Title
MAC and Routing Protocols for Mobile Underwater Acoustic Sensor Swarms

Author
Noh, Young Tae

Publication Date
2012

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

MAC and Routing Protocols

for Mobile Underwater Acoustic Sensor Swarms

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Young Tae Noh

2012

ABSTRACT OF THE DISSERTATION

MAC and Routing Protocols for Mobile Underwater Acoustic Sensor Swarms

by

Young Tae Noh

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2012

Professor Mario Gerla, Chair

Underwater Acoustic Sensor Networks (UW-ASNs) have recently been proposed as a way to explore and observe the ocean, which covers two-thirds of the Earth's surface. In particular, we consider a SEA Swarm (Sensor Equipped Aquatic Swarm) architecture for short-term *ad hoc* real-time aquatic exploration, such as oil and chemical spill monitoring, submarine detection, and surveillance, by deploying drifting sensor nodes (e.g., UCSD Drogues) to the venue of interest that form a swarm and move as a group with the ocean oceanic current. Each sensor monitors local underwater activities and reports critical data or events in real-time using acoustic multi-hop routing to a distant data collection center, e.g., surface buoys or Autonomous Underwater Vehicles (AUVs).

As SEA Swarm architecture adopts acoustic links as a means of communication, it is accordingly confronted with long propagation delays, low bandwidth, and high transmission power con-

sumption. To put SEA Swarm architecture into practical use and alleviate these limitations, we propose the Delay-aware Opportunistic Transmission Scheduling (DOTS) algorithm to increase channel utilization by harnessing both temporal and spatial reuse. Extensive simulation results show that DOTS outperforms existing solutions, S-FAMA, DACAP, and CS-ALOHA in a line topology, in a highly competitive medium access star topology, and in a random topology with an underwater mobility by harnessing temporal and spatial reuse. Furthermore, in a SEA Swarm architecture, a sensor *cloud* that drifts with water currents and enables 4D (space and time) monitoring of local underwater events such as contaminants, marine life and intruders, is escorted at the surface by drifting sonobuoys that collect the data from underwater sensors via acoustic modems and report it in real-time via radio to a monitoring center. Thus, to realize SEA Swarm architecture, designing an efficient anycast routing algorithm for reliable underwater sensor event reporting to any one of the surface sonobuoys is imperative. Major challenges are the ocean current and the limited resources (bandwidth and energy). We address these challenges and propose two hydraulic pressure based anycast routing protocols, namely HydroCast and VAPR, which exploit the measured pressure levels to route data to surface buoys. The proposed routing protocols are validated via extensive simulations.

The dissertation of Young Tae Noh is approved.

Mani Srivastava

Jack Carlyle

Gregory Pottie

Mario Gerla, Committee Chair

University of California, Los Angeles

2012

To my family...

who have devotionally encouraged and devotedly supported me

to complete the long journey of my PhD study

TABLE OF CONTENTS

1 Introduction **1**

2 Acoustic Channel and its Communication Architectures **5**

2.0.1 Factors Posed by Acoustic Signal 5

2.0.2 Communication Reference Architecture 7

2.0.3 MAC and Routing Protocol Design Challenges in SEA Swarms 10

3 DOTS: A MAC protocol for SEA Swarm Architecture **12**

3.1 Backgrounds 12

3.2 Related Work 14

3.2.1 Temporal Reuse 15

3.2.2 Spatial Reuse 18

3.3 DOTS Prerequisite 19

3.4 DOTS Design 22

3.4.1 Delay Map Management 23

3.4.2 Transmission Scheduling 25

3.4.3 Schedule Recovery 27

3.4.4 Guard Time 28

3.5 Simulation & Evaluation 28

3.5.1 Simulation setup 28

3.5.2 Simulation results 31

4 Hydrocast: A Routing protocol for SEA Swarm Architecture **40**

4.1 Backgrounds . . . 40
4.2 Related Work . . . 42
4.3 Problem Statement . . . 44
4.3.1 Forwarding set selection . . . 45
4.3.2 Geographic routing recovery mode . . . 46
4.4 Forwarding Set Selection . . . 48
4.4.1 Packet delivery probability estimation . . . 48
4.4.2 Packet forwarding prioritization . . . 49
4.4.3 Forwarding set selection methods . . . 50
4.5 Recovery Mode . . . 52
4.5.1 Local lower-depth-first recovery . . . 52
4.5.2 2D void floor surface flooding for recovery path search . . . 54
4.6 Simulations . . . 57
4.6.1 Simulation setup . . . 57
4.6.2 Simulation results . . . 58

5 VAPR: Void Aware Pressure Routing for Underwater Sensor Networks . . . 64

5.1 Backgrounds . . . 64
5.2 Related Work . . . 68
5.3 VAPR Overview . . . 71
5.4 VAPR details . . . 73
5.4.1 Enhanced beaconing . . . 74
5.4.2 Opportunistic directional data forwarding . . . 78
5.4.3 Discussion on the loop free property . . . 83

5.5 Simulations 88
5.5.1 Simulation setup 88
5.5.2 Simulation results 90
5.6 Discussion 96

6 Conclusions **98**

References **99**

LIST OF FIGURES

2.1 Seabed UW-ASNs Architecture 7

2.2 Anchored UW-ASNs Architecture 8

2.3 Anchored UW-ASNs with AUVs Architecture 9

2.4 SEA Swarm Architecture 10

3.1 Temporal Reuse 15

3.2 Spatial Reuse 16

3.3 A deployable UANT system 20

3.4 Number of beacons used in TSHL vs. skew estimate 22

3.5 Example of a transmission decision 25

3.6 Simulation Topologies 31

3.7 Line Topology (Exposed Terminal) 37

3.8 Star Topology 38

3.9 Star and Random Topologies 39

4.1 SEA Swarm architecture 41

4.2 Impact of direction in pressure routing (Dir #2 $>$ Dir #1) 47

4.3 Timer scheduling for prioritization 49

4.4 Recovery mode 52

4.5 Illustration of domination and non-surface node 54

4.6 Minimum alpha value 58

4.7 Probability of a redundant packet transmission 59

4.8 EPAs of different forwarding set selection schemes 62

4.9 Fraction of local maximum nodes over time 62
4.10 Void floor surface detection using a Monte Carlo method 62
4.11 Packet delivery ratio 63
4.12 Number of transmissions for delivery 63
4.13 Average end-to-end delay 63

5.1 Conventional pressure routing in SEA Swarm 65
5.2 Terminology 72
5.3 Enhanced beacon propagation 77
5.4 Beacon receptions in both directions (node i) 79
5.5 Directional data forwarding 80
5.6 Beacon propagation with nodal mobility of $0.3m/s$ and beacon interval of 104s . . 85
5.7 Fraction of nodes reachable to sonobuoys 87
5.8 PDR (*1 sonobuoy scenario*) 87
5.9 Energy consumption per message (*1 sonobuoy scenario*) 90
5.10 Avg. latency (*1 sonobuoy scenario*) 90
5.11 PDR (*64 sonobuoy scenario*) 91
5.12 Energy consumption per message (*64 sonobuoy scenario*) 91
5.13 Fraction of trapped nodes as a function of the number of nodes, under different number of sonobuoys 93
5.14 Effect of different beacon intervals on PDR (under MCM mobility) 94
5.15 Effect of different beacon intervals on energy consumption (under MCM mobility) 94

LIST OF TABLES

2.1 Available bandwidth based on communication ranges 5

5.1 Terminology 73

VITA

Education

2008–2012	Ph.D., in Computer Science, University of California, Los Angeles, USA
2005–2007	M.S., in Information and Communication, Gwangju Institute of Science and Technology, South Korea
1997–2005	B.S., in Computer Science, Chosun University, South Korea

PUBLICATIONS

CONFERENCE

Youngtae Noh, Paul Wang, Uichin Lee, Dustin Torres, Mario Gerla , "DOTS: Delay-aware Opportunistic Transmission Scheduling Algorithm", in IEEE ICNP'10

Uichin Lee, Paul Wang, **Youngtae Noh**, Luiz F. M. Vieira, Mario Gerla, Jun-Hong Cui, "Pressure Routing for Underwater Sensor Networks", in IEEE INFOCOM'10

Youngtae Noh, Saewoom Lee, Kiseon Kim, "Basestation-Aided Coverage-Aware Energy-Efficient Routing Protocol for Wireless Sensor Networks", in WCNC'08

Youngtae Noh, Saewoom Lee, Kiseon Kim, "Central Angle Decision Algorithm in Coverage-Preserving Scheme for Wireless Sensor Networks", in WCNC'08

Changho Yun, Tae-Sik Cho, **Youngtae Noh**, Kiseon Kim, "Generalized Modified Accelerative Pre-allocation WDMA MAC Protocol for WDMA Networks", in APCC 2006

JOURNAL

Youngtae Noh, Uichin Lee, Paul Wang, Brian Sung Chul Choi, Mario Gerla, "VAPR: Void Aware Pressure Routing for Underwater Sensor Networks", IEEE Transactions on Mobile Computing

POSTER

Youngtae Noh, Joshua Joy, Prerna Vij, Dony George, Mario Gerla, "ILS: Instant Localization Scheme for Underwater Mobile Networks", in WuWNet Work-In-Progress Posters, 2011

Seongwon Han, **Youngtae Noh**, Max Ohlendorf and Mario Gerla, "MDOTS: A Multi-session Delay-aware Opportunistic MAC Protocol for Underwater Sensor Networks", WuWNet'11

Joshua Joy, **Youngtae Noh**, Dae-Ki Cho, Uichin Lee, "Towards Personal Content Networking", in MobiCom Work-In-Progress Posters, 2011

Youngtae Noh, Paul Wang, Uichin Lee, Mario Gerla, "Towards Personal Content Networking", in WuWNet Work-In-Progress Posters, 2010

Luiz Vieira, Uichin Lee, **Youngtae Noh**, Mario Gerla, "ILS: Instant Localization Scheme for Underwater Mobile Networks", in WuWNet Work-In-Progress Posters, 2009

CHAPTER 1

Introduction

Underwater Acoustic Sensor Networks (UW-ASNs) have recently been proposed as a way to explore and observe the ocean, which covers two-thirds of the Earth's surface [4, 54, 93]. A UW-ASN typically consists of a large number of underwater mobile/anchored sensor nodes, equipped with a low bandwidth acoustic modem and various sensors, that are deployed to the venue of interest to perform collaborative underwater monitoring and exploring tasks [57, 106]. It may also include Unmanned or Autonomous Underwater Vehicles (UUVs, AUVs), which assist the underwater sensors by performing their own tasks such as oceanography, environmental monitoring, and underwater resource studies without tethers, cables, or remote control. The sensor nodes and AUVs are networked via acoustic links. There are various applications for UW-ASNs. These applications can be classified into two categories. One of the categories is monitoring, such as ocean sampling, environmental monitoring, disaster prevention, and tactical surveillance, and the other is exploration, such as assisted navigation, undersea explorations, and mine reconnaissance [54, 93, 57, 106].

UW-ASNs are similar to Wireless Sensor Networks (WSN), in that deployed nodes in both networks are equipped with sensing, data processing, and communicating components, and that they perform similar tasks, such as monitoring target areas and reporting sensed results [5, 23, 19]. However, there are challenges that are unique to UW-ASNs, mainly produced by different environmental conditions and different communication media (i.e. air vs. water). The goal of this thesis proposal is to identify such challenges and to discuss existing approaches, and to design MAC and Routing protocols overcome them.

The main differences between WSNs and UW-ASNs are listed below:

- *Device prices*: While sensor nodes (motes) in WSNs have become increasingly inexpensive with the advances in technology, acoustic sensor nodes used in UW-ASNs remain expensive. These elevated prices are the result of complex underwater modem and hardware protection needed in the extreme underwater environment. From the economic viewpoint, the small number of device suppliers in UW-ASNs is also responsible for this high price.

- *Power consumption*: The power needed to transmit acoustic waves is greater than that needed for terrestrial radio waves because of the different physical layer technologies. The higher power consumption is required due to the longer distance and the more complex signal processing that compensates for the lossy nature of the underwater channel.

- *Memory capacity*: Terrestrial sensor nodes are equipped with limited storage to maintain the low price of sensor nodes. Acoustic sensor nodes, however, do not need to follow this tendency. Large data storage can be installed in them, which enables additional data caching to better support intermittent communication inherent to UW-ASNs.

- *Level of spatial correlation*: Deployed nodes in UW-ASNs are sparse compared to those of WSNs, due to the prohibitive device prices and the long communication range of acoustic modems. Consequently, the spatial correlation of sensed information is relatively low due to the sparse deployment. This makes it difficult to perform data aggregation, which is typically done in WSNs to reduce the volume of data traffic.

These differences are mostly caused due to the fact that UW-ASNs use acoustic waves, unlike terrestrial WSNs [4, 32, 72, 95]. Is the acoustic wave the only available option for underwater communication?

A large number of studies have been conducted to identify reliable communication methods to provide required throughput and bounded latency to the aforementioned target applications. Radio waves, which are the most popular choice in WSNs, can propagate at long distances through conductive salty water only at extra low frequencies (30 - 300Hz), which require large antennae and high transmission power [87]. According to Zhang *et al.*, the Berkeley MICA2 Motes [1] have

been reported to reach an underwater transmission range of 120 cm at 433MHz in experiments performed at the University of Southern California [102]. The use of radio waves is discouraged for this reason. Optical waves are considered to be another way of underwater communication. These waves do not suffer from high attenuation, but are severely affected by scattering. Furthermore, transmitting optical signals requires high precision in aiming the narrow laser beams at the intended receiver. Electrostatic transconduction is considered as a new underwater communication method, which sets up electric fields that are static for a brief moment (in time) and change the field, through which communication is established [51]. This technology is under development at UCLA's Networked & Embedded Systems Laboratory (NESL) [2]. For the long range underwater communication, acoustic waves are the only practical method. Thus, links in underwater networks have typically been based on acoustic waves.

Despite the technological advances of acoustic communication, we still have limitations that need to be addressed in order for UW-ASNs to be put into practical use. In underwater communication, the available bandwidth is severely limited because acoustic modems use low frequencies. The underwater channel is also severely impaired due to prevalent multipath and fading, and the propagation delay is five orders of magnitude higher and more variable than that of terrestrial channels. As a result, high bit error rates and temporary losses of connectivity can be experienced. Furthermore, because it is difficult to replace and recharge UW-ASN nodes once they are deployed, the nodes must be energy efficient to prolong their lifetime. These limitations must addressed by designing suitable Media Access Control (MAC) and Routing protocols to overcome these limitations.

In Section 2, we discuss the unique challenges that influence underwater MAC and Routing protocol design as well as different underwater communication architectures. In Section 3, we discuss the unique factors that influence underwater MAC protocol design, propose an innovative MAC protocol called DOTS to alleviate the long propagation latency and the severely limited bandwidth of acoustic communication, and provide its extensive simulation results. In Section 4, we discuss challenges of hydraulic pressure based routing, propose a generalized hydraulic

pressure based anycast routing protocol called HydroCast, and provide its extensive simulation results. In Section 5, we discuss Void Aware Pressure Routing (VAPR) protocol that uses surface reachability information to set up each node's next-hop direction toward the surface and provide its extensive simulation results. Finally, in Section 6, we conclude this dissertation.

CHAPTER 2

Acoustic Channel and its Communication Architectures

2.0.1 Factors Posed by Acoustic Signal

Acoustic communications in the underwater environment are made difficult by several factors, including limited bandwidth, lossy channel, and high and variable propagation speed. These factors aggravate the temporal and spatial variability of the acoustic channel.

Table 2.1: Available bandwidth based on communication ranges

Communication range (km)	Bandwidth (kHz)
1000	< 1
10-100	2-5
1-10	10
0.1-1	20-50
< 0.1	> 100

2.0.1.1 Limited Bandwidth

Since the acoustic wave relies on low-frequency, its available bandwidth is limited [91]. The bandwidth of an acoustic communication channel, as shown in Table 2.1, also depends on its communication ranges [87]. As the communication range becomes wider, the available bandwidth becomes narrower. For instance, for the communication range of 1000km, the bandwidth becomes 1kHz.

On the other hand, when the communication range covers several tens of meters, the available bandwidth becomes 100kHz.

2.0.1.2 Lossy Channel

Underwater channels are known to be lossy, because of the inherent properties of acoustic waves and their interaction with the medium (water). The major sources of channel quality degradation are attenuation, fading, and geometric spreading [87]. Attenuation and fading are both due to the absorption of acoustic waves into the medium, and they drastically increase with distance and carrier frequency. Geometric spreading is related to the spreading of sound energy caused by the expansion of the wavefronts, and while it also increases with the propagation distance, it is independent of frequency.

Noise, multipath propagation, and Doppler frequency spread cause additional problems. Noise can be classified into two categories: man-made and ambient. Man-made noise is mainly caused by machinery noise (pumps, reduction gears, and power plants), and shipping activity (hull fouling, animal life on hull, and cavitation), while ambient noise is generated by hydrodynamics (movement of water including tides, current, storms, wind, and rain) and seismic and biological phenomena. Multipath propagation of an acoustic wave may cause severe degradation of the acoustic communication signal quality by generating Inter Symbol Interference (ISI). Doppler frequency spread causes simple frequency translation, which moves signals in a certain frequency band to another band, and continuous spreading of frequencies, where the amount of frequency shift is not constant. Simple frequency translation can easily be compensated at the receiver, while it is difficult to deal with continuous spreading [76]. These variations cause the acoustic channel to be temporally and spatially variable, and cause underwater communication to be less reliable.

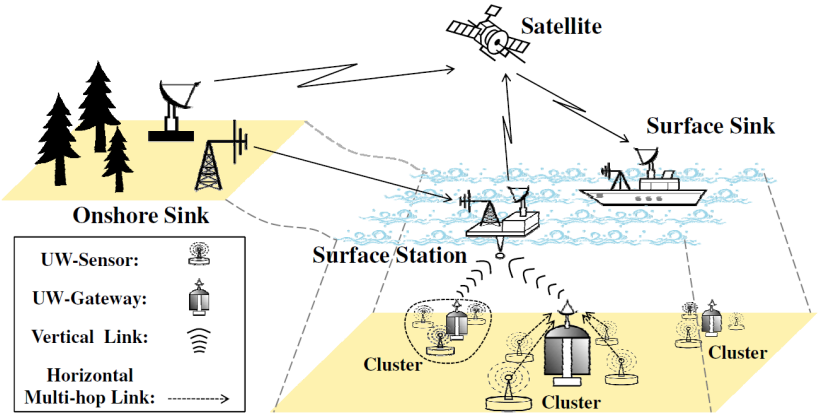


Figure 2.1: Seabed UW-ASNs Architecture

2.0.1.3 Slow and Variable Propagation Speed

Unlike radio waves that travel at the speed of light, the propagation speed of an acoustic wave is 1.5km/s, which is five orders of magnitude slower. Also, the propagation speed is variable, because of shadow zones, surface scattering, bubbles and noise of breaking waves, biological sources, and rain [87, 78]. Slow and variable propagation speed may further reduce the performance of UW-ASNs.

2.0.2 Communication Reference Architecture

In this section, we provide four communication reference architectures, and discuss the method of communications, possible applications, and unique features and challenges of each architecture.

2.0.2.1 Seabed UW-ASNs

In Seabed UW-ASNs, depicted in Fig. 2.1, all underwater sensor nodes are anchored to the bottom of the ocean and compose a two-dimensional communication architecture. The sensor nodes are interconnected to one or more underwater gateways (UW-gateways) via wireless acoustic links, and are grouped into clusters. UW-gateways relay the data from the sensor nodes on the ocean

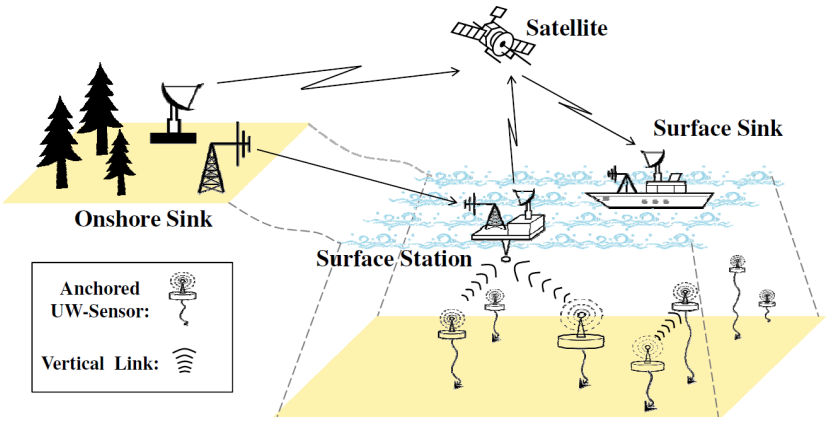


Figure 2.2: Anchored UW-ASNs Architecture

bottom to a surface station. They are equipped with two acoustic modems: vertical and horizontal. The horizontal modem is used by the UW-gateway to send commands to the sensor nodes and to receive sensed data from them. On the other hand, the vertical modem is used only by the UW-gateways to relay gathered data to a surface station. A surface station is equipped with multiple interfaces such as acoustic, RF, and satellite modems. It can also handle multiple parallel communications with the UW-gateways. Communication link between a surface station and a onshore sink or a surface sink is made by long range RF or satellite.

2.0.2.2 Anchored UW-ASNs

In Anchored UW-ASNs, depicted in Fig. 2.2, anchored acoustic sensor nodes compose a three-dimensional communication architecture. In this architecture, each sensor is anchored to the ocean bottom and equipped with a floating buoy that can be inflated by a pump. The buoy pushes the sensor towards the ocean surface. The depth of the sensor can then be adjusted by the wire that connects the sensor to the anchor. The node can observe phenomena and detect events such as disaster symptoms and tactical underwater vehicles between the seabed and the surface [77] [80].

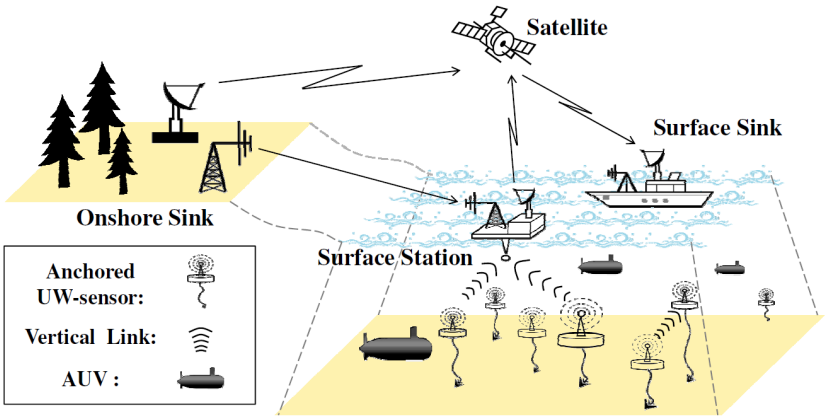


Figure 2.3: Anchored UW-ASNs with AUVs Architecture

2.0.2.3 Anchored UW-ASNs with AUVs

As illustrated in Fig. 2.3, Anchored UW-ASNs can be enhanced with the addition of AUVs. AUVs are underwater vehicles that can perform tasks such as oceanography, environmental monitoring, and underwater resource studies without tethers, cables, or remote control. AUVs help provide enhanced communication reliability and more structural flexibility to the anchored sensor nodes [44]. The reliability is enhanced by making AUVs automatically detect connectivity holes generated by node failures or channel impairment. The flexibility is increased by AUVs that place or replace sensors where their sensed data will be most useful. AUVs can further be used for the installation and maintenance of the sensors.

2.0.2.4 SEA Swarm

In the SEA Swarm (Sensor Equipped Aquatic Swarm) architecture, a large number of acoustic mobile sensor nodes are deployed to the venue of interest to form a SEA Swarm that moves as a group with the water current [57, 106]. Fig. 2.4 illustrates this architecture. The swarm sensor nodes are escorted by surface stations, equipped with multiple interfaces such as acoustic, RF, and satellite modems. There are several advantages of the SEA Swarm communication architecture. First, mobile sensors provide four-dimensional (space and time) monitoring, thus forming dynamic

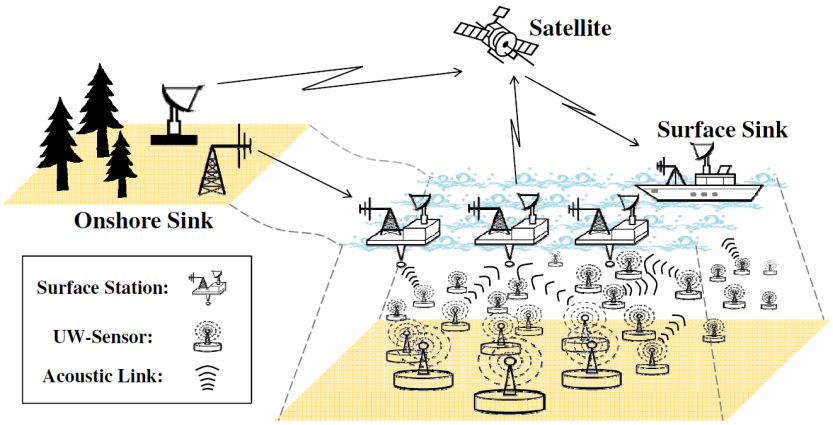


Figure 2.4: SEA Swarm Architecture

monitoring coverage. Second, the multitudes of sensors help provide extra control on redundancy and granularity. Third, floating sensors increase system re-configurability because they can control their depth. Fourth, they resurface once they are depleted of energy, so that they can be recovered and reused. In this architecture, each sensor monitors local underwater activities and reports time-critical data to any one of the surface stations in a multi-hop fashion. The data are then delivered to a monitoring center using radio communications. The mobility of deployed sensor nodes is what makes this architecture unique and poses a new challenge [13].

