of acoustic modems, deployment and recovery operations.
- Severe resource constraints including available energy.

To address these challenges and resource constraints of UASN, it is advocated to use a cross-layer protocol stack implementation. In cross-layer design, functionalities of two or more layers can be combined to form a single coherent framework. Recent studies on WSNs reveal that cross-layer integration and design techniques result in significant improvement in terms of energy conservation. Such an approach is now being tested for the UASN as well.

In this work, a problem of designing three-dimensional UASN has been considered. Such a network can be used for long-term ocean column monitoring application. For this network, a static and structured three-dimensional network architecture has been proposed. To achieve a longer operational life-time of the network, energy efficiency has been the main focus of the design.

The proposed network has a Base-Station node deployed on the sea-surface. BS communicates with the on-shore control station using the RF link. Multiple clusters have been deployed below each other at various depths from the sea-surface. These clusters consist of Cluster-Head node along with certain number of cluster member nodes. The functionalities and addresses of nodes have been configured at the time of deployment of network. The optimum frequency, data rate, modulation technique and power levels have been chosen for nodes based on the characteristics of deep water acoustic channel model.

For the proposed cluster-based three-dimensional network, a cross-layer protocol stack has been developed encompassing important functionalities of physical layer, data link layer, network layer, transport layer, and application layer. Various protocols of network management planes such as time synchronization, clustering, and power level management have also been incorporated in the proposed cross-layer protocol stack. Novel features of the protocol stack can be briefly summarized layer-wise as follows:

1. Physical layer - The design and modeling of physical layer has been done taking into consideration the role of a node in the network as well as the location of deployment. A multi-hop communication link is preferred over the single direct communication link for improving energy as well as bandwidth utilization. Cluster-Head node communicates over long range vertical link as well as small range horizontal link. For these communications, Cluster-Head node adapts an optimum power level for transmission, making effective use of energy. Also, the power levels have been chosen in such a way to allow multiple parallel intra-cluster communication without requirement of multiple frequencies or codes. The provision of multiple parallel communication increases the channel utilization in a severely bandwidth constrained UASN.
2. Data link layer - At the Data Link Layer, a new variant of TDMA MAC has been proposed. TDMA based MAC has been considered, since it offers fairness among network nodes and simplicity of design while avoiding packet collisions, overhearing, and idle listening issues. Proposed variant of TDMA is termed as “Dynamic Cluster-Based TDMA (DCB-TDMA)” MAC protocol, since it utilizes dynamic power level setting of the nodes, and also allows the nodes to be dynamically added or removed from the network. It has been shown that the protocol consists of route formation, Cluster-Head selection and effective power level management schemes. The protocol has been implemented on SUNSET, an open source simulation platform developed by SENSES lab, Sapienza University, Rome. The features of protocol such as node addition, node deletion, Cluster-Head selection, route formation has been showcased using the simulation results. It has been demonstrated that the performance of protocol is better as compared to Basic TDMA based MAC protocol implemented on the same network architecture. Also, the energy awareness feature has been evident from the fact that the protocol has around 50% duty cycle and it balances the energy consumption

of nodes by rotating Cluster-Head nodes based on energy consumption information. Nodes are programmed to sleep for the duration other than the pre-assigned time slots leading to effective energy utilization.

3. Network layer - A multi-hop route has been provided for the forward and reverse link on the network. Route re-establishment in case of Cluster-Head selection has also been showcased using simulation tools.
4. Transport layer - An end-to-end link between BS and every cluster member node has been provided in the network. The reliability and fairness of data transmission on these links has been demonstrated using software tools.
5. Application layer - The application involving collection of data from sensor nodes at regular intervals has been showcased in the work.
6. Time synchronization - In this work, a multi-hop time synchronization protocol has been proposed and developed. This protocol is an extension of Tri-message time synchronization which achieves the time synchronization using only three message exchanges. It has been demonstrated that the protocol provides better performance in terms of accuracy under various situations such as i) delay variations, ii) jitter distribution, iii) time gap from last synchronization. It has also been shown that the protocol consumes very less energy making it energy efficient protocol. Computational cost and time required for achieving synchronization is also lesser as compared to other regression based techniques of time synchronization.
7. Cluster-Head selection - In a proposed protocol stack, energy consumption information of all the network nodes has been regularly collected by the BS. Depending on the energy threshold, a Cluster-Head node has been selected from among the cluster member nodes. This feature has been demonstrated using the simulation tools. The Cluster-Head selection ensures the balancing of energy consumption among all nodes, improving life-span of network. Any failure of network because of energy depleted nodes is also prevented. A chain of Cluster-Head nodes act as a backbone link. In case of a non-responsive Cluster-Head node, a new Cluster-Head node can be selected by the network.