2.0.3 MAC and Routing Protocol Design Challenges in SEA Swarms

Unlike prevalent underwater communication architectures, a large number of underwater mobile sensor nodes are dropped to the venue of interest to form a SEA Swarm (Sensor Equipped Aquatic Swarm) that moves as a group with water current [57, 106](See Figure 2.4). Since MAC protocols define the ways in which a common medium is accessed by multiple users, the primary goal of MAC protocols is to maximize channel efficiency and to improve the fairness of channel use by multiple users. Although MAC protocols in terrestrial radio packet networks have been intensively studied, they cannot be applied to SEA Swarm because of long propagation delay and lossy nature of the underwater channel. Therefore, it is imperative to design new protocols tailored to the unique

underwater characteristics [70, 86, 78].

As another important issue in the SEA swarm architecture, each sensor monitors local underwater activities and reports time-critical data to any one of the sonobuoys using acoustic multi-hopping; then the data are delivered to a monitoring center using radio communications. Thus, designing an efficient anycast routing protocol is imperative to provide reliable underwater sensor event reporting to any one of the surface sonobuoys. However, this is challenging due to node mobility and limited resources (bandwidth and energy) of mobile sensors. An underwater acoustic channel has low bandwidth and propagation latency five orders of magnitude higher than the radio channel [88]. Acoustic transmissions consume much more energy than terrestrial microwave communications. Such severe limitations in communication bandwidth coupled with high latency and limited energy make the network vulnerable to congestion due to packet collisions. Under these circumstances, minimizing the number of packet transmissions is important for at least two reasons: minimizing congestion and minimizing energy consumption.

In this section, we have analyzed the unique challenges in underwater acoustic communications that affect the design of underwater MAC and routing protocols. We present four reference communication architectures for UW-ASNs and discuss the challenges associated with the underwater environment. In the following section, we will discuss the unique factors that influence underwater MAC protocol design, propose an innovative MAC protocol called DOTS to alleviate the long propagation latency and the severely limited bandwidth of acoustic communication, and provide its extensive simulation results.

CHAPTER 3

DOTS: A MAC protocol for SEA Swarm Architecture

3.1 Backgrounds

In the SEA swarm architecture, medium-access protocol (MAC) protocols designed for terrestrial packet radio networks cannot be directly used because the propagation latency of acoustic signals is much greater than the packet transmission time (e.g., $0.6sec$ vs. $0.04sec$ to transmit a $256KB$ data packet $50kbs$ over a $1km$ range) – carrier sensing in carrier sense multiple access (CSMA) may not prevent concurrent transmissions or packet collisions. This unique situation permits multiple packets to concurrently propagate in an underwater channel, which must be exploited in order to improve the channel throughput. While this phenomenon is also observed in transatlantic wirelines or wireless satellite links, the main departure is that these are point-to-point links without any contention and that the large bandwidth-delay product (BDP) is exploited at a higher layer, namely TCP. In general, long propagation latency in an underwater wireless network creates a unique opportunity for *temporal reuse* that allows for multiple concurrent packets propagating within the same contention domain. Note that temporal reuse is an extra opportunity on top of well-known *spatial reuse* in wireless networks which allows for concurrent, non-colliding transmissions to different destinations if they are sufficiently removed from one another, solving the exposed terminal problem.

Recently a great deal of attention has been focused on exploiting temporal and/or spatial reuse of acoustic channels to improve the throughput. For instance, Slotted FAMA (S-FAMA) uses time slotting in order to lower the probability of collisions by aligning packet transmissions into slots (as in Slotted Aloha) while PCAP allows a node to send multiple reservation requests for trans-

mission time slots (i.e., request to transmit, RTS). In UW-FLASHR, time slots are divided into reservation and data transmission periods to realize efficient channel reservation and to minimize data packet losses caused by control packet exchanges. For better channel utilization, most protocols attempt to build a time domain multiple access (TDMA) schedule using brute-force learning via repeated trial-and-errors [98] or running expensive optimization methods as in ST-MAC [45] and STUMP [55] (an optimal edge coloring problem over a spatial-temporal conflict graph, which is computationally hard). However, discovering a reasonable TDMA schedule requires a network-wide consensus, requiring a rife amount of packet exchanges and taking a considerable amount of time. Such TDMA-based methods are not suitable for mobile sensor networks, because nodes must periodically perform expensive scheduling operations.

However, the key insights from TDMA-based scheduling allows us to enhance conventional CSMA-like random channel access protocols as follows. We need to ensure that transmissions are scheduled carefully such that they do not interfere with the reception of each others' frames by their intended receivers. To satisfy this requirement, each node must evaluate the collision conditions for neighboring frame receptions prior to transmitting a frame. Recall that a *collision* occurs when a receiver tries to decode a frame when more than one frame arrives from different senders simultaneously [8]. Our intuition is that a node can *predict* whether its upcoming frame transmission will collide with another's if it has the neighboring nodes' propagation delay information and their transmission schedules.

In this dissertation, we consider this idea and propose the Delay-aware Opportunistic Transmission Scheduling (DOTS) algorithm designed for underwater mobile sensor networks. The following are the key contribution of DOTS.

- DOTS can effectively exploit temporal and spatial reuse by using only local information by keeping track of neighboring transmissions. DOTS does not require additional packet exchanges because each node can learn a neighboring nodes' propagation delay information and their expected transmission schedules by passively overhearing the transmissions of neighboring nodes. Our extensive simulation results confirm that the use of temporal and

spatial reuse can significantly improve throughput in underwater communications.

- One of the key assumptions of DOTS is clock synchronization because nodes build local delay maps by overhearing packets. Syed *et al.* proposed a protocol called Time Synchronization for High Latency (TSHL) and validated that TSHL can correct clock offset and skew in a reliable and efficient manner using simulations [91]. In this thesis proposal, we implement this protocol on the UANT platform that is composed of a software defined radio and a mix of custom and commercially available hardware for the acoustic transmitter and receiver, and demonstrate that TSHL can effectively synchronize clock offset and skew. To the best of our knowledge, this is the first real implementation of its kind.

- DOTS can effectively handle spatial-unfairness that is similar to channel capture caused by physical location and propagation latency rather than back-off estimates (i.e., the closer the distance between a pair of nodes, the higher the chance of capturing the channel) [92]. We propose a very simple solution based on transmission time stamping in the channel reservation and clearance packets (request/clear-to-send, RTS/CTS). Our simulation results validate that DOTS provides fair channel access.

3.2 Related Work

In multi-hop wireless networks, it is important to efficiently utilize limited network resources and to provide fair access for competing data flows. It has been proven that CSMA provides reasonable performance and fairness [7]. Since CSMA does not require strict scheduling, it can support node mobility, which is also a major challenge in the SEA Swarm architecture (Fig. 2.4). However, the handshaking mechanism of CSMA leads to a severely degraded system throughput due to the presence of long propagation latencies of acoustic waves in UW-ASNs, which is a well recognized problem. Moreover, carrier sensing may fail to detect an ongoing transmission due to the propagation delay, which impairs the performance of CSMA protocols [76].

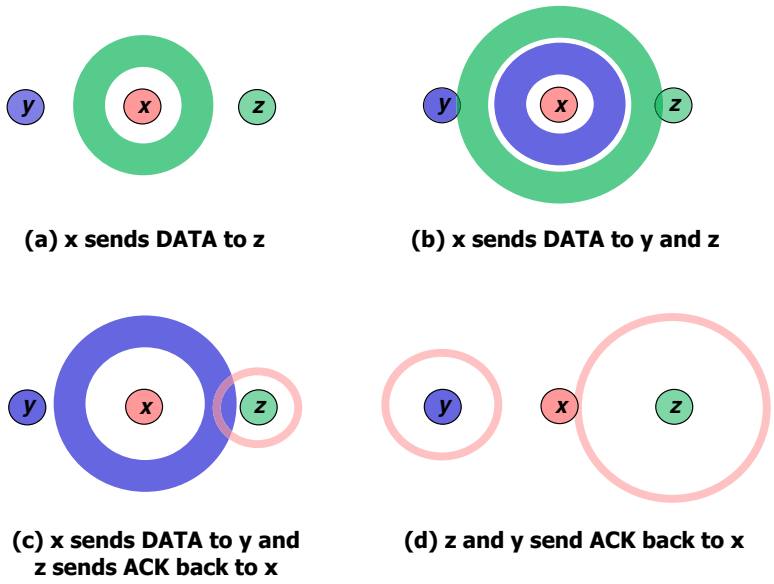
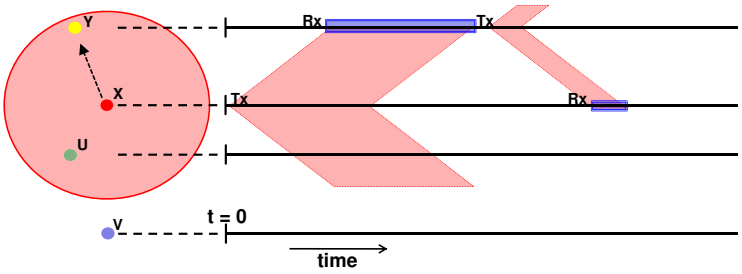


Figure 3.1: Temporal Reuse

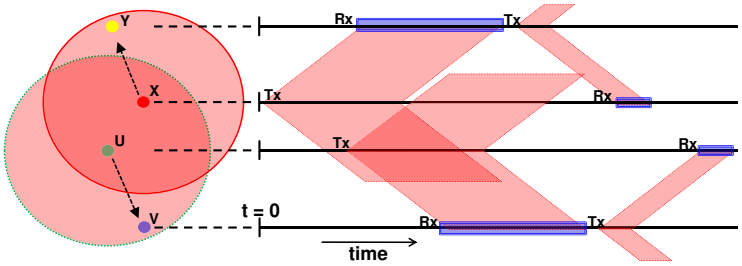
3.2.1 Temporal Reuse

One potential solution for improving CSMA in UW-ASNs is to utilize *temporal reuse* exploiting the long propagation latencies of acoustic waves. Fig. 3.1 demonstrates the notion of temporal reuse. Here, node x sends a DATA frame to node z in 3.1(a) and again at a later time another DATA frame to node y in 3.1(b). Node z sends an acknowledgment (ACK) back to node x as node y is about to receive the transmission from node x in 3.1(c). Finally, node y sends an ACK back to node x in 3.1(d). This case enables the data and ACKs to be transmitted and received without any collision.

To harness this temporal reuse, Yackoski *et al.* proposed UW-FLASHR, a variant of Time Division Multiple Access (TDMA) protocol, which can achieve higher channel utilization than the maximum utilization possible with existing time-based exclusive medium access MAC protocols [98]. Hsu *et al.* proposed ST-MAC, another underwater TDMA protocol, which was designed to overcome spatial-temporal uncertainty based on TDMA-based MAC scheduling for better channel utilization [45]. It operates by constructing Spatial-Temporal Conflict Graph (ST-CG) to describe the conflict delays among transmission links and reduces the ST-CS model to a new vertex



(a) Exclusive medium access: *exposed terminal*



(b) Concurrent transmissions w/o causing collision

Figure 3.2: Spatial Reuse

coloring problem. A heuristic, called the Traffic-based One-step Trial Approach (TOTA), is then proposed to solve the coloring problem. However, TDMA scheduling is performed at a centralized terminal which is a single point of failure. More importantly, in the more realistic SEA Swarm architecture (Fig. 2.4), operation of these protocols is questionable.

CSMA-like scheduling based protocols [96, 20], known as reservation-based protocols, use random access schedules in order to increase channel utilization or to reduce energy consumption through duty cycling. These protocols have explicit reservation periods for complete channel allocation schedules which enable nodes to access the channel based on pre-allocated schedules after reservation periods. While such protocols provide increased channel utilization and collision-free transmission, they do not support network dynamics such as node join, leave, failure, and mobility due to the strict reservation schedules. Furthermore, explicit reservation periods may not only increase throughput but also increase the latency caused by lengthy reservation periods, which will be worsen by long propagation latencies.

Guo *et al.* proposed a propagation-delay-tolerant collision avoidance protocol (PCAP), which fixes the time duration of the handshaking process and then schedules the activities of the deployed sensors to increase throughput and avoid collisions [41]. Due to this difference, the latency of the sender's handshaking procedure becomes predictable, thereby allowing the sender to process more than one simultaneous data frame, providing more scheduling feasibility. However, PCAP gains the ability to schedule transmissions and to increase throughput at the expense of increasing latency in setting up each link.

Chen *et al.* proposed Ordered CSMA, which transmits each data frame in a fixed order [19]. They made an observation that while carriers in opposite directions will collide at a point, two sequential carriers traveling in the same direction will not. By using this phenomenon, each station transmits immediately after receiving a data frame from the previous station sequentially, instead of waiting for a period of maximum propagation delay. However, this protocol is not appropriate for large-scale multi-hop networks because generating collision free transmission order requires relative positions of all nodes in the network. Ordering and maintenance algorithms still need to

be refined in order to adopt this protocol.

Unlike existing underwater CSMA solutions, DOTS does not require an additional phase for reservation scheduling, nor does not restrict transmission schedules to a specific order. Instead DOTS relies solely on passively overhearing neighboring transmissions and estimated delays between transmitting nodes. With this information, DOTS makes intelligent local decisions based upon its own transmission schedule in order not to interfere with a neighboring reception.

3.2.2 Spatial Reuse

Spatial reuse in UW-ASNs also improves the channel utilization by allowing concurrent transmissions. In Fig. 3.2(a), a network topology consisting of 4 nodes is depicted and its corresponding signal propagation in time is drawn on the side. Node x gains the exclusive access of the channel in its collision domain, preventing node u from transmitting to node v, since node u's transmission will interfere with node x's reception of an ACK from node y, known as the *exposed terminal problem*. However, Fig. 3.2(b) shows that it is still possible for node u to transmit concurrently without affecting x's transmission, enabling spatial reuse of the medium.

While such spatial reuse is well-investigated in terrestrial wireless communications [3, 63, 17, 85, 28], to the best of our knowledge, none of the existing work has been done in UW-ASNs. Henceforth, we follow with a short discussion of related works in opportunistic concurrent transmissions for terrestrial networks and its applicability to UW-ASNs.

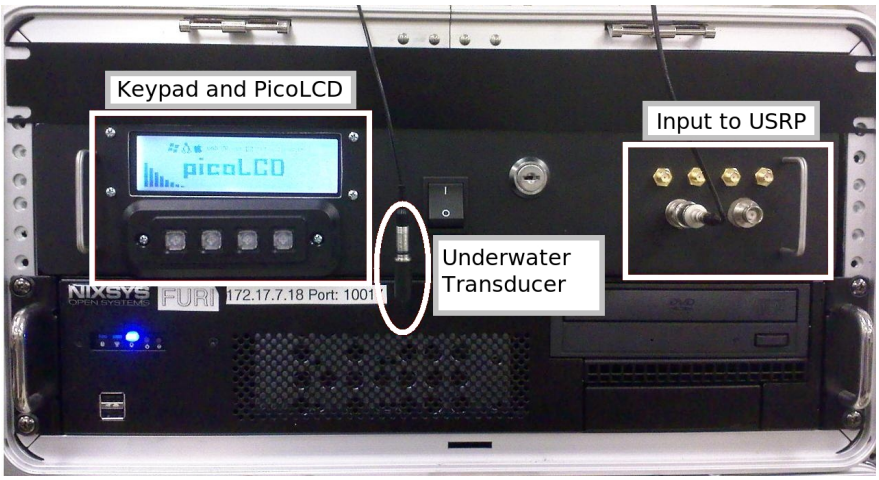
Mittal *et al.* devised an offline training algorithm to determine which nodes may transmit concurrently without collisions [63]. However this approach does not include any mechanisms to deal with exposed terminals. Shukla *et al.* proposed identifying exposed terminals by performing a *Request-To-Send*/*Clear-To-Send* (RTS/CTS) exchange on the basis of overhearing a RTS without overhearing the corresponding CTS [17]. This method, however, does not identify all exposed terminal opportunities, and thus misses exposed terminals where a sender may hear another receiver's CTS. Deepanshu *et al.* proposed the Interference Aware (IA) MAC protocol [85]. Nodes in the IA

MAC make transmission decisions by using the SINR estimates at receivers, which are embedded in CTS frames. However, the IA MAC misses exposed terminals where one of the exposed senders does not hear the CTS from the other receiver. Eisenman *et al.* devised a transmission decision algorithm based on receiver-based feedback to improve the CSMA protocol. The E-CSMA protocol uses observed channel conditions at the transmitter (e.g., RSSI), and builds a per-receiver probability distribution of transmission success on the channel conditions at the sender side [28]. Then a node makes a transmission decision based on channel conditions before attempting to send a frame.

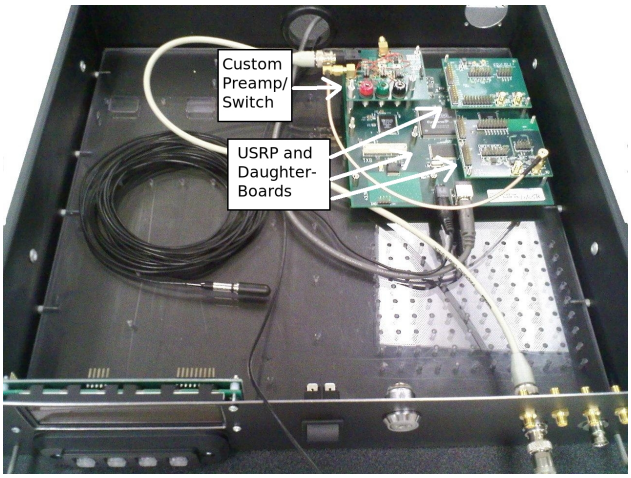
These terrestrial MAC protocols, however, cannot be applied to SEA Swarm architecture due to network dynamics such as node join, leave, and failure and the existence of node mobility caused by the oceanic currents. Unlike existing terrestrial CSMA solutions, DOTS harnesses spatial reuse by passively received information to increase concurrent opportunistic transmissions and uses a guard time to support node mobility.

3.3 DOTS Prerequisite

It has been shown that observed information obtained from passively overhearing neighboring transmissions can be useful in estimating collisions at the intended receivers [81]. DOTS uses the passively obtained information by building a *delay map* to achieve both temporal and spatial reuse by making intelligent transmission scheduling decisions. DOTS therefore is able to compensate for the long propagation latencies and severely limited bandwidth of the acoustic medium by using passively observed information to increase the chances of concurrent transmissions while reducing the likelihood of collisions. However, the lack of clock synchronization could make it difficult for an overhearing node of a transmission to gauge the delay between itself and the transmitting node, and thus, the DOTS protocol makes the assumption of time synchronization amongst all nodes in the network, similar to existing underwater CSMA solutions proposed in [64, 90, 92]. This assumption is necessary in order to accurately enable estimation of the transmission delay between



(a) Front view of UANT system



(b) Internal view of UANT system

Figure 3.3: A deployable UANT system

nodes in a passively promiscuous mechanism.

Syed *et al.* showed that clock offset and skew can be corrected in a reliable and efficient manner to achieve time synchronization for underwater acoustic networks using the Time Synchronization for High Latency (TSHL) protocol [91]. Using this protocol a leading transmitter will send out multiple timestamped beacons. All receiving nodes will calculate the difference between the re-

ceived timestamp and the local time, compute a linear regression over all these values and find the slope of the line. Finally in the second phase offset is found using the skew compensated time. We have implemented this protocol on the UANT platform, seen in Fig. 3.3, which uses a software defined radio and a mix of custom and commercially available hardware for the transmitter and receiver [42]. We show in Fig. 3.4, that after enough beacons are sent the skew between nodes converges and the nodes share the same notion of time. Using a software defined radio introduces non-deterministic latency, so the accuracy of the skew and the offset of two nodes will increase with a dedicated hardware solution even if fewer beacons are used.

Due to clock drift that is apparent in all oscillators, even after nodes have been time synchronized, they will eventually drift apart. This leads to the need for periodic resynchronization. The rate at which a protocol such as TSHL needs to be run depends on the properties of the crystals being used. While very cheap oscillators tend to have a drift of 30-50 parts per million (ppm), many underwater ranging solutions use more precise clocks that are temperature compensated that can achieve accuracies of less than 1 ppm [29]. Two nodes with 50 ppm clocks can accumulate a maximum error of 50ms in approximately 8.3 minutes, while the clock used by Eustice *et al.* in [29] will accumulate 2ms of error in just under 14 hours. Therefore depending on the nodes hardware, resynchronization rates can vary dramatically but are still feasible with limited overhead.

In addition to an accurate clock source being used to reduce overhead of resynchronization, timestamp information of beacons can be piggybacked in the header of a data transfer from the node with the reference clock. In this way when a node is receiving data it can also perform the linear regression and update the values of skew and offset. Since phase two of TSHL requires one packet from the receiving node to be sent back to the transmitter, this information can be appended to the acknowledgement that is sent after the data transfer.

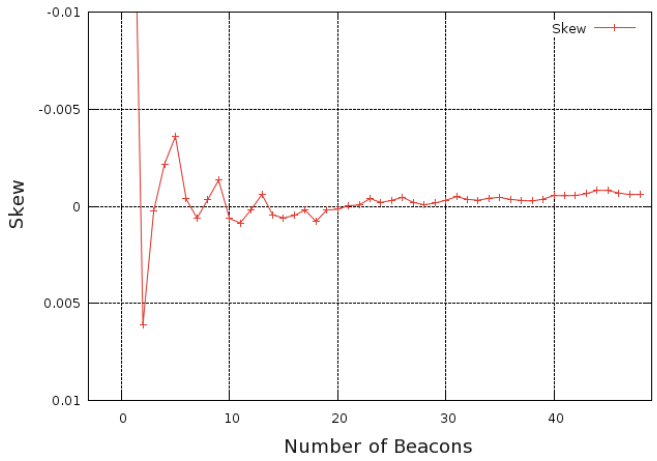


Figure 3.4: Number of beacons used in TSHL vs. skew estimate

3.4 DOTS Design

We now describe our underwater transmission scheduling algorithm, DOTS, which can exploit long propagation delays by using passively observed one-hop neighboring transmissions to improve channel utilization. The design of DOTS is based on CSMA. Because of this design choice, it is confronted with the problem that data transmission between two nearby nodes after RTS/CTS handshaking can be collided with RTS control frames of a distant node due to long propagation delays [33]. Recall that this will happen more frequently and be more expensive in underwater acoustic networks than in terrestrial radio networks due to the high latency and transmission costs. Fullmer *et al.* identified the problem in [34] and provided the following two conditions for collision free transmission:

- *RTS wait time* should be greater than the maximum propagation delay, which is the propagation delay for a transmitted frame to reach its maximum transmission range.
- *CTS wait time* should be greater than the RTS transmission time plus twice the maximum propagation delay plus the hardware transmit-to-receive transition time.

Thus, these two conditions are the basis of DOTS protocol in order to avoid frame collisions. With the assumption of synchronization, DOTS can locally calculate the distributed transmission

and reception schedules to perform concurrent transmission when viable by promiscuously overhearing neighboring transmissions. DOTS maintains minimal internal state in a delay map database to keep track of observed neighboring transmission and reception schedules. This database is updated based on each observed frame's MAC header. In addition to standard source, destination, sequence number, frame size and Cyclic Redundancy Check (CRC) checksums in the MAC header, DOTS necessitates two additional information to be placed in the MAC header, namely an accurate clock synchronized timestamp of when the frame was sent and an estimate of the propagation delay between the source and destination. This estimate of the propagation delay between the source and the destination of the overheard frame can be performed during the clock synchronization process by examining the time of flight information during the frame exchanges and later updated through further communication between the nodes. Moreover, the delay map database entries can expire and be removed over time with the knowledge of data size of each entry and the maximum propagation delay time for each overheard frame in order to keep the number of database entries small.

Whenever a node has a frame to send, it performs a transmission scheduling decision algorithm based on its delay map database to make a decision as to whether or not to begin its transmission, which will be further discussed in Section 3.4.2. If no conflicts are detected, it begins its transmission. Otherwise, it backs off for a random amount of time. Since each node may miss a neighbor's RTS or CTS transmission due to channel fading in underwater, conflict detection schedules may still cause collisions. Thus, to reduce the damage and to avoid deadlock, DOTS provides for a recovery scheme, the details of which will be discussed in Section 4.5. Finally, since deployed nodes are moving along with the ocean current, it requires a guard time to avoid invalid transmission scheduling caused by the node mobility, which will be further discussed in Section 3.4.4.

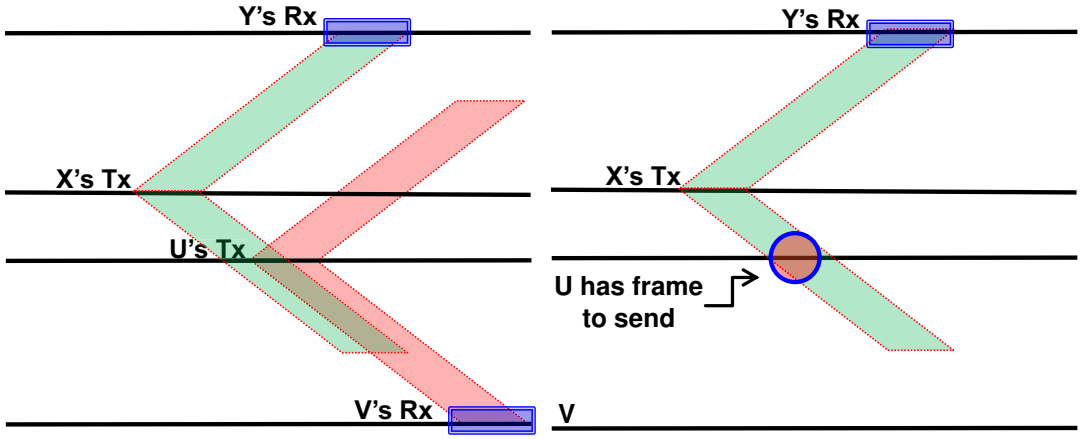
3.4.1 Delay Map Management

By passively observing neighboring transmissions, each node can maintain a *delay map*, which must contain the following information:

- *source*: the sender of the observed MAC frame
- *destination*: the intended destination of the observed MAC frame
- *timestamp*: the time at which the observed MAC frame was sent
- *delay*: the estimated propagation delay between the source and the destination for the MAC frame

With clock synchronization, the value of the timestamp can not only provide time information for each frame but also be an accurate indicator of the distance between the sender and the overhearing node itself. Each node can calculate a neighbor's propagation delay to itself by subtracting the timestamp of the MAC frame from the reception time of the MAC frame. Thus, the timestamp and delay fields provide additional distance information between the sender and overhearing node and between the sender and intended frame receiver. Given this additional information, each node can build a delay map of its one-hop neighbors and calculate the expected time a response back to the sender of the observed MAC frame will occur.

Due to network dynamics, neighboring nodes' transmissions can be backed-off or canceled. Furthermore, information of delays between each node and its one-hop neighbors can become stale. To adapt to these dynamics, an update process of the delay map is required. Whenever a new transmission is overheard, each node searches the delay map to check for the existence of existing entries based on source and destination fields. When a duplicate entry is detected, the node checks the freshness of the existing item. If the entry is staler than the latter, then the latter replaces the former. As time passes, the delay map may become unnecessarily larger. To keep the size of the delay map manageable, outdated entries are removed. Whenever an entry is added to the delay map, a timer for each entry is set. Once the timeout is triggered for an item, the item is removed from the delay map.



(a) Node U receives node X's frame and runs the tx decision algorithm

(b) Node U sends a frame to node V.

Figure 3.5: Example of a transmission decision

3.4.2 Transmission Scheduling

Based on the delay map, a node decides whether or not it can transmit without interfering with a neighbor's reception. Fig. 3.5 provides an example of the transmission scheduling decision process. Node x sends a RTS to node y. When node u receives this RTS and has data to send, it can begin its own transmission to node v concurrently if the following two conditions hold:

- *Neighboring non-interference*: Its current transmission (RTS) and future transmission (DATA) must not interfere with neighbors' ongoing and prospective receptions.
- *Prospective non-interference*: Its future receptions (CTS and ACK) must not be interfered with by neighbors' prospective transmissions.

As for the *neighboring non-interference* condition, node u needs to check whether its RTS will interfere with the reception duration of y's CTS at node x or not. The arrival time of y's CTS at node x can be calculated as follows:

$$r_{CTS(y)} = t_{RTS(x)} + \Delta_{MaxProp} + \tau_{CTS(y)} + \Delta_{y \to x} \tag{3.1}$$

where $t_{RTS(x)}$ denotes the timestamp of node x's RTS transmission, $\Delta_{MaxProp}$ denotes the sum of the maximum propagation delay between any two nodes, the control packet processing time, and the packet reception duration, $\tau_{CTS(y)}$ denotes CTS frame processing time, and $\Delta_{y->x}$ denotes the delay between node y and node x. Reception duration of node y's CTS at node x can be calculated as below:

$$\Delta_{CTS(y)} = [r_{CTS(y)},\ r_{CTS(y)} + \frac{\ell_{CTS}}{\lambda_{DATA}}] \tag{3.2}$$

where ℓ_{CTS} denotes data length of CTS and λ_{DATA} denotes data rate. Similarly, the arrival time of y's ACK at node x can be calculated as follows:

$$r_{ACK(y)} = r_{CTS(y)} + \Delta_{DataProp} + \tau_{ACK(y)} + \Delta_{y \to x} \tag{3.3}$$

where $\tau_{ACK(y)}$ denotes ACK frame processing time and $\Delta_{DataProp}$ denotes the sum of the processing time for a DATA frame, the propagation delay between a sender and its intended receiver, and the transmission duration for the DATA frame. Reception duration of y's ACK at node x can be calculated as below:

$$\Delta_{ACK(y)} = [r_{ACK(y)},\ r_{ACK(y)} + \frac{\ell_{ACK}}{\lambda_{DATA}}] \tag{3.4}$$

Finally, node u makes a decision to launch its RTS transmission when its *current time* + *delay from node u to x* is not in the time ranges of (3.2) and (3.4).

As for the *prospective non-interference* condition, node u needs to check whether its CTS and ACK reception duration (received from node v) will be interfered with by the reception duration of x's DATA whose intended receiver is y or not. Expected arrival time of v's CTS at node u can be calculated as follows:

$$r_{CTS(v)} = t_{RTS(u)} + \Delta_{MaxProp} + \tau_{CTS(v)} + \Delta_{v \to u} \tag{3.5}$$

where $t_{RTS(u)}$ denotes the time-stamp of node u's planned RTS transmission. Reception duration of v's CTS at node u can be calculated as below:

$$\Delta_{CTS(v)} = [r_{CTS(v)},\ r_{CTS(v)} + \frac{\ell_{CTS}}{\lambda_{DATA}}] \tag{3.6}$$

We can similarly calculate the expected arrival time of v's ACK at node u as below:

$$r_{ACK(v)} = r_{CTS(v)} + \Delta_{DataProp} + \tau_{ACK(v)} + \Delta_{v \to u} \tag{3.7}$$

Reception duration of v's ACK at node u can be calculated as follows:

$$\Delta_{ACK(u)} = [r_{ACK(v)},\ r_{ACK(v)} + \frac{\ell_{ACK}}{\lambda_{DATA}}] \tag{3.8}$$

The expected arrival time of x's DATA at node u can then be calculated as below:

$$r_{DATA(x)} = t_{RTS(x)} + 2{\times}\Delta_{MaxProp} + \tau_{DATA(x)} + \Delta_{x \to u} \tag{3.9}$$

Then, the reception duration of x's DATA at node u can be calculated as follows:

$$\Delta_{DATA(x)} = [r_{DATA(x)},\ r_{DATA(x)} + \frac{\ell_{DATA}}{\lambda_{DATA}}] \tag{3.10}$$

Now, node u makes a decision to launch its RTS transmission when the time ranges of (3.6) and (3.8) is not in (3.10).

3.4.3 Schedule Recovery

Collisions may occur during successive transmissions. A node may miss its neighbors' RTS/CTS due to the half-duplex nature of the acoustic modem or the lossy nature of the acoustic channel, and begin its transmission sequence causing a frame collision. Since each transmission decision is made locally, there is no way to provide collision-free scheduling. DOTS provides for a schedule recovery scheme to minimize the damage caused by a collision or a lost frame and avoid deadlocks.

Transmission scheduling recovery occurs in both sender and receiver sides. At the sender side, when sending a RTS or a DATA frame, a timer is set to when the corresponding CTS and ACK frames should arrive by. Once the timer is triggered, the sender knows that its transmission has been lost or a collision has occurred. The sender backs off and will retry to send the frame later. At the receiver side, a collision can be detected in a similar fashion when the DATA frame does not arrive before a timer is triggered. Once the timer is triggered, then the receiver can reset its state to either send frames if it has any or to receive future frames.

When two or more transmission schedules conflict at a node by network dynamics, this algorithm can use the timestamp knowledge in its delay map database to give preference to one of the transmission schedules. The other schedules can be allowed to have their timers expire, effectively canceling the schedule. When the timers expire, yielded nodes fall into random backoff and then run transmission decision algorithm again to reschedule their transmissions.

3.4.4 Guard Time

DOTS uses a guard time to support node mobility caused by the oceanic currents. Each node calculates this guard time as $2 * (average\ movement\ distance/speed\ of\ sound)$ when it checks the transmission scheduling algorithm. The multiplier, 2, is used since both the sender and the receiver may move in opposite directions from each other. This guard time is then added to the guard time in the frame reception duration, which leads to a smaller range of allowable concurrent transmissions.

3.5 Simulation & Evaluation

3.5.1 Simulation setup

3.5.1.1 Simulation parameters

For acoustic communications, the channel model, described in [88] and [62], is implemented in the physical layer of QualNet to estimate delivery probability. The path loss over a distance d for a signal of frequency f due to large scale fading is given as below:

$$A(d, f) = d^k a(f)^d \tag{3.11}$$

where k is the spreading factor and $a(f)$ is the absorption coefficient. The geometry of propagation is described using the spreading factor $(1 \leq k \leq 2)$; for a practical scenario, k is given as 1.5. The absorption coefficient $a(f)$ is described by the Thorp's formula [62]. The average Signal-to-Noise

Ratio (SNR) over distance d is thus given as follows:

$$\Gamma(d) = \frac{E_b/A(d,f)}{N_0} = \frac{E_b}{N_0 d^k a(f)^d} \tag{3.12}$$

where E_b and N_0 are constants that represent the average transmission energy per bit and noise power density in a non-fading additive white Gaussian noise (AWGN) channel. As in [88, 12], we use Rayleigh fading to model small scale fading where SNR has the following probability distribution:

$$p_d(X) = \frac{1}{\Gamma(d)} e^{-\frac{X}{\Gamma(d)}} \tag{3.13}$$

The probability of error can be evaluated as follows:

$$p_e(d) = \int_0^\infty p_e(X) p_d(X) dX \tag{3.14}$$

where $p_e(X)$ is the probability of error for an arbitrary modulation at a specific value of SNR X. In this dissertation, we use BPSK (Binary Phase Shift Keying) modulation because BPSK is widely used in the state-of-the-art acoustic modems [32]. In BPSK, each symbol carries a bit. In [79], the probability of bit error over distance d is given as below:

$$p_e(d) = \frac{1}{2}\left(1 - \sqrt{\frac{\Gamma(d)}{1+\Gamma(d)}}\right) \tag{3.15}$$

Thus, for any pair of nodes with distance d, the delivery probability of a frame with size m bits is simply given as follows:

$$p(d,m) = (1 - p_e(d))^m \tag{3.16}$$

We generate different channel fading conditions in the simulations by adjusting the transmission power in dB re μ Pa and SNR threshold [88].[1]

Unless otherwise mentioned, the data rate is set to 50*kbps* as in [105]. We vary transmission ranges from $750m$ to $1.5km$ and data size from 512*bytes* to 1*kbyte* to observe behavior of each protocol in terms of varying distance (longer propagation delays) and varying data size. Note again

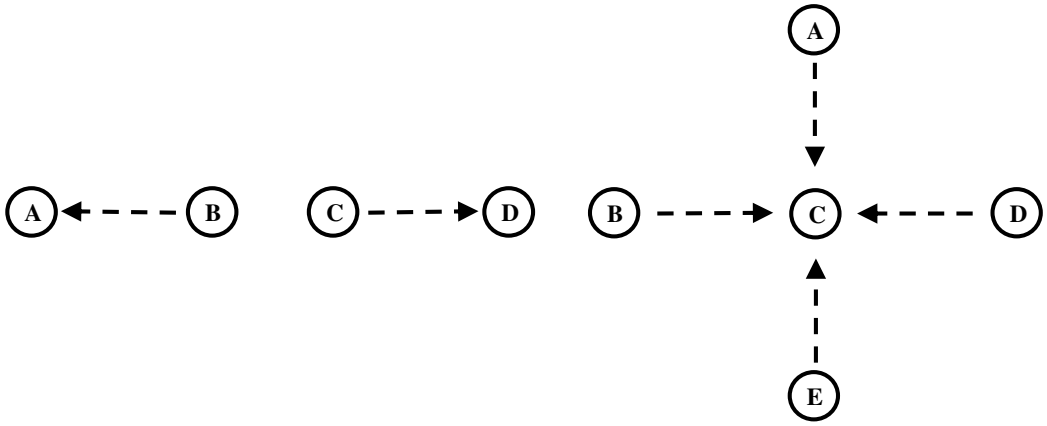
[1]The signal intensity is measured in *dB* re μPa of the power flux $[Wm^{-2}]$ delivered into the water by a source.

that at a data rate of 50*kbps* a 1*kbyte* frame requires 0.16384*sec* to transmit and the one-way trip delay on a $1.5km$ link is approximately 1*sec* ($>>$ tx duration = 0.16384*sec*) considering acoustic propagation delay and transmission duration. We measure throughput and energy consumption per node as the functions of the offered load on the sensor network. To further observe behaviors of each protocol, we also vary data size and distance between a pair of nodes. The load is varied between generating a single frame every 30*sec* down to a single frame every 0.25*sec*. In our simulation, each run lasts 1*hour*. Unless otherwise specified, we report the average value of 50 runs with the 95% confidence interval.

3.5.1.2 Topology

As shown in Fig 3.6, we deployed the nodes in a line and a star topologies in a 3D region of $5km \times 5km \times 5km$. In the line topology depicted in Fig. 3.6(a), four nodes are deployed in a line and with a fixed distance between one-hop neighbors. The distance between the nodes are varied from $750m$ to $1.5km$ for the experiments, and thus the two nodes, B and C, are exposed to each other. We adopt this line topology to show how spatial reuse affects system throughput. As the distance between each pair increases, simulation results will also indicate how temporal reuse can affect system throughput. The star topology, depicted in Fig. 3.6(b), shows a more aggressive traffic toward the center node (c) since the four surrounding nodes attempt to simultaneously send their data to the center node. In this scenario, we create a high contention situation between the four outer nodes for the center node. The distance between the center node and the four surrounding nodes is varied over our experiments ranging from $750m$ to $1.5km$. Here, increasing the distance between the nodes will attest to the benefits of temporal reuse in the presence of high contention.

In addition to these static topologies, we randomly deployed 10 nodes in a 3D region of $430m \times 430m \times 430m$ with a transmission range of $750m$ to test node mobility to support the SEA Swarm architecture. This region enables all deployed nodes to be fully connected and exposed to high levels of channel contention as in [92, 90]. We adopt an extended 3D version of the Meandering Current Mobility (MCM) Model [13] to pattern the motility of each sensor node. Unlike



(a) Line Topology (b) Star Topology

Figure 3.6: Simulation Topologies

most existing sensor node mobility patterns from literature which assumes that each node moves independently of all others, wherein its path vector is determined from an independent realization of a stochastic process, the MCM model considers fluid dynamics whereby the same velocity field advects all nodes. Here, the MCM model considers the effect of meandering sub-surface currents (or jet streams) and vortices on the deployed nodes to pattern its path vector. In our simulations, we restrict the nodes move with a maximum speed of $0.3m/s$ with the MCM model to test the resiliency of the guard time in DOTS.

3.5.2 Simulation results

3.5.2.1 Throughput

To evaluate the protocol performance, we measure the throughput as a function of the offered load, defined as follows:

$$\text{Throughput} = \frac{\text{\# of } data \text{ rx} \times \Delta_{data}}{\text{Simulation Duration}} \tag{3.17}$$

where Δ_{data} denotes the duration of transmitting a data frame.

$$\text{Offered Load} = \frac{\text{\# of Generated } data \times \Delta_{data}}{\text{Simulation Duration}} \tag{3.18}$$

The performance of DOTS was compared to that of three CSMA protocols, namely Slotted FAMA (S-FAMA) [33], DACAP [71], and CS-ALOHA with ACK [35]. S-FAMA is a synchronized underwater MAC protocol based on RTS/CTS handshaking. The main idea of S-FAMA is to time slot exclusive access to the channel medium so that the time duration of each slot is long enough to ensure that any frame transmitted at the start of the slot will reach the destination before the slot duration ends. DACAP is a non-synchronized protocol that allows each node to use different handshaking lengths for different distances between the sender and the receiver. To reduce collision, DACAP follows these two collision avoidance conditions: 1) when a receiver overhears a RTS threatening its pending data reception, the receiver sends a very short warning frame to its intended sender to defer its data transmission until the predefined waiting period 2) after sending a RTS, if a sender overhears a CTS threatening the neighbor's pending data reception, then it defers its data transmission. CS-ALOHA with ACK is ALOHA adapted for the underwater environment, where each node transmits whenever the channel is idle without performing the RTS/CTS handshaking process.

Fig. 3.7(a), 3.7(b), and 3.7(c) show the throughput of the four protocols with different data sizes and transmission ranges in the line topology (exposed terminal). As shown in Fig. 3.7(a) and 3.7(b), DOTS outperforms S-FAMA by a factor of two and DACAP and CS-ALOHA by around 15% for a $750m$ transmission range with both $512bytes$ and $1024bytes$ data frame sizes. It is noteworthy that DACAP outperforms S-FAMA by two times because DACAP allows for concurrent transmissions of the two sender-receiver pairs in Fig. 3.6(a); when a sender-receiver pair (A-B) is undergoing data transmission in the line topology, the other pair (C-D) can also perform parallel data transmission because the two collision avoidance conditions of DACAP cannot suppress the transmissions of the two sender nodes (B and C). Consequently, this allows DACAP to perform concurrent transmission possibly with collisions; however, it is the result of these minor collisions which explains the utilization gain of DOTS over that of DACAP. By varying the data size, Fig. 3.7(a) and 3.7(b) show that data size is proportional to the increase in throughput of all handshaking based protocols.

Fig. 3.7(b) and 3.7(c) show the throughput behaviors of the four protocols with different transmission ranges for $1024byte$ frames. Fig. 3.7(c) particularly shows that DOTS substantially outperforms S-FAMA by 10 times and DACAP by 4 times. This can be attributed to DOTS's ability to exploit temporal reuse. As the transmission range doubles, performance of S-FAMA and DACAP degrades by half while that of DOTS increases. DOTS factors in the increased propagation delays to exploit temporal reuse. With the increase in transmission range, one way propagation latency increases from $0.5sec$ to $1sec$, creating more opportunities for DOTS to exploit temporal reuse. Given that a $1kbyte$ data transmission duration takes approximately $0.16384sec$ while that of a $40bytes$ control frame takes about $0.0008sec$, this implies that data frames gain about three more reusable data transmission slots whereas control frames gain about 625 more transmission slots by this temporal reuse. The throughput of CS-ALOHA remains the same regardless of the transmission range increase because although it takes advantage of temporal reuse, it lacks the capability to avoid collisions, thereby offsetting the gains from temporal reuse. It is also interesting to note that the all four protocols show a saturation point. The throughput increases as the offered load increases until a threshold limit. After reaching the threshold point, the all four protocols suppress their transmissions and thus their performance becomes saturated.

In the star topology, the four outer nodes compete to send their frames to the one center node. Fig. 3.8(a) and 3.8(b) show that DOTS outperforms S-FAMA and CS-ALOHA by two times and DACAP by 70% for a $750m$ transmission range with both $512bytes$ and $1024bytes$ data frame sizes. By varying the data size, these two figures show that the three handshaking based protocols exhibit the behavior that throughput is proportional to data frame size. On the other hand, CS-ALOHA shows unstable throughput performance; when the data size exceeds a threshold, CS-ALOHA significantly increases its collision rate and reduces its overall throughput.

By increasing the transmission range from $750m$ to $1500m$, Fig. 3.8(b) and 3.8(c) show that DOTS outperforms S-FAMA by 8 times and DACAP by 4 times. This is because S-FAMA's performance degrades as the transmission range increases, as does DACAP's; however DACAP is more aggressive in the channel medium access and therefore compensates for some of the perfor-

mance degradation. In contrast, DOTS shows a vastly superior behavior. As the communication range increases, it allows for more temporal reuse. In this star topology, DOTS outperforms S-FAMA by 8 times and DACAP by 4 times. Inversely, CS-ALOHA provides the worst throughput due to absence of collision avoidance.

3.5.2.2 Energy consumption

Fig. 3.9(a), which represents the four throughput lines of the protocols in Fig. 3.8(a), shows the average power consumption of the four protocols in the star topology with a $750m$ transmission range and $1024bytes$ data frame size. It shows the average energy consumption of each protocol per node during entire simulation. When it is compared with the throughput lines of the four protocols in Fig. 3.8(a), it implicitly indicates that the number of collisions which occur in each protocol. DOTS consumes more energy than S-FAMA and DACAP because it delivers, by far, more frames than these two protocols. Inversely, throughput for CS-ALOHA about 20% lower than that of DOTS, yet the energy consumption of CS-ALOHA is several times higher illustrating that CS-ALOHA consumes significantly more energy due to collisions.

3.5.2.3 Mobility

The effect of random topologies and node mobility are examined in Fig. 3.9(b). Ten nodes are randomly deployed to a region which enables full connectivity between all nodes, whereby each node follows a jet stream path vector based on the MCM model. The main jet stream speed of each node is capped at $0.3m/s$ with each node having a $750m$ transmission range. Five pairs of sender-receiver nodes are actively engaged in data communication, transmitting 512*byte* data packets. Note that with a $0.3m/s$ jet stream, nodes can move approximately $20m$ in 60 $seconds$, henceforth a $20ms$ guard time is amply chosen for use in DOTS to allow for approximately up to a $30m$ variation of node locality.

Fig. 3.9(b) shows that DOTS outperforms DACAP by 30% and S-FAMA by 3 times. With a

random topology and node mobility, DOTS clearly provides reliable throughput and performance gains over DACAP and S-FAMA by utilizing smart and adaptive scheduling techniques to harness temporal and spatial reuse. On the other hand, CS-ALOHA shows the best performance in the random topologies with node mobility for our test parameters, however, this comes at a steep price in terms of energy efficiency and fairness, which will be addressed in Secton 3.5.2.4.

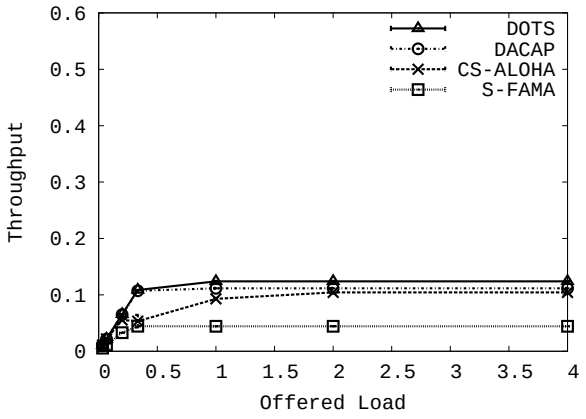
3.5.2.4 Fairness

MAC protocols with backoff schemes (i.e., binary exponential) based on insufficient information about the network congestion may cause *spatial unfairness*, a form of channel capture, as described in [92]. Since a frame's propagation latency is proportional to the distance from a sender, the channel clears earlier for nodes closer to the sender. Closer nodes consequently have more opportunities to recapture the channel, resulting in unfairness amongst the nodes. To characterize the fairness, we use the Jain Fairness Index [48], defined as below:

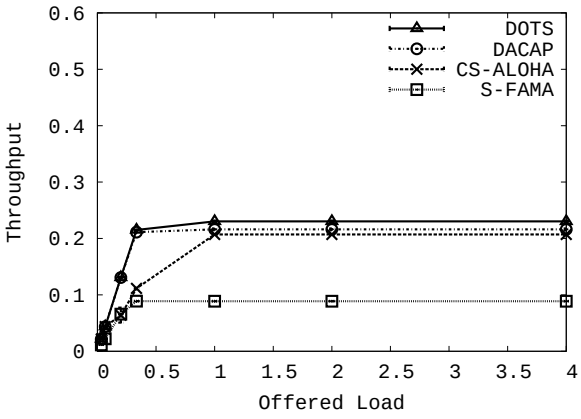
$$\text{Fairness Index} = \frac{(\sum x_i)^2}{(n \cdot \sum x_i^2)} \tag{3.19}$$

where x_i denotes the throughput of node i and n denotes the number of nodes in the network. Fig. 3.9(c), which is the corresponding fairness plot to Fig. 3.9(b), shows that S-FAMA and DOTS exhibit a high fairness index (0.9 and above) and also remain stable and constant with increased offered load. As described in 4.5, when more than one transmission schedule contends in a node, DOTS uses the timestamp knowledge in its delay map database to give preference to one of the transmission schedules. DOT with random backoff exhibits high fairness for this reason. The reason for the slightly lower fairness of DOTS compared to S-FAMA is due to the use of temporal and spatial reuse. In DOTS, every sender-receiver pair has a fair chance of accessing the medium as in S-FAMA, yet some pairs are given the chance of concurrently accessing the medium, thus slightly affecting the fairness index. DACAP provides a lower fairness index than both S-FAMA and DOTS. This is because DACAP gives priority to the nodes already accessing the channel and consequently causes this bias. CS-ALOHA shows the lowest fairness index and the

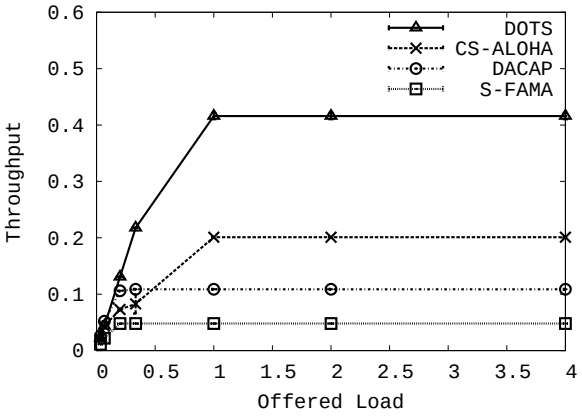
largest variation. Due to CS-ALOHA's binary exponential backoff, it allows close sender-receiver pairs to potentially capture the channel, thereby severely degrading the fairness but providing best throughput performance as indicated in Fig. 3.9(b). This channel capturing also leads to severe data collisions at other nodes which have not captured the channel, inducing poor energy utilization. Furthermore, as Fig. 3.9(b) indicates, CS-ALOHA is subject to far greater amounts of instability and throughput variation as a result of this capture effect.