8. Power level management - A Cluster-Head node communicates with other Cluster-Head nodes on vertical link of 500 m using a higher power level of transmission. On the other hand, it communicates with cluster member nodes located in radius of 10 m using a lower power level. This power levels are dynamically adjusted by the Cluster-Head node. Similarly, every node turns off the radio and enters into sleep mode in sleep cycle as well as for duration other than pre-assigned time slots to conserve energy. These features of power level management has been showcased using analytical modeling and simulation tools.

Different software simulation tools have been explored to demonstrate various functionalities of the proposed protocol stack and to validate the results of theoretical modeling. MATLAB simulation has been used to demonstrate time synchronization protocol and to perform comparative analysis with other protocols of time synchronization. SUNSET simulation platform has been used to successfully implement the DCB-TDMA MAC protocol. A three-dimensional cluster-based network has been used for testing the MAC protocol. Around 45 nodes have been included in the network to showcase the scalability. Various features of the protocol such as node addition, deletion, Cluster-Head selection, energy consumption has been successfully demonstrated. The complete protocol stack including multi-hop Tri-message time synchronization has been implemented on a similar three-dimensional network using UnetSim simulator. Analysis of various network parameters has been performed using this simulation tool.

It is a common understanding in the research community that “Simulations are doomed to succeed.” Actual testing of protocol stack on hardware testbed is very important to understand the effects of practical environment. For this purpose, a laboratory based testbed set-up has been built to identify issues related to implementation of the network protocol stack on actual hardware set-up. Proposed cross-layer protocol stack has been deployed on this testbed set-up using various possible topologies. The effect of limitations of hardware on the protocol stack have also been tested. In this way, a testbed set-up has served as a platform for validation of theoretical and simulation results of protocols developed for UASN.

Overall, this work can be taken as a reference guide for design and development of three-dimensional UASN applications. Knowing the requirements of application such as i) types of sensors, ii) time interval of data collection, and iii) dimensions of column can help in deciding the required networking parameters using this work.

8.2 Future scope

Work undertaken in this thesis can be extended in future in following directions :

1. In the proposed network architecture, it has been assumed that the nodes are placed at their respective levels and positions at the time of initial deployment. Hence, the nodes are assumed to be aware of their locations. Such a network architecture is an example of structured deployment. In sensor networks, nodes are usually dropped in a sensing field in a random manner with no fine control over the placements. In such a random deployment scenario, nodes are supposed to have self-organization capability to form an easily manageable network architecture by performing neighbor discovery and positioning algorithms. To obtain position information of the node, a localization algorithm needs to be developed. Localization algorithm provides many features such as correlation of collected data, geographical routing, and node addressing. Also, for the efficient utilization of resources in sensor networks, the sensor nodes are often grouped into disjoint, non-overlapping clusters. For organizing sensor nodes into clusters, a distributed clustering algorithm is necessary. Various such important network services needs to be developed for the self organization of the network, if random deployment is considered.
2. Depending on the interval of sensing and data collection, a huge amount of data transfer takes place in the network. Typical characteristic of the sensor network is the redundant low-rate data and many-to-one flows. An effective data aggregation algorithm can be designed in the sensor network to combine data from several sensors to eliminate redundant transmissions. Such a data-centric approach can minimize the energy consumption prolonging the life-time of the network. In the proposed network architecture, data fusion or aggregation scheme can be applied at Cluster-Head nodes. Such data aggregation scheme may require provision of variable packet size and varying time slot period.