(a) Throughput as a function of offered load with fixed data size (512*bytes*) and transmission range ($750m$)

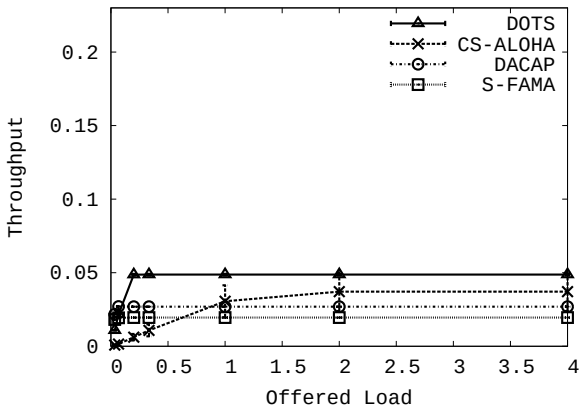


(b) Throughput as a function of offered load with fixed data size (1*kbyte*) and transmission range ($750m$)

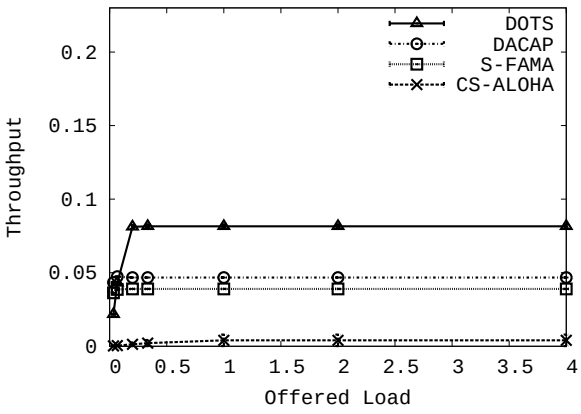


(c) Throughput as a function of offered load with fixed data size (1*kbyte*) and transmission range ($1.5km$)

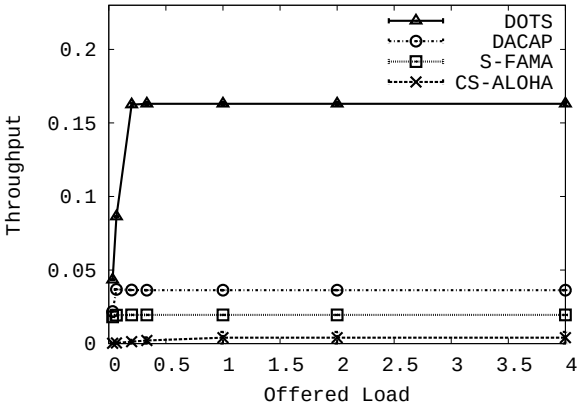
Figure 3.7: Line Topology (Exposed Terminal)



(a) Throughput as a function of offered load with fixed data size (512*bytes*) and transmission range ($750m$)

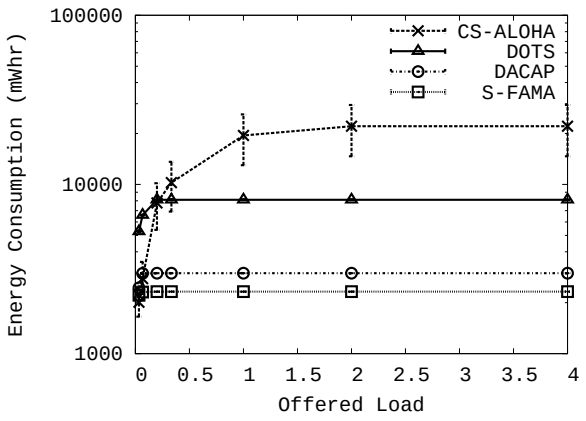


(b) Throughput as a function of offered load with fixed data size (1*kbyte*) and transmission range ($750m$)

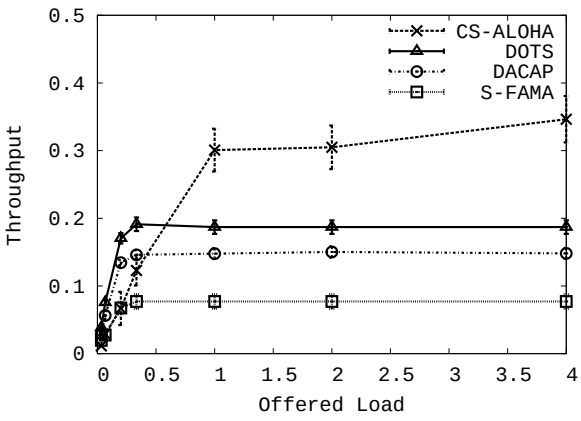


(c) Throughput as a function of offered load with fixed data size (1*kbyte*) and transmission range ($1.5km$)

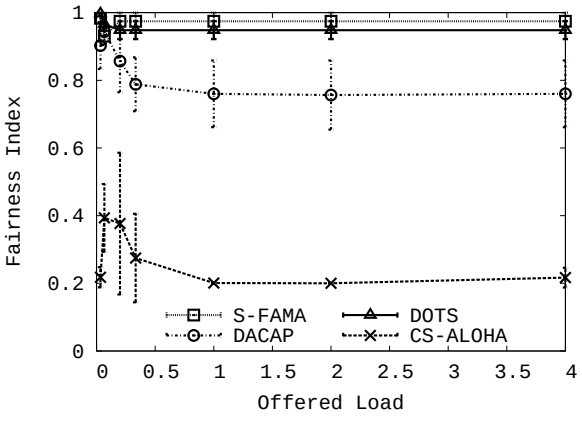
Figure 3.8: Star Topology



(a) Energy Consumption in the star topology with fixed data size (512*bytes*) and transmission range ($750m$)



(b) Throughput as a function of offered load with fixed data size (1*kbyte*) and transmission range ($750m$)



(c) Jain's Fairness Index for the four protocols with a data size of 512*bytes* and a transmission range of $750m$

Figure 3.9: Star and Random Topologies

CHAPTER 4

Hydrocast: A Routing protocol for SEA Swarm Architecture

4.1 Backgrounds

In a SEA swarm, each sensor monitors local underwater activities and reports time-critical data to any one of the sonobuoys using acoustic multi-hopping; then the data are delivered to a monitoring center using radio communications. The main focus of this paper is to design an efficient anycast routing protocol from a mobile sensor to any one of the sonobuoys on the sea level. However, this is challenging due to node mobility and limited resources (bandwidth and energy) of mobile sensors. An underwater acoustic channel has low bandwidth and propagation latency five orders of magnitude higher than the radio channel [88]. Acoustic transmissions consume much more energy than terrestrial microwave communications. Such severe limitations in communication bandwidth coupled with high latency and limited energy make the network vulnerable to congestion due to packet collisions. Under these circumstances, minimizing the number of packet transmissions is important for at least two reasons: minimizing congestion and minimizing energy consumption.

Conventional proactive/reactive routing protocols (e.g., OLSR, AODV, etc.) rely on systematic flooding for route discovery and maintenance, potentially causing excessive energy consumption and collisions. In a SEA swarm scenario, general 3D geographic routing is preferable as it is stateless. However, geographic routing requires online, distributed localization of mobile sensors which is expensive and takes a long time to converge. Also, Durocher et al. [25] showed that efficient recovery from a local maximum may not always be feasible in 3D geographic routing, thus requiring an expensive exhaustive search such as 3D flooding and random walks [31].

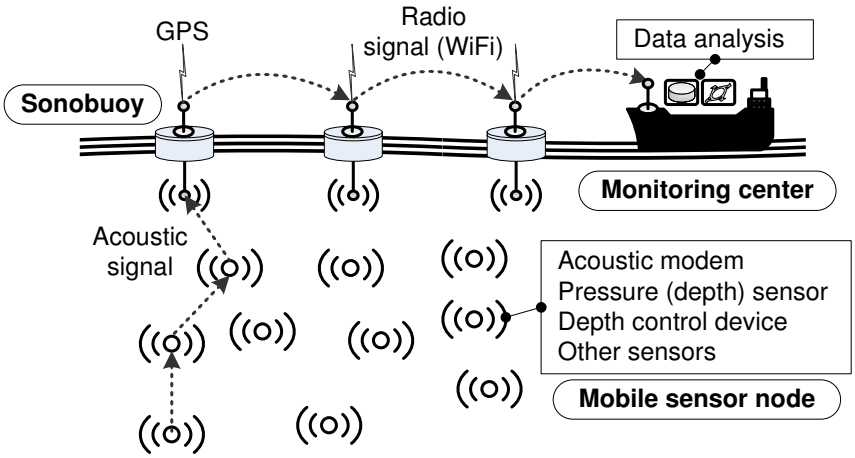


Figure 4.1: SEA Swarm architecture

Fortunately, our georouting problem is specialized in that it is anycast to any buoy on the surface. Thus, it suffices to route a packet upwards to lower depths. Given that the on-board hydraulic pressure gauge can accurately estimate depth (avg. error $<$ 1m [49]), we can use depth information for geographic anycast routing. Yan et al. [99] recently proposed a *greedy* method called Depth Based Routing (DBR) [99] where packet forwarding decisions are locally made based on the measured pressure level (or depth) at each node such that a packet is greedily forwarded to the node with lowest pressure among the neighbors. However, a forwarding node may find no other neighbors with a lower pressure level as it encounters a void region in the swarm. Like face routing in 2D [53], it must fall back to the recovery mode to route the packet around the void, but this was not addressed in [99]. Note that this hydraulic pressure based anycast routing is stateless and does not require expensive distributed localization [18]. In our scenario, the tagging of the sensed data with its location can be performed when the data come to surface. For example, a monitoring center can efficiently perform *off-line localization* using only local neighbor information collected from each node.

The main challenges of hydraulic pressure based routing are the unreliable acoustic channel and the presence of voids, thus requiring efficient greedy forwarding and dead end recovery methods. In this dissertation, we address these challenges and propose a generalized hydraulic pressure based

anycast routing protocol called HydroCast. The following are the key contributions of HydroCast.

We consider wireless channel quality and take advantage of simultaneous packet receptions among one's neighbors to enable opportunistic forwarding by a subset of the neighbors that have received the packet correctly. To suppress hidden terminals, existing forwarding set selections use a heuristic to pick nodes in a geographic region facing the direction towards destination (in our case, upward direction) [37, 107, 99, 52]. We show that these approaches do not maximize the expected progress toward the destination, and in general, finding such a set is computationally hard. Thus, we propose a simple greedy heuristic that searches for a cluster of nodes with maximum progress and without hidden terminal problems, using only local topology information. Our simulation results validate that the proposed approach can find a set whose expected progress is very close to that of the optimal solution.

We then propose an efficient recovery method with delivery guarantee. The key idea is that a node can tell whether it is on the local maximum because only depth information is used for routing; i.e., a local maximum happens when there is no neighboring node with lower depth than the current depth. In our scheme, each local maximum node maintains a recovery route to a node whose depth is lower than itself. After one or several path segments that go through local maxima, a packet can be routed out of the void and can switch back to the greedy mode. Since any nodes located beneath the void area can potentially suffer from the void and opportunistic forwarding along the recovery path is feasible, our approach is more efficient than a random walk-based approach [31]. For efficient route discovery, we propose a route discovery method that implements hop-limited 2D flooding over the surface of void regions, which is a major improvement over simple 3D flooding.

4.2 Related Work

Geographic routing under channel fading: In geographic routing, a packet is greedily forwarded to the closest node to the destination in order to minimize the average hop count. Due to channel

fading, however, the further the transmission range, the higher the attenuation, and the more the likelihood of packet loss. Researchers have tried to incorporate the associated cost such as the number of transmissions and energy consumption in geographic routing [56, 77]. For instance, Lee et al. [56] proposed a generalized link metric called Normalized Advance (NADV) where the amount of progress is normalized by its associated cost. However, these protocols did not consider the simultaneous packet receptions by one's neighbors and their ability of opportunistic packet forwarding by scheduling the set of nodes that received the packet correctly based on their distances (or associated costs) to the destination [10, 24, 107, 37].

A key design issue of opportunistic routing is the selection of a subset of neighbors that can make the best progress toward the destination, yet without the hidden terminal problem: i.e., when a higher priority node transmits a packet, other low priority nodes should be able to suppress forwarding to prevent redundant packet transmissions and collisions. Most opportunistic routing protocols (also called anypath routing) such as ExOR [10], Least Cost Opportunistic Routing (LCOR) [24] that do not use geographic information, require global topology and link quality information (like link state routing) to find a set of forwarding groups toward the destination; thus, they are more suitable for *static* wireless mesh or sensor networks. In practice, geographic routing can also benefit from opportunistic forwarding as in Geographic Random Forwarding (GeRaF) [107], Contention Based Forwarding (CBF) [37], and Focused Beam Routing (FBR) [52], though not optimal due to lack of global knowledge. In the literature, researchers typically used a geometric shape (e.g., a triangle/cone [37, 52]) that is *faced toward the destination* for forwarding set selection to prevent hidden terminal problems. The notion of the expected progress of opportunistic forwarding (in meters), called Expected Packet Advance (EPA) was recently established by Zeng et al. [100]. However, none of the previous works [107, 37, 100, 52] attempted to find a forwarding set with the maximum EPA and without the hidden terminal problem. In this dissertation, we show that finding such a set is a variant of the maximum clique problem, which is computationally hard, and thus propose a simple greedy heuristic method that well approximates the optimal solution.

Geographic routing recovery mode: The recovery mode in geographic routing can be classified

as stateful or stateless. In 2D, face routing [53] is a widely used stateless (memoryless) strategy. The basic idea is to planarize a network graph using a simple local method and to forward a packet along one or possibly a sequence of adjacent faces, thus providing progress towards the destination node. To our surprise, for 3D networks it has been shown that there is no *local* memoryless routing algorithm that delivers messages deterministically as in 2D face routing [25]. Based on this observation, Flury et al. [31] proposed a randomized geographic routing using random walks. Nodes in the network are arranged in a virtual 3D grid coordinate using a localized algorithm where each grid point is a cluster of nodes in close proximity. A random walk is then performed on this virtual coordinate.

There are several stateful approaches proposed in the literature [60, 61, 103]. Greedy Distributed Spanning Tree Routing (GDSTR) [60] uses a spanning tree where each node has an associated convex hull that contains within it the locations of all its descendant nodes in the tree. A node exhaustively searches the tree for recovery by traversing sub-trees one by one. Liu et al. [61] proposed a backtracking method over a virtual coordinate system where a packet is routed towards one of the anchors (used to build the virtual coordinate system), hoping that it can switch back to the greedy mode on its way. Geo-LANMAR [103] inherits the group motion support of Landmark Routing (LANMAR) that dynamically elects cluster-heads (landmark nodes). It circumvents a void in the network using the topology knowledge of landmark nodes as in [61]. In this dissertation, given the unique characteristic of our scenario where any nodes located beneath the void area can potentially suffer from the void, we consider keeping some state to reduce the recovery overhead (preventing expensive random walks to overcome the same void) and to exploit opportunistic packet forwarding along recovery paths.

4.3 Problem Statement

Instead of attacking generalized 3D geographic routing which requires an expensive distributed localization due to slow convergence speed, we propose to use 1D geographic anycast routing in a

single (vertical) direction to the surface of the ocean using the depth information from a pressure sensor.[1] This routing simplification is justified by the fact that in our scenario communications are strictly *vertical*, from sensors to surface nodes. The need for global, distributed localization is relaxed via off-line localization at a monitoring center which uses local distance measurements (collected along with sensor data). Given this, the problem boils down to exploiting the opportunistic packet receptions under channel fading and developing an efficient recovery mechanism from a local maximum.

4.3.1 Forwarding set selection

Due to channel fading, the further the distance, the higher the signal attenuation and the more the likelihood of packet loss. We need to normalize the progress by its associated cost, which can be represented using Normalized Advance (NADV) [56]: for a given node, NADV to a neighbor node n that has the packet delivery probability p_n and the progress to the destination d_n^P (in meters) is given as $\frac{d_n^P}{1/p_n} = d_n^P \times p_n$. NADV can be extended to opportunistic forwarding as well. All neighboring nodes who receive the packet will access their priority based on how close they are to the destination; i.e., the closer to the destination, the higher the priority. A node will forward the packet when all the nodes with higher progress to the destination fail to send it. This can be easily scheduled by setting a back-off timer proportional to the distance to the destination. Since nodes can hear each other, those nodes with lower priorities will listen to the packet (either data or ACK packet) transmitted by a higher priority node and suppress their transmissions, thus excluding the possibility of collisions and redundant packet transmissions. Assume that source S has a set of k neighboring nodes Γ_k ordered based on their priorities as $n_1 > n_2 > \cdots > n_k$. The Expected Packet Advance (EPA) is simply the normalized sum of advancements made by this neighboring set [100]. The highest priority contributes on average $d_{n_1}^P p_{n_1}$ (=NADV). Since the next node can

[1] Note that distributed localization typically requires many iterations, each of which takes a considerable amount of time because of large propagation delay and limited bandwidth underwater (often exacerbated by node mobility). We confirm this in the extended version of this paper and show that the overhead is closely related to the localization accuracy requirement [58].

only contribute if the highest node fails, its contribution is $d^P_{n_2} p_{n_2}(1 - p_{n_1})$. In general, the EPA is given as:

$$EPA(\Gamma_k) = \sum_{i=1}^{k} d^P_{n_i} p_{n_i} \prod_{j=0}^{i-1} (1 - p_{n_j}) \tag{4.1}$$

The above equation shows that as long as a node can make a positive advancement, we can include it to maximize the EPA, but we realize that including too many nodes may result in the hidden terminal problem that leads to redundant transmissions and packet collisions. Since the node degree is higher in 3D than 2D networks, 3D networks have higher probability of suffering from hidden terminals collisions than 2D networks [75]. Despite the fact that minimizing the number of transmissions in resource constrained sensor networks is one of the most important design criteria, none of the existing solutions [37, 107, 100, 52] consider the EPA metric and the hidden terminal problem simultaneously. However, the challenge is that finding such a forwarding set is a variant of the maximal clique problem that finds the largest clique in a graph, which is computationally hard; i.e., to be precise, find a clique with maximal EPA. Recall that a clique in a graph is an induced subgraph which is complete (i.e., every node can hear one another). As a simple heuristic, we could use a geometric volume, say a cone with the vertex on the transmitter and the base facing the direction to destination, which is a 3D extension of 2D methods reported in [37, 107, 52]. The problem is that they often fail to maximize EPA, as illustrated in Figure 4.2. In this dissertation, we propose simple heuristics that search for a cluster of hidden-terminal-free nodes that maximizes EPA, using only local topology information. Also, we validate that our approach can find a set whose EPA is very close to that of the optimal solution.

4.3.2 Geographic routing recovery mode

After it was reported that for 3D networks there is no *local* memoryless routing algorithm that delivers messages deterministically [25], the state-of-the-art recovery scheme proposed in the field is a randomized geographic routing protocol using random walks [31]. However, we claim that this randomized approach may not be suitable for our scenario where nodes need to periodically send

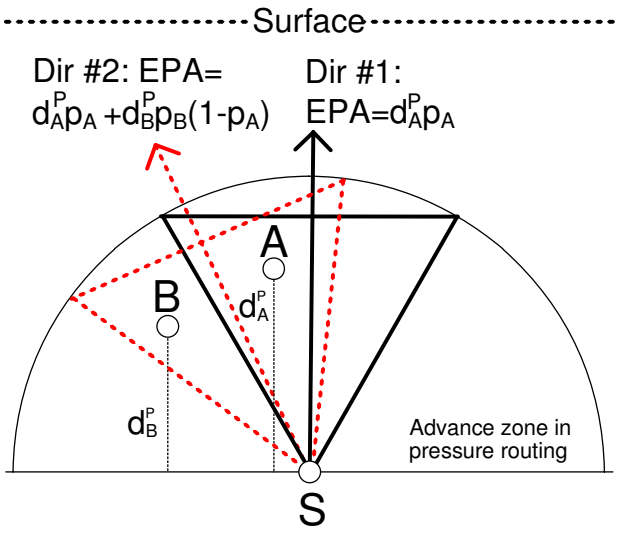


Figure 4.2: Impact of direction in pressure routing (Dir #2 > Dir #1)

their local coordinate information (for off-line localization) and sensor data to the surface nodes. Since nodes vertically forward packets to the surface, any nodes located beneath the void area can potentially suffer from the void, and every packet originating that area has to perform an expensive random walk to overcome the same void. The overall amortized cost will be very high. Besides, it is not clear how we can exploit the opportunistic packet forwarding using random walks.

To these reasons, we opt for a stateful approach as in [43, 60, 61, 103]. The main departure from existing methods is that a node can easily tell whether it is on a local maximum by checking its neighbors' pressure level; i.e., it is on the local maximum if there is no neighboring node with a lower pressure level. If we assume that every local maximum node has a recovery route to a node whose depth is lower than itself (either another local maximum or a non-local maximum node where greedy forwarding can resume), the scheme successfully recovers from the voids. Namely, after one or several path segments that go through local maxima, the packet can be routed out of the void and can switch back to the greedy mode. Then, the key step is to efficiently find the recovery routes. The brute force approach is 3D flooding: i.e., local maximum nodes perform hop-limited 3D flooding until they find better escape nodes. In our scenario, we note that the route discovery overhead can be significantly reduced via *route discovery over the void floor surface* using 2D

flooding. However, the challenge is to detect whether a node is on the void floor surface. In this dissertation, we present an efficient localized void surface floor detection algorithm and show that the aforementioned local *lower-depth-first* recovery method guarantees packet delivery.

4.4 Forwarding Set Selection

4.4.1 Packet delivery probability estimation

We use the following underwater acoustic channel model to estimate delivery probability [88, 62]. The path loss over a distance d for a signal of frequency f due to large scale fading is given as $A(d, f) = d^k a(f)^d$ where k is the spreading factor and $a(f)$ is the absorption coefficient. The geometry of propagation is described using the spreading factor $(1 \leq k \leq 2)$; for a practical scenario, k is given as 1.5. The absorption coefficient $a(f)$ is described by the Thorp's formula [62].

The average Signal-to-Noise Ratio (SNR) over distance d is thus given as

$$\Gamma(d) = \frac{E_b/A(d, f)}{N_0} = \frac{E_b}{N_0 d^k a(f)^d} \tag{4.2}$$

Here, E_b and N_0 are constants that represent the average transmission energy per bit and noise power density in a non-fading additive white Gaussian noise (AWGN) channel. As in [88, 12], we use Rayleigh fading to model small scale fading where SNR has the following probability distribution

$$p_d(X) = \frac{1}{\Gamma(d)} e^{-\frac{X}{\Gamma(d)}} \tag{4.3}$$

The probability of error can be evaluated as

$$p_e(d) = \int_0^\infty p_e(X) p_d(X) dX \tag{4.4}$$

where $p_e(X)$ is the probability of error for an arbitrary modulation at a specific value of SNR X. In this dissertation, we use BPSK (Binary Phase Shift Keying) modulation that is widely used in the state-of-the-art acoustic modems [32]. In BPSK, each symbol carries a bit. In [79], the probability

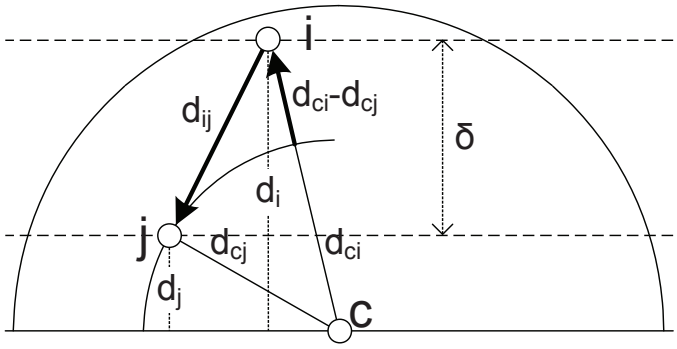


Figure 4.3: Timer scheduling for prioritization

of bit error over distance d is given as

$$p_e(d) = \frac{1}{2}\left(1 - \sqrt{\frac{\Gamma(d)}{1+\Gamma(d)}}\right) \tag{4.5}$$

Thus, for any pair of nodes with distance d, the delivery probability of a packet with size m bits is simply given as

$$p(d, m) = (1 - p_e(d))^m \tag{4.6}$$

4.4.2 Packet forwarding prioritization

We use a distance based timer to prioritize packet forwarding where the distance denotes the progress toward the surface. When the current forwarder broadcasts a packet, nodes that receive the packet set the timer such that the greater the progress, the shorter the timer. Among those who receive the packet, the highest priority node becomes a next hop forwarder. The rest of lower priority nodes then suppress their packet transmissions after listening to the next hop forwarder's data or ACK packet.[2]

Unlike [107, 37], we define a linear timer function of a receiver x, which is customized for acoustic communications, as $f(d_x^P) = \alpha(R - d_x^P)$ where α is a constant, R is the maximum progress (i.e., transmission range), and d_x^P is progress of a receiver. Consider two nodes i and j

[2]Note that compared to a short ACK packet, a passive ACK (i.e., overhearing a data packet) is unreliable due to channel fading/collision.

with progress d_i^P and d_j^P respectively (see Figure 4.3). If we have $d_i^P > d_j^P$, we want to guarantee that $f(d_i^P) < f(d_j^P)$. Assuming that we use an ack for suppression, the timer function has to satisfy the following inequality: $t_{ci} + f(d_i^P) + t_{ij} + t_{ack} < t_{cj} + f(d_j^P)$ where t_{ab} is the propagation delay from node a to node b and t_{ack} is the transmission delay of an ACK packet. By plugging in $f(d^P)$, we have the following:

$$\alpha > \frac{t_{ci} - t_{cj} + t_{ij} + t_{ack}}{d_i^P - d_j^P} \tag{4.7}$$

The numerator is the sum of the propagation delay to travel $d_{ic} - d_{jc} + d_{ij}$ and the ack transmission delay, as shown in the figure (thick arrows). The progress difference between two nodes ($d_i^P - d_j^P$) is critical; if it is too small, the constant α will be very large, thus resulting a very long delay. For a given candidate forwarding set, we can find the α by examining every pair using local topology information, which takes $O(n^2)$ steps where n is the number of neighbors. However, α may be too large. To handle this case, we have a system parameter that sets the maximum allowable delay per hop, denoted γ. In the following, we address how we choose a forwarding set that satisfies the delay constraint.

4.4.3 Forwarding set selection methods

Nodes in the forwarding set must hear each other to prevent hidden terminal collisions. 3D networks have a higher probability of suffering from collisions than 2D networks, because for equal connectivity the node degree is higher in 3D than 2D networks. At the same time, we want to maximize progress (i.e., EPA). As discussed earlier, finding the optimal set is computationally hard and thus, we propose a simple clustering heuristic, inspired by the Multi-Point distribution Relay (MPR) selection in OLSR [46]. To this end, the current forwarder C requires the knowledge of 2-hop connectivity and neighboring nodes' pairwise distances. We assume that each node measures the pair-wise distance via Time of Arrival (ToA),, which is widely used in underwater networks [18], and the data are periodically reported to the surface for off-line localization. We take advantage of such periodic reports to obtain 2-hop neighbor information.

We assume that node C has computed the NADV of each neighbor as a forwarder *upwards*

to surface. Like MPR selection where a node that covers the highest number of nodes is greedily picked, we use a simple greedy approach. The greedy clustering starts from the highest NADV neighbor, say S. Node S acquires all the other neighbors (of C) at distance $< \beta R$, where β is a constant ($\beta < 1$) and R is the acoustic range. In our design, we simply use $\beta = 1/2$ so that all the nodes clustered by S can hear each other. Then, if other neighbors are left, clustering proceeds starting from the highest value remaining neighbor and so on, until no nodes left. After this, each cluster is expanded by including all the additional nodes such that the distance between any two nodes in the cluster is smaller than R. This condition guarantees that nodes in the set can hear each other. We repeat this for all other clusters in turn and find the cluster with the highest EPA. Note that for a given set, we can find the minimum α value for priority scheduling, and this should be smaller than the maximum allowable delay per hop (γ). Thus, we remove one of the nodes with a lower NADV when detecting $\alpha > \gamma$ during the clustering process.

As an alternative, we can use a *cone shape* (3D counterpart of a Reuleaux triangle) to select a forwarding set. Unlike existing approaches [37, 107] that always orient a geometric contention shape along the line between source and destination, we need to search for the forwarding direction that maximizes the EPA. This requires local topology information – given n neighboring nodes with their depth information and pairwise distances, it is a realization of a graph with n nodes whose edges are weighted based on distance. We find the local topology using the Sweep algorithm that is known to work well for both sparse and dense networks [40]. In Sweep, we start from three known vertices and localize each neighboring node one at a time by computing all possible positions consistent with neighbor positions via a series of bilaterations until all vertices are localized. Using the local topology information, we discretize the 3D space into a unit degree of θ, thus generating total $2\pi^2/\theta^2$ directions over the hemisphere (advance zone). We then linearly scan each direction and calculate the EPA to find the direction with maximum EPA.

After forwarding set selection, we need to include the chosen forwarding set in the data packet. To reduce the overhead, we use a Bloom filter, a space efficient membership checking data structure. The membership checking is probabilistic and false positives are possible, but we can bound

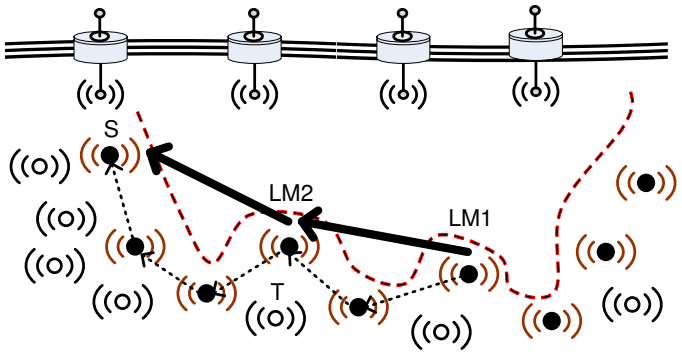


Figure 4.4: Recovery mode

the probability of false positive by properly adjusting the filter size. In a practical scenario, the set size will be smaller than 15 (in the hemisphere advance zone). Fan et al. [30] showed that a filter size of 150 bits (19B) to represent 15 items has a false positive rate smaller than 1%. We can also include sender's pressure level and max/min angle information to filter out quite a few of neighboring nodes that are not in the forwarding set. Furthermore, noting that there could be many other packets that have to travel through a certain node, and topology slowly changes over time, we may only need to include the set information in the data packet whenever there is a sufficient change. Thus, the amortized overhead could be much smaller.

4.5 Recovery Mode

We present a local lower-depth-first recovery method that guarantees the delivery and provide an efficient recovery route discovery method using 2D surface flooding, instead of expensive 3D flooding. Note that opportunistic forwarding over a recovery path is illustrated in the extended version of this paper [58].

4.5.1 Local lower-depth-first recovery

Unlike traditional geographic routing where the local maximum is determined by the location of a destination node, in our scenario, each node can easily tell whether it is on the local maximum

by checking its neighbors; i.e., a node is on the local maximum if there is no neighboring node with lower pressure level. Having said that, we propose a lower-depth-first recovery method as follows. Every local maximum node searches for a node whose depth is lower than its current depth and explicitly maintains a path to the node (via some route discovery method). This node could be another local maximum where there is a new recovery path, or the point where the greedy forwarding can be resumed. Whenever a packet hits a local maximum, it is re-routed along the recovery path either safely to a node that can resume greedy forwarding or to a new local maximum. In Figure 4.4, for instance, we have two local maxima, namely $LM1$ and $LM2$. $LM1$ maintains a path to $LM2$ which has a path to node S. A packet can be routed from $LM1$ to $LM2$ to S. It then can be switched back to the greedy mode and can be delivered to a node on the ocean surface. In practice, we can recover from the local maximum after a few iterations.

The following theorem proves the delivery guarantee and loop-free property of lower-depth-first routing. Local lower-depth-first routing is loop-free and guarantees the packet delivery.

Proof. Consider a local maximum graph $G = (V, E)$. A vertex $v \in V$ in the graph represents a local maximum node, and two vertices are connected if there is a recovery path. There is also a sink vertex that can reach the surface. If each vertex (local maximum) can reach the surface directly without visiting into other local maxima, it is connected to the sink. Assume that a packet arrives at a local maximum, say vertex v_i. If v_i is connected to the sink, the packet is safely delivered. Otherwise, it will be re-routed to another local maximum (say v_j) whose depth is lower than the current depth by definition; i.e., $D(v_i) > D(v_j)$ where $D(v_k)$ returns the depth of node v_k. Since the distance to the surface decreases in each step, a packet can be delivered after a finite number of steps that is strictly less than the total number of local maxima. This monotonic behavior also guarantees that there is no loop. □

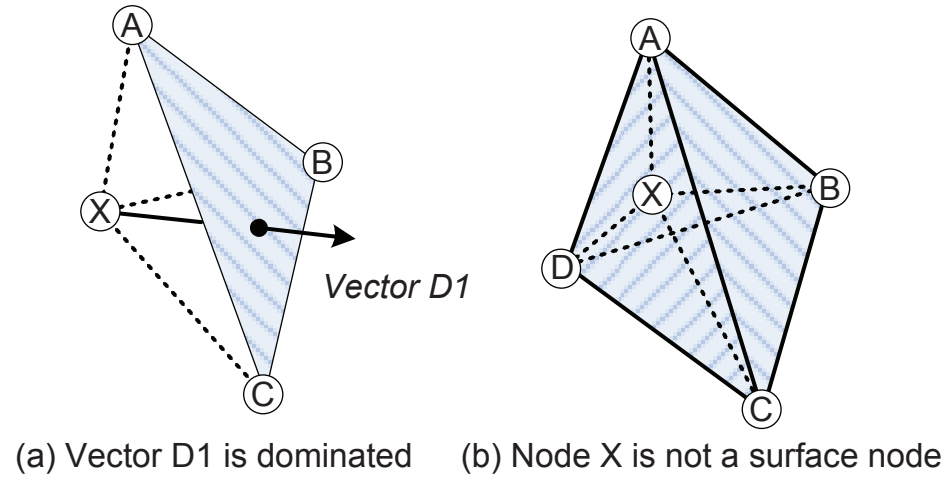


Figure 4.5: Illustration of domination and non-surface node

4.5.2 2D void floor surface flooding for recovery path search

Now the important step is to find the recovery route. The brute force approach is 3D flooding; i.e., nodes at the local maxima perform *expensive* hop-limited 3D flooding to discover escape nodes where greedy mode can resume or there are recovery paths to better escape nodes. This brute force approach is not deemed suitable because the appropriate *scope* of the limited 3D flooding is difficult to estimate and the 3D flood can degenerate to involve all nodes in the sensor mesh. To improve efficiency, we use *2D flooding on the void floor surface*. This flood involves a much more manageable set of nodes. Figure 4.4 illustrates the approach in 2D. Nodes on the envelope (or surface) can become aware of their void floor surface status using local connectivity information and thus forward the packet. Nodes that are dominated by surface neighbors are not on the surface and refrain from forwarding. For instance, node S does not have any nodes on its right and is a surface node. Node T is surrounded by its neighboring nodes and it is not on the surface. We now formally define *domination* and a *void surface node* in 3D environments.

For a given node, a random vector emanating from the node is dominated if and only if there is a dominating triangle formed by the node's neighbors that intersects with the vector. A node is on the surface if and only if there exists a vector that is not dominated (no dominating triangle for

the vector).

Consider Figure 4.5(a). Random vector $D1$ emanating from node X is dominated because it intersects with the triangle ABC. Any random vector pointing inside the tetrahedron $XABC$ is dominated by the triangle. In Figure 4.5(b), node X is completely surrounded by a set of tetrahedra that dominates every possible direction. Thus, node X is a non-surface node.

Surface node detection can be formally described as follows. Consider the point set P in 3D Euclidean space where the set is composed of node X and its neighbors. Any pair of points in the set is connected if their distance is less than or equal to the transmission range. The point set centered at node X is now normalized (as a unit vector), thus lying on the surface of a unit ball. The connectivity among points in the normalized point set does not change. Given that we have a normalized point set and connectivity information among set members, the surface node detection is to decompose the point set into a set of non-overlapping tetrahedra that exhaustively cover the unit ball. It is known that the length constraints (due to the communication range) make these kinds of tetrahedralization problems intractable [82, 36]. If there is no length constraint, the problem becomes a decomposition of the convex hull of P into non-overlapping tetrahedra, which can be solved in $O(n \log n)$ where n is the point set size [27].

In this dissertation, we propose a simple Monte Carlo approximation method: pick k random directions and check whether there is a dominating triangle for each direction. The number of directions k should be large enough to correctly identify the surface node. Otherwise, void floor surface detection may fail, generating a false negative – a surface node is declared as a non-surface node. We detail our approximation method as follows. We first generate a set R of k random vectors. There are $O(n^3)$ triangles that can be formed by node X's neighbors. For each triangle, we repeat the following procedure: check all vectors in the set R to know whether they are dominated by the triangle and remove the dominated vectors from the set R. If R becomes empty, we declare that node X is not on the void floor surface. Otherwise, our algorithm declares that node X is on the void floor surface. Thus, the worst case complexity is given as $\Theta(kn^3)$. Note that the detection algorithm is localized, requiring only 1-hop topology which can be constructed using

only periodic beacons. Thus, it does not cause any additional packet exchanges. Moreover, the processing overhead is minimal because it is triggered only when nodes detect that local network topology has sufficiently changed.

The accuracy of this method can be analyzed as follows. Let X be a node that is on the void floor surface. The volume of a sphere centered at node X is given as $\frac{3}{4}\pi R^3$ where R is the transmission range. Assuming that the volume of a void area that intersects with the sphere is x, a random vector hits the void area with probability $p = \frac{4x}{3\pi R^3}$. A false negative happens when all k random vectors miss the void. Thus, the probability of a false negative is given as $(1-p)^k$, thus exponentially decreasing with k. In practice, the volume size of a void is large enough, and we can achieve high accuracy with small k. For instance, when the intersecting void volume is one fifth of the sphere ($p = 1/5$) and $k = 20$, the probability of a false negative is about 1%.

So far we assume that a void area always causes the local maximum. In our pressure routing, interestingly not every void area causes the local maximum. Such void areas are usually located inside the swarm (à la the air bubbles in bread dough), and greedy forwarding can successfully bypass the void areas (say an egg-shape area). We call this kind of void a *bubble*. The size of a bubble is closely related to node density; as node density increases, there will be fewer bubbles whose sizes are also diminishing. Figure 4.5(b) shows that we need at least four nodes in order not to rule out a surface node. Given that a well connected 3D network requires each node to have around 30 neighbors (in 2D, 15 neighbors) [75], these bubbles will likely happen particularly when node density is very low. In practice, those nodes on the bubble surface will not cause a problem. The special case that needs our attention happens when a bubble *contacts* the real void floor surface. Under this circumstance, those nodes will receive route discovery packets and also participate in the flooding process, causing redundant packet transmissions. To prevent this, nodes should be able to tell whether the void area is a bubble or not. However, this is an expensive process, requiring more than 2-hop information. It is part of our future work to investigate how this case can be efficiently handled.

4.6 Simulations

In this section, we evaluate the proposed approaches via simulations using QualNet. First, we investigate the forwarding set selection to answer: how important is the hidden terminal problem?; and how good are our forwarding set selection heuristics? Second, we evaluate the recovery mode to answer: how often is a packet trapped in a local maximum for varying node density?; and how effective is our proposed void surface detection scheme? Finally, we compare the performance of various depth-based routing strategies (e.g., different forward set selection methods and recovery modes).

4.6.1 Simulation setup

For acoustic communications, the channel model in Section 4.4.1 is implemented in the physical layer of QualNet. We generate different channel fading conditions in the simulations by adjusting the transmission power in dB re μ Pa and SNR threshold [88].[3] Unless otherwise mentioned, the transmission power is set to 105 dB re μ Pa. We use a transmission range of 250m, and the data rate is set to be 50Kbps as in [105]. We use the CSMA MAC protocol. In CSMA, when the channel is busy, a node waits a back-off period and senses carrier again. Every packet transmission is MAC layer broadcasting. For reliability, we implement ARQ at the routing layer as follows. After packet reception, the receiver sends back a short ACK packet. If the sender fails to hear an ACK packet, a data packet is retransmitted; and the packet will be dropped after five retransmissions.

We randomly deploy varying numbers of nodes ranging 100 to 450 in 3D region of size 1000m $\times$ 1000m $\times$ 1000m. Due to mobility, nodes would move beyond this region. They move according to an extended 3D version of the Meandering Current Mobility (MCM) Model [13], an underwater mobility model that considers the effect of meandering sub-surface currents (or jet streams) and vortices. In the model, we set the main jet speed to 0.3m/s.

Each node measures the distance to its neighbors every 30 seconds (with random jitters to

[3]The signal intensity is measured in dB re μPa of the power flux $[Wm^{-2}]$ delivered into the water by a source.

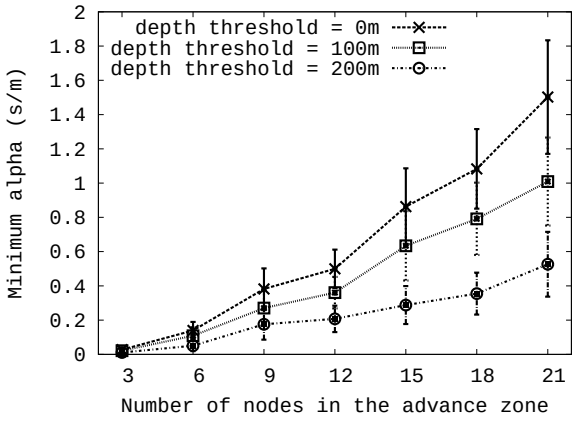


Figure 4.6: Minimum alpha value

prevent synchronization) and broadcast the measured information to its one hop neighbors. Every 60 seconds, each node reports the sensed data and distance measurements up to the surface. Note that a node in the main jet stream will have moved 20m in 60s. With a range of 250m, we expect the 60s refresh rate to be adequate to track topology changes for off-site localization. The size of a packet is a function of the number of neighbors, and the average packet size is less than 200B in our simulations. We measure delivery ratio, delay and overhead. The delivery ratio of a source is the fraction of the packets delivered; the delay is the time for a packet to reach any of the sink nodes on the surface; and the overhead is measured in terms of the total number of packet transmissions. In our simulations, each run lasts 3600s. Unless otherwise specified, we report the average value of 50 runs with the 95% confidence interval.

4.6.2 Simulation results

In Section 4.4, we show that the forwarding set selection and its prioritization must be properly done to prevent the hidden terminal problem. Otherwise, there will be redundant transmissions and collisions. To show the impact, we evaluate a simple forwarding set selection method proposed in DBR (Depth Based Routing) [99]. Recall that DBR is the first underwater routing scheme to exploit pressure (and thus depth) awareness at each node for routing packets to surface. It

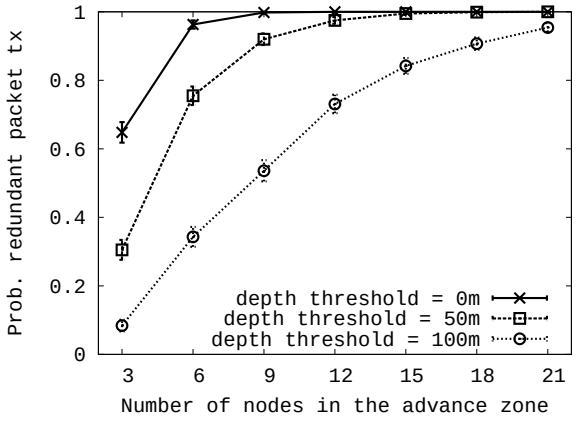


Figure 4.7: Probability of a redundant packet transmission

implements a basic greedy forwarding design with an opportunistic forwarding flavor. All the nodes higher than the current forwarder by more than a depth threshold (h) act as opportunistic forwarders. Moreover, DBR uses a fixed holding timer at each hop. We randomly deploy varying numbers of nodes in the hemisphere (3-21 nodes). For a given configuration, we calculate the minimum α value using Equation 4.7 using three different minimum depth thresholds (h=0m, 100m, 200m). We plot the average α value of 1000 random configurations with 95% confidence interval. Figure 5.6 clearly shows that as density increases, it is more likely that two nodes are in close proximity (or have low depth difference), and thus, the average minimum alpha value significantly increases. For example, the alpha value of 1 can result in the maximum delay of 250s in our scenario. In Figure 4.7, we plot the probability of redundant packet transmissions caused by the hidden terminal problem. We see that the hidden terminal problem persists even with a high depth difference value. For instance, a 10 node scenario has more than a 60% chance of redundant packet transmissions with depth difference of 100m. Therefore, it is mandatory to suppress such redundant transmissions.

We now evaluate how good our forwarding set selection algorithm is. In Figure 4.8, we plot the progress (EPA) of the different forwarding set selection schemes: optimal, cone-based, clustering, and simple NADV. In the optimal scheme, we perform exhaustive search on the neighbor set to find the maximum EPA. NADV denotes the case where we only choose the node whose NADV is the

largest (i.e., a single node in the forwarding set). Cone-Vert only considers the vertical direction as in [37, 107]. The figure shows that our clustering method is very close to the optimal solution, outperforming the cone based approaches. The results also show that the more the number of nodes, the higher the EPA (as expected).

We measure the fraction of local maximum nodes over time under MCM mobility (half a day). Since this is closely related to the node density, we vary the number of nodes ranging from 100 to 400 nodes. For a given configuration, we sample the number of local maximum nodes every 1.7 hours. Figure 4.9 shows the results. When the node density is low, the fraction of local maximum nodes is high. As time passes, the fraction of local maximum nodes increases. This is due to the fact that nodes tend to disperse over the simulated area (beyond the original 1000x1000x1000 cube) due to ocean currents (i.e., jet streams and vortices).

We analyze the accuracy of our proposed void floor surface detection method. We measure the fraction of the surface nodes detected by varying the number of nodes in the network (100-400) and the number of random vectors ($k = 1 - 1000$). For the sake of clarity, we divide the number of detected surface nodes by that of the case with $k = 10,000$. Figure 4.10 shows the results. The lower the density, the higher the detection probability because the area that intersects with the void is larger (i.e., larger p). As the number of random vectors increases, the detection probability approaches to 1. The figure shows that the detection probability is over 95% with $k = 20$.

Finally, we compare the performance of HydroCast with DBR under different settings. Recall that DBR implements a basic greedy forwarding design with an opportunistic forwarding flavor and uses a fixed holding timer at each hop. HydroCast uses a more elaborate opportunistic forwarding strategy and supports also recovery from voids. To show the benefit of our 2D surface flooding (denoted as SD-R), we also implement a simple angle-based selection heuristic: i.e., when a node X broadcasts a route discovery packet, any neighboring node A whose adjacent angle formed by the X-axis and XA is less than 60 degrees participates in the flooding (denoted as SD-A). Figure 4.11 shows the packet delivery ratio. When node density is low, DBR has higher delivery ratio than HydroCast without recovery. Unlike HydroCast, DBR does not suppress redundant packet

transmissions and thus, it is likely to deliver packets on multiple paths, improving reliability. The same figure also reports the plot for HydroCast with forwarding set selection and recovery. We note that recovery support from voids significantly improves the reliability of HydroCast and puts it above DBR. Accurate surface detection helps us to achieve better PDR, because angle-based selection may not include some of the surface nodes, failing to find the recovery path (especially, when density is low). In Figure 4.12, we plot the average number of packet transmissions to deliver a data packet, including the recovery process. Due to redundant packet transmissions and multi-path packet delivery, DBR results in significantly more number of transmissions than other schemes. Interestingly, the impact of recovery reduces as the density increases. This is because there will be fewer voids and fewer hops to switch back to the greedy mode, and more number of nodes is involved in packet forwarding; thus, the amortized recovery cost decreases. In the case of angle-based selection, the overall overhead remains the same, because there will be more redundant packet transmissions as density increases (fewer voids, but much higher costs). Finally, Figure 4.13 shows that HydroCast has lower end-to-end delay than DBR thanks to HydroCast's adaptive timer setting at each hop. As density decreases, the average delay in HydroCast slightly increases because of the increased frequency of voids requiring recovery and thus longer paths.

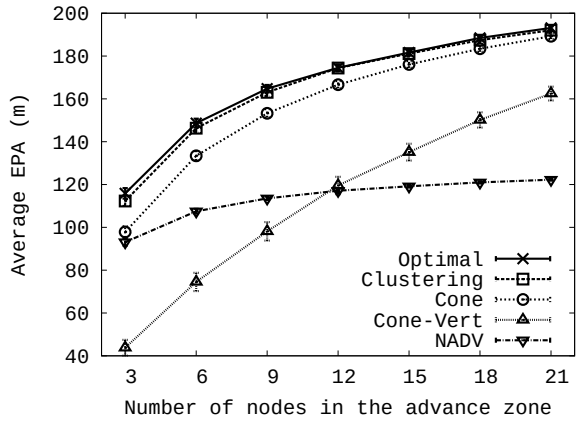


Figure 4.8: EPAs of different forwarding set selection schemes

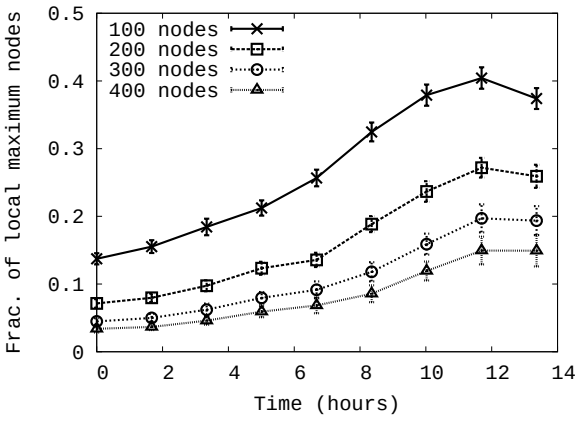


Figure 4.9: Fraction of local maximum nodes over time

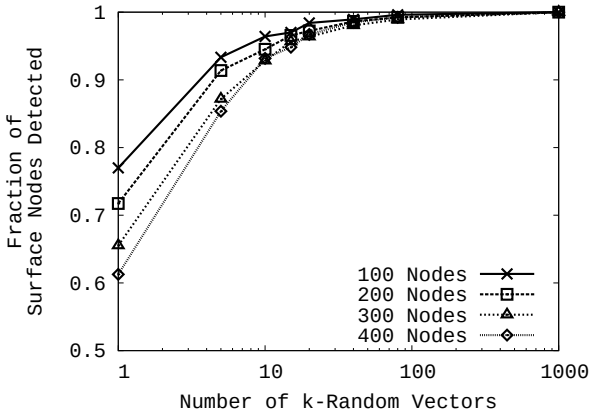


Figure 4.10: Void floor surface detection using a Monte Carlo method

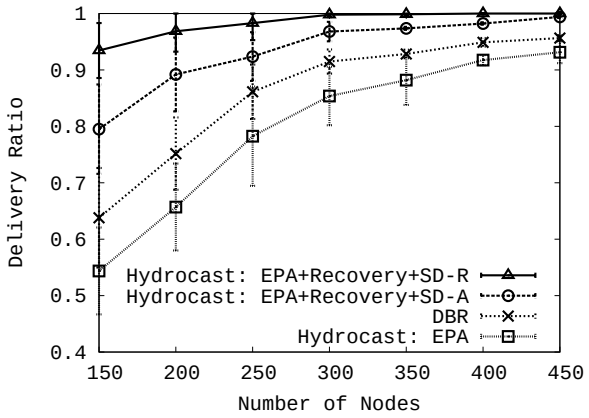


Figure 4.11: Packet delivery ratio

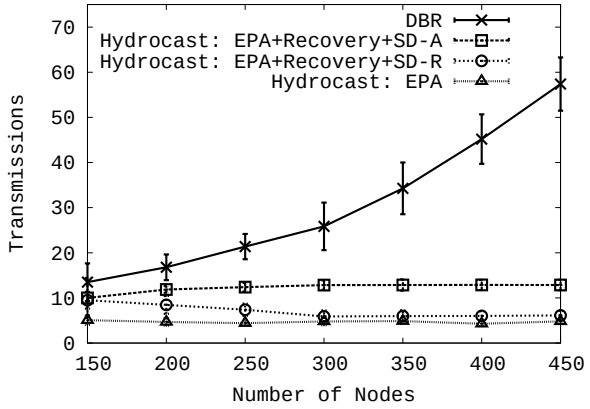


Figure 4.12: Number of transmissions for delivery

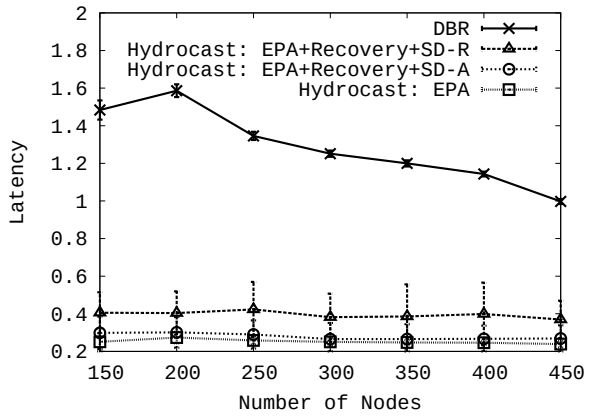


Figure 4.13: Average end-to-end delay

CHAPTER 5

VAPR: Void Aware Pressure Routing for Underwater Sensor Networks

5.1 Backgrounds

In a SEA Swarm, each node is equipped with a variety of sensors and a low bandwidth acoustic modem. Moreover, each node has a fish-bladder like apparatus and a pressure gauge, and its depth can be configured when deployed (e.g., Drogues [47]). A swarm of sensor nodes is escorted by sonobuoys on the sea surface, where sonobuoys are equipped with both acoustic and radio modems (Wi-Fi or satellite communications) and GPS. Each sensor node in the swarm reports relevant data to any one of the sonobuoys with acoustic multi-hop routing (called *anycast routing*); the data can then be offloaded to a monitoring center via radio communications for further off-line processing. In a GPS-denied underwater environment, the need for global, distributed localization for sensor data geo-tagging is relaxed via off-line, approximate localization at a monitoring center that uses local distance measurements or distance estimates from sonobuoys (collected along with sensor data) [22, 18].

Our goal in this dissertation is to design an efficient anycast routing protocol for underwater data collection that addresses several challenges unique to underwater communications. Most notably, the underwater acoustic channel is severely constrained by long propagation latency and low bandwidth (usually less than 100Kbps) [89], and is prone to packet losses and collisions in a congested network. Energy efficiency is a critical factor as well, given that acoustic transmissions consume far more energy than terrestrial radio communications (reception to transmission power

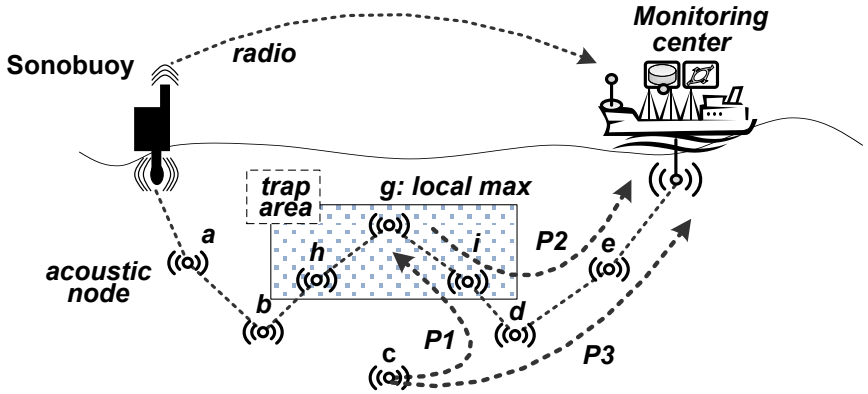


Figure 5.1: Conventional pressure routing in SEA Swarm

ratio of 1:125 [32]).

One may modify existing terrestrial routing protocols in mobile underwater networks (e.g., OLSR [21], DSDV [73], AODV [74], DSR [50]) to support anycast routing by assigning a single virtual node ID to all sonobuoys [69]. However, the major shortcomings of this approach are two-fold at least: (1) these protocols require frequent systematic flooding and route maintenance with neighboring nodes, which are very expensive operations under water, and (2) it is challenging to incorporate opportunistic forwarding mechanisms (e.g., ExOR [10], LCOR [24]) into the stateful routing protocols due to node mobility [59]—under unreliable acoustic channels, opportunistic forwarding can combat packet losses by taking advantage of simultaneous packet reception among one node's neighbors.

Therefore, recent research in underwater networks has been directed to position-based or geographic routing because it does not require any link state exchanges or route maintenances (i.e., *stateless* and *localized*). Another key advantage of geographic routing is that it enables localized geo-opportunistic forwarding; i.e., a subset of nodes that have correctly received the packet can collaboratively schedule a packet transmission to *maximize* its progress toward the destination [107, 59]. Note that our geographic routing problem is specialized in that it is anycast to any buoy on the surface. Thus, it suffices to route a packet upwards to shallower depths. Given that the onboard hydraulic pressure gauge can accurately estimate depth (avg. error $< 1m$ [49]), we can

use depth information for geographic anycast routing (called *pressure routing*) [99, 59].

Despite its benefits, simple greedy pressure routing often fails in *sparse* underwater networks due to the presence of 3D *voids*—packets must be routed around such routing holes. As depicted in Fig. 5.1, a data packet originating from node c may eventually be routed to a local maximum node g when greedily forwarded based on depth (via path P1). Node g cannot make any progress toward the surface because it does not have any neighboring node with depth shallower than its own. Node g must thus perform a route recovery process to get around the void via path P2. For 3D networks, however, it has been proven that there is no efficient memoryless routing algorithm that delivers messages deterministically in 2D face routing [26], which is also true for pressure routing. Researchers therefore have proposed several heuristic recovery methods such as random walks [31] and 2D void surface flooding [59].

There are at least two major drawbacks of such heuristic recovery methods: (1) the fallback mechanism must discover and maintain recovery paths, which are expensive in mobile networks, even more so in an underwater environment; and (2) some of the nodes around a void area will eventually route packets to local maxima (called *trapped nodes*, e.g., node i and h in Fig. 5.1), and any nodes located beneath the trapped nodes can potentially suffer from route hop stretch because local greedy forwarding may lead packets to local maxima and then invoke fallback mechanisms. In Fig. 5.1, node c's packet is greedily forwarded to the local maxima via path $P1$ and then is re-routed via path $P2$ (total 7 hops), whereas this packet can be directly delivered via path $P3$ (total 3 hops). Note that these problems will be more pronounced when the number of sonobuoys is sparse or when node density is low (both cases incur more voids in the network).

This serious shortcoming of pressure routing is inherently due to the nodes' blindness to the network topology, as they make localized routing decisions. In terrestrial stateful routing (e.g., DSDV, OLSR), each node has a *global view* of the network topology, with which packets can always be efficiently routed via shortest paths, at the cost of energy-hungry route discovery and maintenance mechanisms. This observation suggests that there is a trade-off between routing efficiency and route maintenance cost. In this dissertation, we improve pressure routing by providing

nodes with a partial view of the network topology such that greedy pressure routing is guided by *soft-state breadcrumbs* (i.e., *up/down* directions) from the sonobuoys; this method completely obviates the need of handling voids with heuristic methods.

This soft-state breadcrumb approach, which exploits periodic beaconing to build directional trails toward the surface, is much more efficient and robust than conventional underwater pressure routing protocols in that nodes maintain an enhanced view of network topology without incurring the extra cost of energy-hungry route discovery and maintenance mechanisms; nodes utilize *geo-opportunistic forwarding* along the directional trails. In this dissertation, we make the following contributions:

- We propose the Void Aware Pressure Routing (VAPR) protocol that uses surface reachability information to set up each node's next-hop direction toward the surface through which local *opportunistic directional forwarding* can always be used for data packet delivery even in the presence of voids. VAPR takes advantage of geo-opportunistic forwarding and is very robust to network dynamics such as node mobility and failure. VAPR neither requires additional recovery path maintenance nor incurs any hop stretch caused by the recovery fallbacks in existing solutions [31, 59].

- We provide a new framework of attaining loop freedom using our soft-state breadcrumb approach in mobile networks. Also, we perform extensive simulations and verify that VAPR's enhanced beacon based directional forwarding outperforms existing pressure routing protocols (e.g., DBR [99] and HydroCast [59]) and a simple hop-based greedy routing protocol under the scenarios considered.

This dissertation significantly enhances our preliminary work [67] in that we include (1) a thorough review of underwater pressure routing protocols and route recovery techniques (Section 5.2), (2) an enhanced protocol design and elaborate description of the proposed protocol (Section 5.4), (3) a detailed discussion on the loop-free property (Section 5.4.3), and (4) extensive simulation results of the proposed protocol, incorporating the Meandering Current Mobility model under

various system parameter configurations (Section 5.5.2). The rest of this Chapter is organized as follows. In Section 5.2, we review the related work in the field. In Section 5.3, we provide a brief overview of VAPR. In Section 5.4, we provide a detailed description of VAPR. In Section 5.5, we validate the performance of VAPR by comparing it with existing approaches.

5.2 Related Work

Underwater Routing Protocols: Pompili *et al.* [77] proposed two routing protocols for delay-sensitive and delay-insensitive applications in a 3D underwater environment. The delay-sensitive routing protocol is based on virtual circuit routing. Primary and backup multi-hop node-disjoint data paths are calculated by a centralized controller to achieve an optimal delay. The delay-insensitive routing protocol is a distributed geographic solution aimed at minimizing the energy consumption via back-to-back packet transmissions and cumulative acknowledgments. Vector-Based Forwarding (VBF) [97, 65] prescribes that packets be forwarded to the nodes that are located within a route of the given width between the source and the destination. This relay selection algorithm saves energy consumption by reducing the number of packet relays. Note that there are also geographic routing protocols that exploit the opportunistic forwarding features in underwater environments [52, 99, 59], which will be detailed later. Vieira *et al.* [94] proposed Phero-Trail routing that efficiently delivers packets to a mobile sink by following a pheromone trail of the sink. Besides unicast routing and converge-cast routing, broadcasting is also required by some underwater sensor applications (e.g., reprogramming sensor nodes). Casari *et al.* [15] proposed several reliable broadcasting protocols that leverage the ability to use small bands to transmit an alert packet for a long distance. After sending alert signals, nodes reduce the transmission range and select only certain neighboring nodes in order to repeat broadcast, thereby lowering the total number of transmissions required. Similar ideas can be found in other related work [14, 66]. Readers can find more detailed survey of recent underwater routing protocols in the survey papers [16, 6].

Opportunistic Routing: Most opportunistic routing protocols (also called anypath routing) such

as ExOR [10], Least Cost Opportunistic Routing (LCOR) [24], which do not use geographic information, require global topology and link quality information (like link state routing) to find a set of forwarding groups toward the destination; thus, they are more suitable for *static* wireless mesh or sensor networks. In practice, geographic routing can also benefit from opportunistic forwarding, as in Geographic Random Forwarding (GeRaF) [107], Contention Based Forwarding (CBF) [37], and Focused Beam Routing (FBR) [52], though these methods are not optimal due to lack of global knowledge. In the literature, researchers have typically used a geometric shape (e.g., a triangle/cone [37, 52]) that is *faced toward the destination* for forwarding set selection to prevent hidden terminal problems.

Pressure Routing: Yan *et al.* proposed a greedy anycast routing solution called Depth Based Routing (DBR) [99]. They suggested that packet forwarding decisions be made locally based on the pressure (or depth) level measured at each node. Packets would then be geographically forwarded to nodes with shallower depth in a greedy fashion toward the water surface. This hydraulic pressure based anycast routing protocol benefits from being stateless and does not require expensive distributed localization [84, 9]. DBR exploits the simultaneous packet reception induced by the broadcast nature of the wireless medium and performs *opportunistic greedy forwarding* via a subset of the neighbors that have received the packet correctly. However, DBR lacks an efficient forwarding set selection method and a recovery method from local maxima. Lee *et al.* proposed HydroCast [59] to remedy these problems. HydroCast improves the efficiency of the forwarding set selection method by choosing a set that maximizes greedy progress yet limits co-channel interference. Additionally, HydroCast has a route recovery scheme that uses a hop limited ring search over the 2D surface of a convex hull around the void area to discover a recovery path. Like these protocols, our protocol relies on opportunistic greedy forwarding with a forwarding set selection algorithm borrowed from HydroCast.

Recovery Mechanisms: The techniques for routing around the local maxima can be classified into two categories: *stateless* (memoryless) and *stateful*. Recently, Durocher *et al.* [26] proved that stateless recovery in 3D networks is infeasible unless it is as naïve as random walks [31].

There was an attempt to project a 3D network onto a 2D plane [68], but it was shown that face routing on the projected 2D plane cannot guarantee packet delivery. Liu *et al.* proposed a partial unit Delaunay triangulation (PUDT) algorithm to construct hulls that partition the 3D network into subspaces so that recovery can be simply done by exploring the subspace. Zeng *et al.* [101] proposed to embed the 3D network into a hyperbolic space using a discrete hyperbolic Ricci flow. Nodes are mapped to virtual coordinates in the hyperbolic space, which intrinsically have paths to avoid holes—greedy forwarding is always possible in hyperbolic space. However, in mobile underwater sensor networks, this mapping must be periodically refreshed (due to mobility), thus leading to de facto flooding. Thus the cost is comparable to that of flooding heuristics.

Several *stateful* approaches have been suggested mainly for 2D networks, but they are generally extensible also to 3D networks [43, 60, 61, 103, 101]. He *et al.* proposed SPEED, which reactively uses backpressure-based backtracking to inform upstream nodes to prune paths that reach a local maximum [43]. In [60, 104], a spanning tree was used in which each node has an associated convex hull that contains within it the locations of all its descendant nodes in the tree. Liu et al. [61] proposed using a virtual coordinate system to route packets, in which a packet can be backtracked toward one of the anchor nodes in the event that recovery is needed. Geo-LANMAR [103] inherits the group motion support of Landmark Routing (LANMAR) to identify landmark nodes (cluster-heads) and maintains routes to such landmarks using a combination of georouting and directional forwarding. A periodic beacon propagates, from which the geo-distance, hop distance and cluster membership are derived by each landmark, thus functioning as a kind of a distributed DNS. Each node extracts from the beacon the direction to the landmark along the beacon traced shortest path. When georouting gets stuck, the beacon-guided direction is used (as in directional forwarding) to recover from voids. The direction to a landmark is used because next-hops change too rapidly in a mobile environment, whereas the direction changes occur less frequently.

Key Differences: VAPR is also stateful and resembles Geo-LANMAR in that sonobuoys propagate surface reachability information (via enhanced beaconing) for each underwater node to setup its next-hop direction toward the surface. The key difference between VAPR and previous schemes

are three-fold: (1) VAPR always uses *local greedy directional forwarding* for data delivery on the basis of the direction cues (it does not wait until it gets stuck in a local maximum like Geo-LANMAR does); (2) VAPR neither requires recovery path maintenance nor incurs any hop stretch caused by the recovery fallbacks when compared to existing solutions [31, 59] and; (3) VAPR requires a small soft-state, i.e., next-hop direction and hop distance information at each node (as readily available from the Beacon) and is robust to network dynamics such as node mobility, failures and possible sleep/wake up cycles.

5.3 VAPR Overview

VAPR is composed of two major components, namely *enhanced beaconing* and *opportunistic directional data forwarding*. In the former, sonobuoys propagate their reachability information to sensor nodes via enhanced periodic beaconing.[1] In enhanced beaconing, each node's beacon is augmented with additional information, namely the sender's depth, hop count to a sonobuoy, sequence number, and its current *data forwarding direction* (toward the surface). When receiving the augmented beacons from predecessors, each node updates its variables, namely *minimal hop to the surface*, *sequence number*, *data forwarding direction*, and *next-hop data forwarding direction* (i.e., predecessor's data forwarding direction).

In the beginning, every sonobuoy on the surface initializes these variables and starts beaconing. After receiving a beacon message, a node can tell whether it has received the message from deeper or shallower depth, and each node sets its data forwarding direction toward the surface. The direction is set as *up* when a beacon is received from a shallower depth node; otherwise, it is set as *down*. When multiple direction cues from different sonobuoys are received, the direction cue with the minimal hop count is chosen. Also, a node's next-hop data forwarding direction is set based on the beacon sender's data forwarding direction. For instance, in Fig. 5.1, since node d/e receives a beacon message from a shallower depth node, it sets its data forwarding direction

[1] Recall that periodic beaconing is an essential part of the architecture for underwater localization. It comes at zero incremental cost, and it is up to the specific routing scheme to exploit it or not.

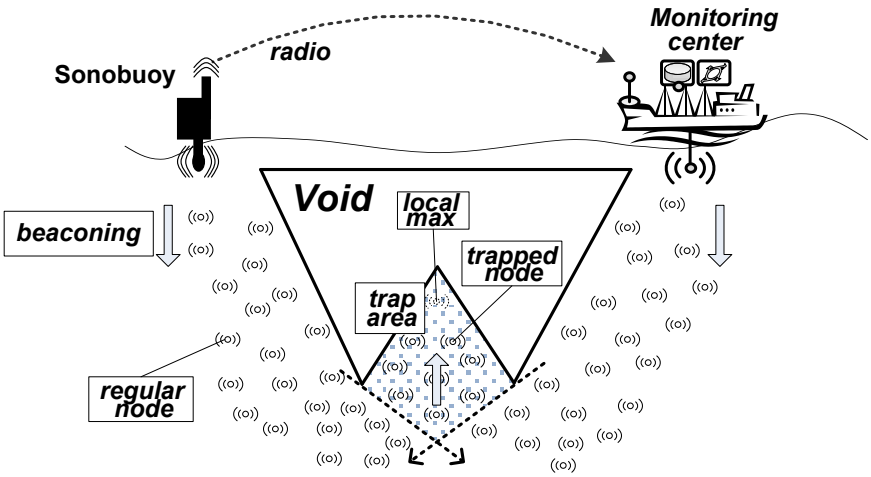


Figure 5.2: Terminology

and next-hop data forwarding direction as *up-up*, respectively. In contrast, when node i receives a beacon message from a deeper depth node (i.e., node d), it sets its data forwarding direction and next-hop data forwarding direction as *down-up*, respectively. After updating its local states, each node prepares a beacon message to broadcast by incrementing the hop count and setting its current depth, data forwarding direction, and sequence number. This beaconing process will repeat, and thus, nodes essentially build directional trails toward their closest sonobuoys on the surface. Note that a direction change is caused by *voids*; e.g., in Fig. 5.1, path $P3$ has only a single direction (*up*), whereas path $P2$ experiences a direction change due to the void (*down* and *up*).

Given this, VAPR performs local *opportunistic directional forwarding* to deliver data according to the directional trails. The forwarding decision is solely made based on the local state variables, namely the data forwarding direction and next-hop data forwarding direction—hop-count information is never used for routing to exploit opportunistic packet receptions. Two cases are possible: (1) if there is no void, packets can always be greedily routed via the upward direction, and we can solely rely on the data forwarding directions for routing; and (2) if there are voids, there will be direction changes, and the next-hop forwarding direction is jointly used with the data forwarding direction to guide the routing direction. Consider the $\bigvee$-shape topology in Fig. 5.1 and assume

that the states of nodes g, i, d, e are *down-down*, *down-up*, *up-down*, and *up-up*, respectively. A data packet that originated from node g can be greedily routed downwards to node i. This node realizes that there is a direction change in the next hop (as its status is *down-up*), and for packet forwarding it only considers the neighboring nodes whose depth levels are deeper and whose data forwarding directions are upward (i.e., node d), ensuring that a change of the routing direction is correctly made.

The following terminology is used throughout the dissertation (see Fig. 5.2). A *local maximum* node is a node whose depth level is shallower than that of all its neighboring nodes but deeper than that of the sonobuoys; local greedy upward forwarding cannot make any progress toward the surface. A *trapped node* is a node in which greedy forwarding eventually leads to a local maximum node. A local maximum node is also by definition a trapped node. As shown in the figure, trapped nodes are usually found beneath the concave area of voids. The area in which trapped nodes reside is called the *trap area*. The rest of the nodes (that are not trapped nodes) are called *regular nodes*.

5.4 VAPR details

Table 5.1: Terminology

Terms	Definitions
V	Set of nodes
S	Set of sonobuoys, where $S \subset V$
$\pi(node)$	Water pressure measured at $node$
DF_dir	$\in \{null, down, up\}$
NDF_dir	$\in \{null, down, up\}$
hop_count	$\in [0, \infty]$
seq_num	Sequence number used for periodic beaconing

Algorithm 1 Routing initialization

1: **procedure** Initialize(node)
2: **if** $node \in S$ **then**
3: $DF\_dir(node) \leftarrow up$
4: $hop\_count(node) \leftarrow 0$
5: $seq\_num(node) \leftarrow 0$
6: **else**
7: $\pi(node) \leftarrow$ pressure measured at $node$
8: $DF\_dir(node) \leftarrow null$
9: $NDF\_dir(node) \leftarrow null$
10: $hop\_count(node) \leftarrow \infty$
11: $seq\_num(node) \leftarrow null$
12: **end if**
13: **end procedure**

5.4.1 Enhanced beaconing

In VAPR, each sonobuoy propagates surface reachability information to underwater nodes to give nodes an enhanced view of the network. We modify periodic beaconing of pressure routing (originally designed to exchange hello messages amongst neighbors) by embedding the sender's depth, hop count, data forwarding direction, and sequence number in a beacon message. Given this information, each node i keeps its local status of $node_{ID}(i)$, $\pi(i)$, $DF\_dir(i)$, $NDF\_dir(i)$, and a tuple of $hop\_count(i)$ and seq_num, where $node_{ID}(i)$ is node i's ID, $\pi(i)$ is i's depth from the sea surface (or pressure level), $DF\_dir(i)$ is i's data forwarding direction toward a sonobuoy, $NDF\_dir(i)$ is the next-hop data forwarding direction (i.e., data forwarding direction of i's predecessor), $hop\_count(i)$ is the number of hops from i to the sonobuoy, and seq_num is the sequence number used for periodic beaconing. Here, we assume that sonobuoys on the surface are equipped with GPS, through which their clocks are synchronized, and that they use the same sequence number for periodic beaconing. The sequence number will be incremented periodically; e.g., with a

fixed beacon interval of 30 seconds. As we will see, the sequence number and hop count allow nodes to handle potential direction loops or oscillations caused by nodal mobility and randomness of periodic beaconing, which will be further discussed in Section 5.4.3. Also, each node maintains minimal information about its one-hop neighbors; i.e., for each neighbor n, we keep $node_{ID}(n)$, $\pi(n)$, $DF\_dir(n)$. Every neighbor entry will be refreshed whenever a beacon message is received from that node. If the timer of an entry expires, the expired entry can be deleted from the current node's storage, thereby removing the node from its neighboring node set.

Algorithm 1 is used to initialize each nodes' internal states (see Table 5.1 for the terminology used in the pseudo codes in this dissertation). Initially, each node in the network begins as an isolated local maximum node (i.e., indicated by $hop\_count(node)$ equaling ∞, which we explain later), with the exception of sonobuoys on the sea surface (as they are the destinations). Naturally, each node in the connected component with at least one sonobuoy will have its status changed to a non-local maximum node.

Algorithm 2 is used to broadcast periodic beacons and handle received beacons. In a beacon message, nodes embed their local states, namely current hop-count, sequence number, depth, and data forwarding direction. To minimize the chance of collisions and synchronization, nodes add random jitters for periodic beaconing using timers when they broadcast beacon messages; then, the nodes set up a new timeout for the next beaconing. After receiving a beacon message, each node checks the validation of the received beacon by checking the sequence number (increasing) and hop-count (smaller), and each node sets its data forwarding direction and updates its next-hop data forwarding direction. As illustrated earlier, the direction is set as *up* if a beacon message is received from a shallower depth node; otherwise, it is set as *down*. After the data forwarding direction is set, a node's next-hop data forwarding direction is also updated based on the data forwarding direction of the beacon sender. Note that when multiple direction cues from different sonobuoys are received, the direction cue with minimal hop count is deterministically chosen. Algorithm 2 summarizes this beaconing and node state update process.

Fig. 5.3 shows an example to illustrate Algorithm 1 and 2. The sonobuoy initializes a beacon

Algorithm 2 Enhanced beaconing

1: **procedure** BroadcastPeriodicBeacon($node$)
   m: a new beacon message
2: **if** beacon timeout expired **then**
3: $m.\pi \leftarrow \pi(node)$
4: $m.DF\_dir \leftarrow DF\_dir(node)$
5: $m.hop\_count \leftarrow hop\_count(node)$
6: Broadcast m
7: Set a new timeout
8: **end if**
9: **end procedure**
10:
11: **procedure** ReceiveBeacon($node$, m)
12: **if** $m.seq\_num > seq\_num$ **or**
   $m.seq\_num = seq\_num$ &
   $m.hop\_count + 1 < hop\_count(node)$ **then**
13: $NDF\_dir(node) \leftarrow m.DF\_dir$
14: $hop\_count(node) \leftarrow m.hop\_count + 1$
15: **if** $\pi(node) > m.\pi$ **then**
16: $DF\_dir(node) \leftarrow up$
17: **else**
18: $DF\_dir(node) \leftarrow down$
19: **end if**
20: **end if**
21: **end procedure**

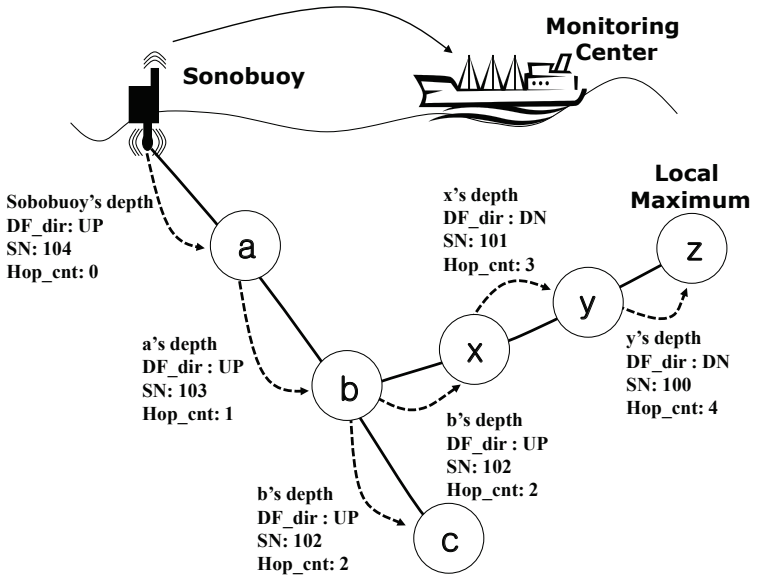


Figure 5.3: Enhanced beacon propagation

message after the beacon timer has expired and then broadcasts the beacon message with the sequence number ($= 0$), depth ($= 0$), data forwarding direction (=*up*), and hop count ($= 0$). Node a receives the beacon and finds that it is a new beacon with a higher sequence number and sets its status (e.g., $seq\_num = 0$ with incremented $hop\_count(a) = 1$). By comparing the depth, node a sets $DF\_dir(a)$ as *up*, and $NDF\_dir(a)$ (i.e., the sonobuoy's data forwarding direction) as *up*. Node a will broadcast an updated beacon, and node b will perform a similar procedure, which will be continued. Later, node x receives a beacon message from node b; it then updates $DF\_dir(x)$ as *down* based on the depth difference and $NDF\_dir(x)$ as $DF\_dir(b)$ = *up*. Node x will broadcast an updated beacon message. After these changes are announced via a beacon message, node y receives the beacon message and will maintain $DF\_dir(y)$ and $NDF\_dir(y)$ as *down-down*. On the basis of this beacon propagation and update process, nodes will set up a set of directional trails toward any one of the sonobuoys.

When multiple direction cues from different sonobuoys are received, direction flapping may occur. VAPR uses the sequence number and hop count to prevent such flapping. Whenever a node receives a beacon message with a higher sequence number than its current sequence number, the

node simply updates its status based on the received beacon. However, if a node receives a beacon message with the same sequence number, we compare the hop counts, and the data forwarding direction is set toward the node that has a smaller hop count. If a tie occurs, there are two possible cases: a node's current data forwarding direction (i.e., DF_dir) is identical or different (beacons from both shallower and deeper depth levels). The former case of the identical direction can be safely ignored as there will be no impact on the direction setting. For the latter case, we must deterministically use a preferred direction to prevent route flapping; in our scenario, the upward direction is used by default. In Algorithm 2, we omit the details about tie-break in the procedure of RECEIVEBEACON for the sake of brevity. In Fig. 5.4, for instance, node i receives beacon messages from both directions (from h and j). Node i chooses the forwarding direction toward the closer sonobuoy (*down* in this case) by comparing the hop counts. If both hop counts are the same (a tie), we deterministically set the data forwarding direction as *up*. Note that hop counts are only used for setting up the trails and are not considered when routing data at all—during data forwarding, nodes forward data based solely on the data forwarding and next-hop data forwarding directions, fully exploiting opportunistic directional forwarding, which will be explained in the following section.

5.4.2 Opportunistic directional data forwarding

Directional data forwarding: In VAPR, nodes forward data packets solely based on the *data forwarding direction* (DF_dir) and *next-hop data forwarding direction* (NDF_dir). Recall that each node sets up the data forwarding direction (either *up* or *down*) that is the opposite direction of the beacon reception direction. If this direction is *up*, we use greedy upward forwarding; otherwise, we use greedy downward forwarding. For instance, in greedy upward/downward forwarding, a node basically forwards a packet to the node whose depth is the shallowest/deepest among its neighbors, respectively.

As the data packet is forwarded upward beneath the concave area of *voids*, a change of data forwarding direction is inevitable. Data forwarding direction alone cannot provide sufficient in-

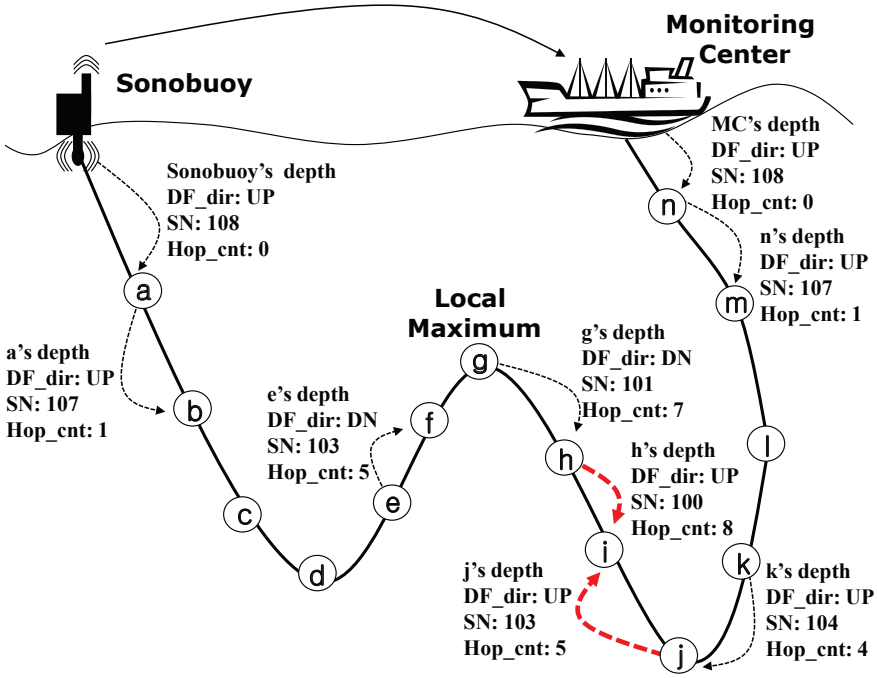


Figure 5.4: Beacon receptions in both directions (node i)

formation to route packets to the destination. In conjunction, we use the next-hop data forwarding direction, which was the predecessor's data forwarding direction during beacon propagation and now becomes the next-hop's data forwarding direction. The forwarding node uses the next-hop's data forwarding direction as an additional *direction constraint* to ensure that routing properly follows the direction trails; i.e., among all neighboring nodes, we only consider the neighboring nodes whose *data forwarding directions* are equal to the *next-hop data forwarding direction* of the current node. As illustrated earlier, there are only four possible cases of data forwarding and next-hop data forwarding direction setting: *up-up*, *down-down*, *down-up*, and *up-down*. The direction changes happen in the latter two cases: from *up* to *down* in the case of *up-down* ($\bigwedge$-shape topology), and from *down* to *up* in the case of *down-up* ($\bigvee$-shape topology).

Consider an example scenario depicted in Fig. 5.5. In particular, let us take a look at the $\bigvee$-shape topology formed by nodes a, b, and x. Here, node x is a trapped node that eventually delivers packets to the local maximum via greedy upward forwarding. DF_dir and NDF_dir of nodes a

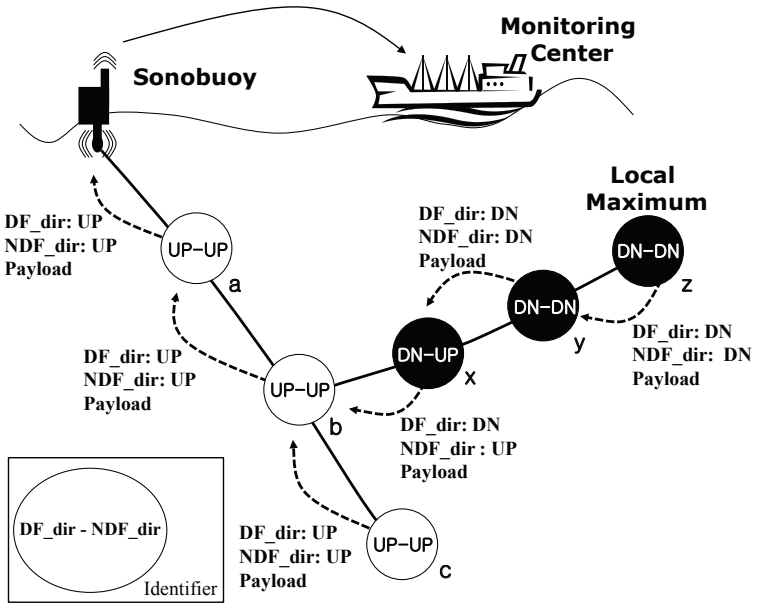


Figure 5.5: Directional data forwarding

and b are *up-up*, whereas those of node x are *dn-up*. Let us say that there is a packet to send in node b. Node b's DF_dir is *up* and will consider nodes whose depth is shallower than that of node b, namely nodes a and x. Since node x's DF_dir (*down*) does not match with that of NDF_dir (*up*), the trapped node x is filtered out, and node a is only considered as a forwarding candidate for local greedy "upward" forwarding.

Enhancement with geo-opportunistic forwarding: So far, a packet is greedily forwarded to the node closest to the destination, in hopes of minimizing the average hop count to the surface. Due to channel fading, however, the farther the transmission range, the higher the attenuation, and the greater the likelihood of packet loss. In VAPR, we consider simultaneous packet receptions by one's neighbors and their ability to opportunistically forward packets by scheduling the set of nodes that have received the packet correctly, which is widely used in geographic routing to improve routing performance under channel fading [107, 37, 52].[2] The key design issue of geo-

[2]Note that conventional opportunistic routing protocols (also called anypath routing) such as ExOR [10], Least Cost Opportunistic Routing (LCOR) [24] do not use geographic information, but require global topology and link quality information (like link state routing) to find a set of forwarding groups toward the destination; thus, they are more suitable for static wireless mesh or sensor networks.

opportunistic forwarding is the selection of a subset of neighbors that can make the best progress for a given direction, yet without the hidden terminal problem: i.e., when a higher priority node (based on the distance) transmits a packet, other low priority nodes should be able to suppress forwarding to prevent redundant packet transmissions and collisions. Given that finding an optimal set is computationally hard, several heuristics were proposed in the past: a geometric shape (e.g., a triangle/cone [37, 52], a depth based threshold (e.g., DBR [99]), or greedy clustering (e.g., HydroCast [59]).

In VAPR, we use a simple greedy clustering approach that is superior to the geometric shape or depth-based approaches [59]. To this end, each node requires the knowledge of 2-hop connectivity and neighboring nodes' pairwise distances. Recall that for off-line localization we assume that each node measures the pair-wise distance [18], and the data are periodically reported to the surface. VAPR takes advantage of such periodic reports to obtain 2-hop neighbor information. We start with a node whose priority is highest (i.e., furthest distance) along the data forwarding direction and choose a group of nodes among its neighbors within the distance $< \beta R$. Here, β is a constant ($\beta < 1$, $\beta = 1/2$ in our design) and R is the acoustic communication range. Then, if other neighbors are left, clustering proceeds to the second highest priority from remaining neighbors and so on, until no nodes are left. After this, each cluster is expanded by including all the additional nodes such that the distance between any two nodes in the cluster is smaller than R. This condition guarantees that nodes in the set can hear each other (i.e., no hidden terminals). We repeat this for all other clusters in turn and find the cluster with the highest expected packet advancement toward the selected direction.

After forwarding the set selection, we need to include the chosen forwarding set in the data packet. To reduce the overhead, we use a Bloom filter, a space efficient membership checking data structure. The membership checking is probabilistic and false positives are possible, but we can bound the probability of false positives by properly adjusting the filter size. In a practical scenario, the set size will be smaller than 15 (in the hemisphere advance zone). Fan et al. [30] showed that a filter size of 150 bits (19B) to represent 15 items has a false positive rate smaller

Algorithm 3 Opportunistic Directional Data Forwarding Set Selection

1: **procedure** Directional_FSS($node$, $data$)
2: F = ϕ // start with empty set
3: // check all neighbors
4: **for** $n \in$ neighbors(node) **do**
5: // FSS for greedy downward forwarding
6: **if** $DF\_dir(node) = down$
and $\pi(node) \geq n.\pi$
and $n.DF\_dir = data.NDF\_dir$ **then**
7: $F \leftarrow F \cup n$
8: **end if**
9: // FSS for greedy upward forwarding
10: **if** $NDF\_dir(node) = up$
and $\pi(node) \leq n.\pi$
and $n.DF\_dir = data.NDF\_dir$ **then**
11: $F \leftarrow F \cup n$
12: **end if**
13: **end for**
14: // Perform greedy clustering to find the best cluster
15: C = Clustering_FSS(F, node, data)
16: Return C
17: **end procedure**

than 1%. We can also include sender's depth, and max/min angle information to filter out quite a few of neighboring nodes that are not in the forwarding set. Furthermore, noting that there could be many other packets that have to travel through a certain node, and topology slowly changes over time, we may only need to include the set information in the data packet whenever there is a sufficient change. Thus, the amortized overhead could be much smaller. Algorithm 3 provides a simplified opportunistic directional data forwarding algorithm. The algorithm invokes a function called, $Clustering\_FSS(F, node, data)$ to select possible forwarding nodes based on DF_dir and NDF_dir among its one-hop neighbors and performs clustering to find the best cluster and to return a subset of one-hop nodes that can make the best progress without the hidden terminal problem.

5.4.3 Discussion on the loop free property

For the completeness of the routing algorithm, loop freedom in static and mobile networks must be provided. Most ad hoc routing protocols guarantee loop freedom based on the following observation. Periodic routing request flooding basically builds a reverse path tree toward the source. Route replies will follow the reverse path along which the hop count monotonically decreases. In fact, this simple property ensures a *strict* ordering of feasible distances along successor paths, and thus, loop freedom is guaranteed. For instance, the RREQ tree is formed via the conventional reverse-path flooding techniques of the Ad hoc On Demand Distance Vector (AODV) routing protocol; similarly, the Destination-Sequenced Distance Vector (DSDV) routing protocol periodically performs network-wide flooding with new sequence numbers to update the routing tables. This strict ordering of feasible distances for a given destination is attained by ensuring the Numbered Distance Condition (NDC), as follows [38, 39].

Definition 1. *(Numbered Distance Condition) Node A may accept a route advertisement from neighbor B from destination D and update its routing table independently of other nodes if A has no information about destination D or if either one of the following two conditions is satisfied:* $seq\_num_D^A < seq\_num_D^B$ *or* $seq\_num_D^A = seq\_num_D^B$ *and* $hop\_count_D^A > hop\_count_D^B$. *Here,*

$seq\_num_D^A$ denotes the sequence number to destination D sent from node A and $hop\_count_D^A$ denotes the hop count to destination D sent from node A.

Loop-free property of VAPR: While existing routing protocols ensure the NDC property using network wide "instant" flooding, we want to show that the enhanced periodic beaconing in VAPR ensures the NDC property and guarantees the loop-free property. If a network is static, the formal proof is quite straightforward.

We can basically use proof-by-contradiction. For the sake of simplicity, the hop count is used to show the progress to the surface, by assuming that an instance of greedy forwarding has the same effect of decrementing a hop count by one. The proof can also be easily extended to consider the physical distance. During route trail construction, a beacon that carries path information from one of the sonobuoys reaches each node in a connected network. The hop distance monotonically increases along the path. The route from the node to the sonobuoy (return route) follows, by construction, a path with monotonically decreasing hop count. The return route must have the same number of hops as the incoming route. If the return route were shorter, the beacon on the shorter route would have labeled the node. If it were longer, it would have been suppressed by the shorter route, and thus, this cannot happen. By the route trail construction, the return route cannot lead to a dead end. Thus, it must end up either at the sonobuoy that labeled the node, or at another sonobuoy at equal distance.

In mobile networks, existing protocols ensure loop-freedom using either on-demand (e.g., AODV) or periodic (e.g., DSDV) network-wide flooding. A given sequence number will be instantly available throughout the network, and a strict ordering of feasible distance happens—the speed of message propagation is an order of magnitude faster than nodal mobility. However, this means that after some time, the strict ordering may break, and to guarantee the loop-free property in a mobile network, the network must be constantly flooded, which is an expensive process in underwater acoustic networks.

Instead of "periodic" instant flooding, VAPR embeds route discovery into the beaconing process. Then the question is how the property of periodic instant flooding can be emulated using the

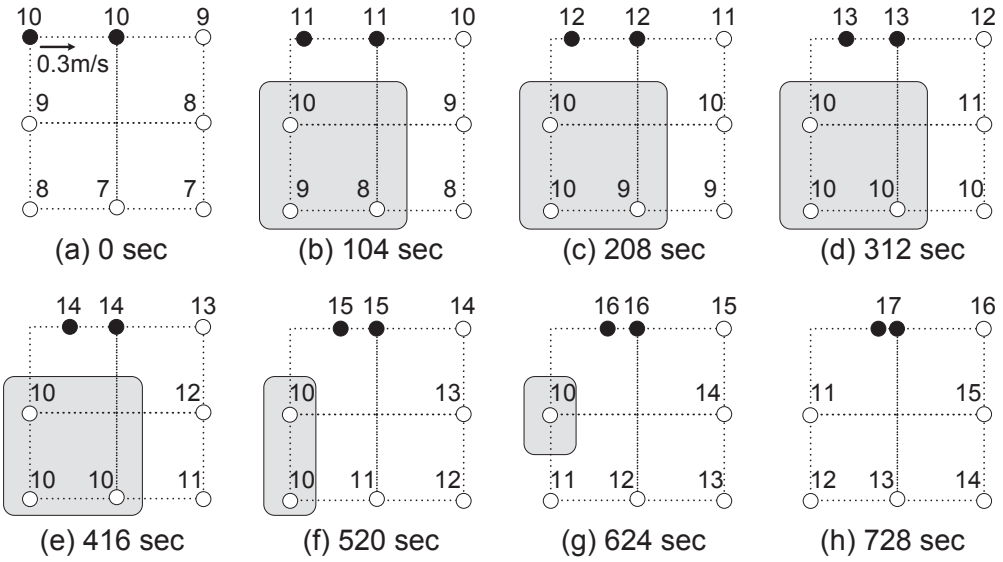


Figure 5.6: Beacon propagation with nodal mobility of $0.3m/s$ and beacon interval of 104s

enhanced beaconing process. We note that the flooding interval in traditional routing (e.g., DSDV) is mainly determined by the transmission range and node mobility. If the transmission range is around 250m, and the relative node speed is 10m/s, we may want to set the interval smaller than the average time of traveling the transmission range (i.e., 25s). For instance, in a highly mobile scenario, DSDV is typically configured to run full table updates every 15 seconds [11]. To illustrate how the periodic beaconing should be configured in VAPR, let us consider the following scenario. Assuming that the maximum distance is K (from the sonobuoys), to ensure the property of "instant" flooding, a given sequence number needs to be propagated within the link lifetime that are mainly dependent on the transmission range and node mobility [83]. The approximate relationship can be represented using the following inequality: K*Beacon Interval (BI) $\leq \alpha$ Transmission Range (TR) / Nodal Speed (NS) where α is constant. Thus, we have $K * NS \leq \alpha TR/BI$. To summarize, the sequence number propagation speed (=TR/BI) and the maximum depth K are the key factors in determining the loop-free property in VAPR.

For instance, consider a scenario with a nodal speed of $0.3m/s$. Assuming that we have $K = 8$, the sequence number should propagate at the speed of $8 \times 0.3m/s$ ($2.4m/s$). Assuming that the

transmission range is $250m$, the beacon interval should be smaller than $250m/(8 \times 0.3m/s) = 104s$. In Fig. 5.6, we present a 2×2 grid topology in which there are two sonobuoys (black dots) and five regular nodes (white dots), and the side length of the grid is $250m$. In this example, we assume that one sonobuoy moves toward the other sonobuoy at a speed of $0.3m/s$, and that the rest of the nodes are stationary. In the initial state shown in Fig. 5.6(a), the up-to-date sequence number is 10, and both sonobuoys have a synchronized sequence number. In Fig. 5.6(b), the gray area becomes disconnected right after the departure; nodes in that area suffer from transient disconnection. At the same time, the sequence numbered 10 is propagated to the gray area (9, 8, 7 to 10, 9, 8). In Fig. 5.6(c), the lead node no longer receives a new sequence number because it goes out of reach from the left sonobuoy. This node keeps the sequence number 10 until it receives a newer sequence number. In the meantime, the nodes outside the gray area continue to propagate the sequence numbers in each beacon interval. In Fig. 5.6(d), nodes in the gray area all have the sequence number 10, and they are waiting for the sequence number 11. After four beacon intervals (i.e., at the time mark of 728s in Fig. 5.6(h)), the transient disconnection is resolved, and all nodes have a strict ordering to the sonobuoy on the right.

In VAPR, to reduce the overhead, we aggressively use a larger beacon interval (i.e., by using a smaller value than K, the maximum distance). In the above example (Fig. 5.6), when we set the K value smaller than 8, it takes a longer time to become loop-free than the case in which K is set to the maximum distance. For instance, if we set $K = 4$, the beacon interval is 204s (twice the original value). Since it takes 7 beacon intervals to receive a new sequence number, the transient instability lasts for 1456 seconds.

Fortunately, in practice the effect of route instability can be minimized due to the unique characteristic of underwater sensor networks and VAPR's routing strategies, namely (1) restricted/clustered mobility patterns of underwater sensors (moving along with water current), (2) the multi-path nature of opportunistic routing, and (3) beacons sent from multiple sonobuoys. Underwater sensor nodes maintain a fixed depth and move along with the water current. Their mobility patterns are naturally clustered and lead to restricted movement within the cluster. Since sensor nodes are or-

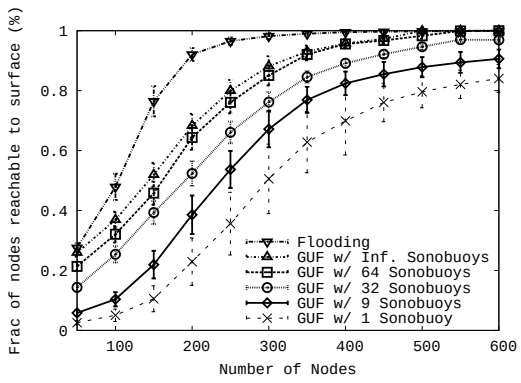


Figure 5.7: Fraction of nodes reachable to sonobuoys

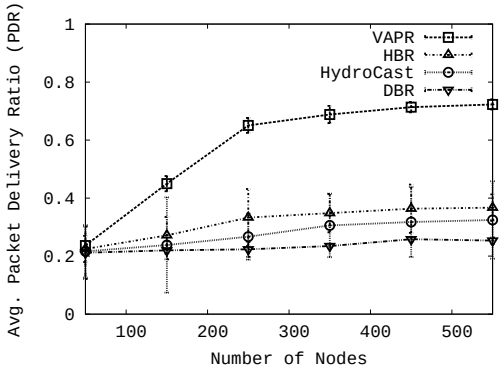


Figure 5.8: PDR (*1 sonobuoy scenario*)

dered based on their depths, it is likely the case that the distance ordering (hop count) follows the depth order—clustered mobility patterns make deviation of ordering small. In VAPR, any node can maintain distance ordering as long as at least one of the next-hop neighboring nodes receives a newer sequence number (due to opportunistic forwarding). Moreover, it is likely the case that a node will receive beacons from multiple sonobuoys, making more paths towards the surface. In Section 5.5.2, we further investigate the effect of different beacon intervals on Packet Delivery Ratio (PDR) and energy consumption.

5.5 Simulations

5.5.1 Simulation setup

For acoustic communications, the channel model described in [88] and [62] is implemented in the physical layer of QualNet. The path loss over a distance d for a signal of frequency f, due to large scale fading is given as $A(d, f) = d^k a(f)^d$ where k is the spreading factor and $a(f)$ is the absorption coefficient. The geometry of propagation is described using the spreading factor $(1 \leq k \leq 2)$; for a practical scenario, k is given as 1.5. The absorption coefficient, $a(f)$, is described by Thorp's formula [62]. As in [88, 12], we use Rayleigh fading to model small scale fading. Unless otherwise mentioned, the transmission power is set to 105 dB re μ Pa. We use a transmission range of 250m; the data rate is set at 50Kbps, as in [105]. Our simulations use the CSMA MAC protocol. In CSMA, when the channel is busy, a node waits a back-off period and attempts to sense the carrier again. Every packet transmission is performed through MAC layer broadcasting. For reliability, we implement ARQ at the routing layer as follows for both HydroCast and our proposed routing algorithms. After packet reception, the receiver sends back a short ACK packet. If the sender fails to hear an ACK packet, a data packet is retransmitted; the packet will be dropped after five retransmissions.

We randomly deploy varying numbers of nodes ranging 50 to 550 in a 3D region of size 1500m $\times$ 1500m $\times$ 1500m. To test routing protocols in a more realistic SEA Swarm scenario, we adopt an extended 3D version of the Meandering Current Mobility (MCM) Model [13] to model the motility of each sensor node. Unlike most existing sensor node mobility patterns from literature, which assume that each node moves independently of all others, wherein its path vector is determined from an independent realization of a stochastic process, the MCM model considers fluid dynamics whereby the same velocity field advects all nodes. Here, the MCM model considers the effect of meandering sub-surface currents (or jet streams) and vortices on the deployed nodes to pattern its path vector.

Meanwhile, additional nodes are deployed in a grid topology on the upper surface of the region

(from 1 to 64) to simulate the presence of sonobuoys. Each node measures the distance to its neighbors every 50 seconds (with random jitters to prevent synchronization) and broadcasts the measured information to its one hop neighbors. Every 50 seconds, each node reports the sensed data and distance measurements to the surface. The size of a packet is a function of the number of neighbors, and the average packet size in our simulations is less than 200B. We measure packet delivery ratio, average latency per packet, and energy consumption per packet as functions of the number of deployed mobile sensor nodes. The packet delivery ratio of a source is the fraction of the packets delivered; the average latency is the averaged time for every packet to reach any of the sonobuoys on the surface; and the energy consumption is measured in *mWhr* in terms of energy spent per node and per message by each node during the simulation to deliver a packet to the sink. In our simulation, each run lasts 1 *hour*. Unless otherwise specified, we report an average value of 50 runs with a 95% confidence interval.

We have evaluated our proposed routing algorithm against two recent routing protocols: DBR [99] and HydroCast [59]. Recall that DBR greedily forwards packets toward the sea surface using a linear back-off timer proportional to the distance to the destination. This ensures that the nodes closest to the broadcasting node will wait for the nodes closer to the destination that have received the packet to broadcast first. Overhearing the broadcast of the packet by a node closer to the destination serves as an acknowledgement that the packet was forwarded toward the sea surface, and suppresses node transmissions of packets by nodes that are closer to the source, providing an opportunistic forwarding flavor. However, due to lack of an optimized forwarding set selection mechanism, DBR suffers from many redundant transmissions and packet collisions. HydroCast uses a similar linear back-off timer but it calculates an optimal forwarding set based on expected packet advancement [100] and directs the packet to be routed in a general direction relying on opportunistic packet receptions. If the packet is routed to a trap area, a hop limited ring search is used to build a discovery path along the 2D surface of the convex hull around the void. We evaluate HydroCast with this recovery process. To further evaluate the performance of opportunistic direction forwarding, we additionally compare VAPR with Hop-Based Routing (HBR) that only

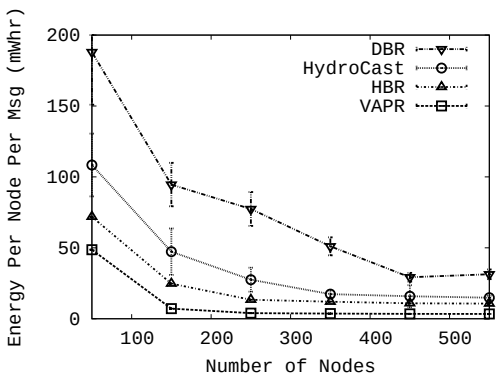


Figure 5.9: Energy consumption per message (*1 sonobuoy scenario*)

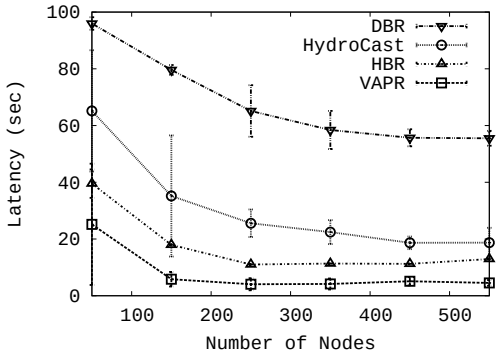


Figure 5.10: Avg. latency (*1 sonobuoy scenario*)

uses hop-based reachability information to make forwarding decisions: i.e., forwarding a packet to any randomly selected neighboring node whose hop count is smaller than that of the current node. Basically, HBR considers neither physical distance nor opportunistic forwarding; any nodes with the same hop counts are treated equally.

5.5.2 Simulation results

We first analyze the network connectivity and its impact on the performance of greedy forwarding under different node and sonobuoy densities. To this end, we perform network-wide flooding from the sonobuoys and measure the fraction of underwater nodes that can reach the surface. We also

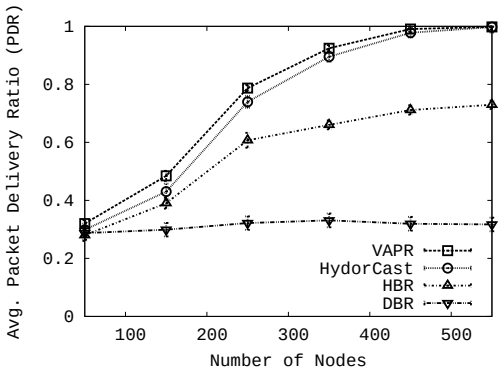


Figure 5.11: PDR (*64 sonobuoy scenario*)

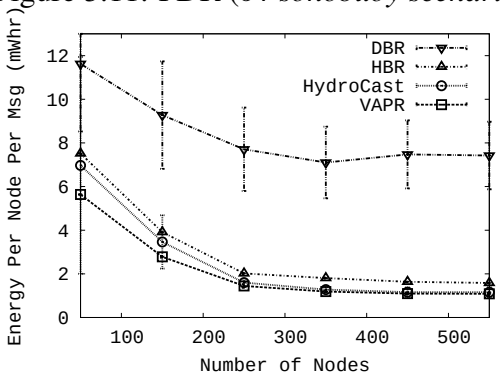


Figure 5.12: Energy consumption per message (*64 sonobuoy scenario*)

measure the number with greedy upward forwarding (GUF) by varying the number of sonobuoys. In the case of network-wide flooding, we do not vary the number of sonobuoys as it is not sensitive to the sonobuoy density. We present the overall results in Fig. 5.7. The result of flooding shows that when density is low, the fraction of isolated nodes (those that requires performing of route recovery) is significant. The network becomes fully connected when the number of nodes is larger than 400. The results of GUF under varying of sonobuoy density show that the performance of greedy upward forwarding is largely dependent on the sonobuoy density. As the number of sonobuoys increases, the reachability also increases (with diminishing returns). Interestingly, we found that infinite sonobuoys cannot attain the same reachability as is found in flooding due to voids. In fact, the gap between GUF with infinite sonobuoys and flooding represents the fraction

of nodes in the trap areas; i.e., these nodes require a route recovery mechanism to re-route packets to sonobuoys. This number is as large as 30% of the total number of nodes, especially when the density is low. As the number of sonobuoys decreases, we observe that the gap further increases. Although the network is fully connected, we see that if there is a single sonobuoy, almost 40% of the nodes suffer from voids; further, the number of trapped areas decreases as the density increases. The results clearly show the importance of providing a preventive measure for handling voids.

Considering a communication range of 250m and a 3D ocean cube size of $(1500m)^3$, optimal deployments require 91.67 nodes to cover the whole 3D ocean cube. Based on this reachability simulation result, we can claim that a 550 node scenario (i.e., roughly 6 nodes per $(250m)^3$ volume) with 64 sonobuoys can provide a reachability ratio of 1 to any of the sonobuoys; moreover, 600 deployed nodes with 1 sonobuoy cannot provide a reachability ratio of 1. To further observe how the number of sonobuoys can affect the protocols' behavior, we deployed a varying number of sonobuoys and provide simulation results for two extreme cases, namely 1 sonobuoy and 64 sonobuoys, under which we can clearly compare each protocol's behavior.

Fig. 5.8 examines the packet delivery ratio (PDR) of VAPR, HBR, HydroCast, and DBR with 1 sonobuoy on the surface. The packet delivery ratio of VAPR and HBR outperform those of the rest of the greedy forwarding protocols, namely HydroCast and DBR. This is because these two protocols provide a preventive measure by avoiding trap areas while maintaining only a soft-state in each node. The performance of VAPR is far better than that of HBR due to VAPR's localized opportunistic forwarding. Here, while the number of nodes increases, the PDR does not increase proportionally due to the increased number of retransmissions. More interestingly, the PDR of HydroCast does not increase as the number of deployed nodes increases. HydroCast's recovery process necessitates more frequent ring searches, thereby potentially creating more congestion in the acoustic channel, making it more difficult for HydroCast to deliver packets. This has the effect of diminishing the delivery ratio.

Fig. 5.9 shows the average energy consumption per message. Energy consumption per message decreases as the number of deployed nodes increases. A higher number of deployed nodes

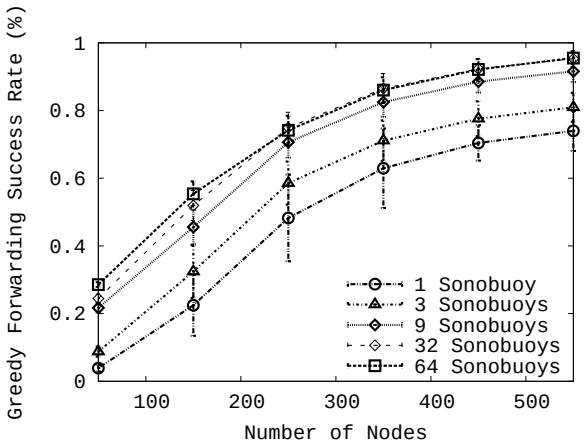


Figure 5.13: Fraction of trapped nodes as a function of the number of nodes, under different number of sonobuoys

consequently can yield more chances for greedy upward forwarding to succeed without requiring a route recovery process to reach any of the sonobuoys, resulting in less energy consumption per message. DBR's failure of suppressing the redundant packet transmissions causes excessive packet collisions and consumes much more energy than is used in HydroCast, HBR, and VAPR. HydroCast's recovery process near the surface has the effect of diminishing the delivery ratio while increasing the energy costs of HydroCast, in particular with lower node densities, creating more of a distinction from both VAPR and HBR. We note, however, that VAPR and HBR save more energy per packet than does HydroCast, as they do not require packet flooding for route recovery, thereby cutting down on the likelihood of packet collisions caused by channel congestion and improving the overall energy consumption.

Fig. 5.10 shows the average latency for all delivered packets. Here, DBR shows the worst performance due to failure of redundant packet suppressions, which causes congestion in the acoustic channel. Both VAPR and HBR outperform HydroCast with route recovery. The improvements are attributed to the clues provided during the beaconing process. Unlike HydroCast, packets generated from trapped nodes no longer require packet re-routing; instead they are routed directly to sonobuoys, without having to be first forwarded to local maxima nodes and then put through a

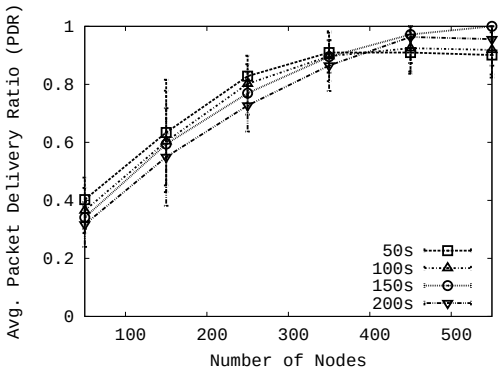


Figure 5.14: Effect of different beacon intervals on PDR (under MCM mobility)

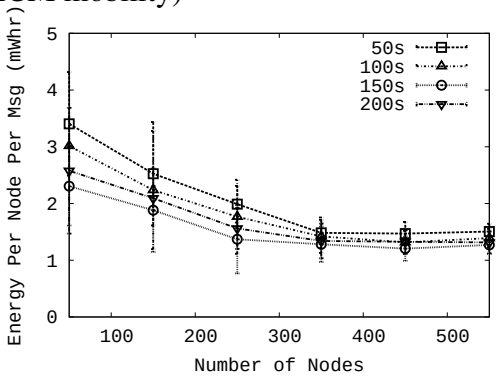


Figure 5.15: Effect of different beacon intervals on energy consumption (under MCM mobility)

recovery process. Note again that the difference between VAPR and HBR is caused by forwarding set selection granularity. In low density scenarios, opportunistic forwarding can improve the packet delivery ratio; similarly, in high density scenarios with 1 sonobuoy, opportunistic forwarding reduces the number of packet transmissions, thereby lowering the co-channel interference (and effectively handling the funneling effect).

Fig. 5.11 examines the packet delivery ratio of VAPR, HBR, HydroCast, and DBR with 64 sonobuoys on the surface. It is possible to see a general trend of positive correlation with node density in VAPR, HBR, and HydroCast but this is not the case with DBR due to the failure of

the redundant packet suppressions (which causes congestion in the acoustic channel leading to excessive packet collisions). The packet delivery ratio of HydroCast is similar to that of VAPR; this is because greedy forwarding with a sufficient number of sonobuoys and deployed nodes does not require any route recovery process. HBR cannot effectively handle channel fading, showing a lower PDR than those of Hydrocast and VAPR because HBR does not consider physical distance but hop counts, which means that nodes with the same hop count are treated equally although they have different advancements in terms of physical distances. Finally, this causes a smaller number of nodes to be considered as forwarding nodes. Recall that deployed nodes are moving on the basis of the MCM model (main jet stream speed of 0.3m/s). Unlike HydroCast, which uses explicit message exchanges to maintain a recovery path, VAPR is a soft-state routing protocol that is more resilient to node mobility and failure.

Fig. 5.12 shows the energy consumption per message with 64 sonobuoys. The overall trend of the four protocols is quite consistent with that in the previous results, shown in Fig. 5.9. DBR's failure of redundant packet suppression causes excessive packet collisions. As a result, DBR consumes much more energy than do other protocols. Due to its higher concentration of deployed sonobuoys on the surface, HydroCast does not require a route recovery process. Its performance consequently becomes similar to that of VAPR. We note, however, that HBR's performance does not increase like that of VAPR or HydroCast as the number of deployed nodes increase due to its absence of opportunistic forwarding.

To show the fraction of trapped nodes with respect to the number of sonobuoys, we vary the number of sonobuoys in a range from 1 to 64. As depicted in Fig. 5.13, the size of the trapped areas depends on the number of sonobuoys on the surface. The lower the number of sonobuoys, the larger the number of trapped nodes. The worst case would be one in which there is only a single sonobuoy. Also note that 3 percent of the nodes are trapped in the 32 and 64 sonobuoys scenarios with 550 deployed nodes, implying that 97 percent of the deployed nodes do not require route recovery to reach any of the sonobuoys (i.e., they are greedy upward forwarding nodes).

Evaluating the beacon interval for the VAPR based on MCM node mobility model is important

to show the beacon interval's sensitivity to the speed of node mobility—shorter beacon intervals cause unnecessary overhead while longer beacon intervals cause stale routing information amongst deployed nodes. In Fig. 5.14 and 5.15, we show the packet delivery ratio and the average energy consumption per message for VAPR with different beacon intervals of 50s, 100s, 150s, and 200s with MCM node mobility (0.3m/s). All intervals show positive correlation with node density. However, it can be seen that the beacon interval of 150s shows the best results. As depicted in Fig. 5.14, a beacon interval of 50s shows the best packet delivery ratio in low densities but saturates earlier than other intervals due to its frequent beacon message generation. However, the beacon interval of 150s shows stable and desirable performance regarding packet delivery ratio. As depicted in 5.15, energy consumption of the 150s beacon interval shows the best energy savings. It is noteworthy that the 200s beacon interval shows degraded energy performance compared to that of the 150s beacon interval in both low and high node densities. As beacon intervals become longer, the routing clues become stale. As a result, beacons provide less accurate routing information, which increases the energy consumption per message necessary to route a message to the sonobuoys.

5.6 Discussion

As illustrated earlier, one of the key design issues of opportunistic routing is the selection of a subset of neighbors that can make the best progress toward the destination without the hidden terminal problem. The major drawback of existing opportunistic routing such as ExOR [10] and LCOR [24] is that it requires global topology and link quality information (like link state routing) to find a set of forwarding groups toward the destination. Due to the protocol overhead, this approach is less suitable for mobile underwater sensor networks. An alternative to this approach is to augment existing routing protocols such as geographic routing and hop-based routing with *localized opportunistic forwarding*; i.e., a set of neighboring nodes that have shorter distance or hop-count than the current node can jointly forward a packet to make better progress toward the destination. In general, geographic routing can better exploit localized opportunistic forwarding

because the expected number of candidate nodes in geographic routing would be much greater than that in hop-based routing. In geographic routing, any node whose distance toward the destination is smaller than that of the current node (or any nodes in the advance zone) is considered, whereas in hop-based routing, any node whose hop-count toward the destination is smaller than that of the current node is only considered.[3] When comparing these approaches, we observe that a significant fraction of the candidate nodes in geographic routing may have the same hop count in hop-based routing. For example, assuming that the current forwarding node has five neighboring nodes in the advance zone of geographic routing, it is possible that only one node has lower hop count in hop-based routing (i.e., 5 vs. 1). This argument justifies our design choice of applying localized opportunistic forwarding to pressure routing, a specialized geographic routing scenario. In this article, we leave the performance comparison of these approaches as part of future work.

[3]As in VAPR, hop-based routing with opportunistic forwarding also requires efficient forwarding set selection methods that choose a subset of neighbors that make the best progress toward the destination, yet without the hidden terminal problem. One simple way would be modifying the greedy clustering method of HydroCast [59] (e.g., just finding the cluster with the largest number of nodes).

CHAPTER 6

Conclusions

In this dissertation, to realize SEA Swarm architecture, we have proposed an innovative MAC protocol, namely DOTS, to alleviate limitations caused by the long propagation latency and the severely limited bandwidth of acoustic communications. DOTS achieves better channel utilization by harnessing both temporal and spatial reuse. We also proposed HydroCast, a hydraulic pressure based anycast routing protocol with the salient features: novel opportunistic routing mechanisms to select the subset of forwarders that maximizes greedy progress, yet limits co-channel interference; and an efficient underwater *dead end* recovery method that outperforms recently proposed approaches (e.g., random walk, 3D flooding, etc). Finally, simulation results confirmed that both proposed protocols can effectively handle their own challenges.

Due to the unique characteristics of the limited bandwidth and high propagation latency coupled with high energy consumption of the acoustic medium, flooding and randomization may cause more harm than good. To improve the performance of HydroCast, we have proposed VAPR, a simple and robust soft-state protocol. VAPR exploits periodic beaconing to build directional trails toward the surface and features greedy opportunistic directional forwarding for packet delivery. We provided a detailed discussion on the loop free property of VAPR and showed that the sequence number propagation speed and the maximum depth are the key factors of ensuring loop-freedom in mobile networks. Our extensive simulations showed that VAPR outperforms existing schemes by significantly lowering the frequency of recovery fallbacks and by effectively handling node mobility.

REFERENCES

[1] MICA2: Wireless Measurement System. http://www.xbow.com/Prod\\ucts/Product_pdf_files/Wireless_pdf/6020-0042-0\%4_A_MICA2.pdf.

[2] Networked & Embedded Systems Laboratory (NESL). http://nesl.ee.ucla.edu/document/show/309.

[3] A. Acharya, A. Misra, and S. Bansal. Design and analysis of a cooperative medium access scheme for wireless mesh networks. In *BroadNets'04*, pages 621–631, Oct. 2004.

[4] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater Acoustic Sensor Networks: Research Challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257–279, March 2005.

[5] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, Aug 2002.

[6] Muhammad Ayaz, Imran Baig, Abdullah Azween, and Faye Ibrahima. A survey on routing techniques in underwater wireless sensor networks. *Journal of Network and Computer Applications*, 2011.

[7] D. Bertsekas and R. Gallager. Data Networks. NJ: *Prentice Hall*, 1992.

[8] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: a media access protocol for wireless LAN's. In *SIGCOMM'94*, pages 212–225, New York, NY, USA, 1994. ACM.

[9] P. Biswas and Y. Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. In *IPSN'04*, Apr. 2004.

[10] Sanjit Biswas and Robert Morris. Opportunistic Routing in Multi-Hop Wireless Networks. In *SIGCOMM'05*, Aug. 2005.

[11] R.V. Boppana and S.P. Konduru. An adaptive distance vector routing algorithm for mobile, ad hoc networks. In *INFOCOM*, 2001.

[12] Cecilia Carbonelli and Urbashi Mitra. Cooperative multihop communication for underwater acoustic networks. In *WUWNet'06*, pages 97–100, New York, NY, USA, 2006. ACM.

[13] A. Caruso, F. Paparella, L.F.M. Vieira, M. Erol, and M. Gerla. The Meandering Current Mobility Model and its Impact on Underwater Mobile Sensor Networks. In *In Infocom'08*, pages 221–225, April 2008.

[14] P. Casari, M. Rossi, and M. Zorzi. Towards optimal broadcasting policies for harq based on fountain codes in underwater networks. In *WONS*, 2008.

[15] Paolo Casari and Albert F. Harris. Energy-efficient reliable broadcast in underwater acoustic networks. In *WuWNet*, 2007.

[16] Paolo Casari and Michele Zorzi. Protocol design issues in underwater acoustic networks. *Computer Communications*, 2011.

[17] M. Cesana, D. Maniezzo, P. Bergamo, and M. Gerla. Interference aware (IA) MAC: an enhancement to IEEE802.11b DCF. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 5, pages 2799–2803 Vol.5, Oct. 2003.

[18] Vijay Chandrasekhar, Yoo Sang Choo, and How Voon Ee. Localization in Underwater Sensor Networks – Survey and Challenges. In *WUWNet'06*, Sep. 2006.

[19] Min Chen, S. Gonzalez, and V.C.M. Leung. Applications and design issues for mobile agents in wireless sensor networks. *Wireless Communications, IEEE*, 14(6):20–26, December 2007.

[20] N. Chirdchoo, Wee seng Soh, and Kee Chua. RIPT: A Receiver-Initiated Reservation-Based Protocol for Underwater Acoustic Networks. *Selected Areas in Communications, IEEE Journal on*, 26(9):1744–1753, December 2008.

[21] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). IETF RFC 3626, 2003.

[22] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A Decentralized Network Coordinate System. In *SIGCOMM'04*, Aug.-Sept. 2004.

[23] I. Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115–121, April 2006.

[24] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli. Least-Cost Opportunistic Routing. In *Allerton'07*, Sept. 2007.

[25] Stephane Durocher, David Kirkpatrick, and Lata Narayanan. On Routing with Guaranteed Delivery in Three-dimensional Ad Hoc Wireless Networks. In *ICDCN'08*, Jan. 2008.

[26] Stephane Durocher, David Kirkpatrick, and Lata Narayanan. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. *Wirel. Netw.*, 2010.

[27] H. Edelsbrunner, F. P. Preparata, and D. B. West. Tetrahedrizing Point Sets in Three Dimensions. *Symbolic and Algebraic Computation*, 1990.

[28] S.B. Eisenman and A.T. Campbell. E-CSMA: Supporting Enhanced CSMA Performance in Experimental Sensor Networks Using Per-Neighbor Transmission Probability Thresholds. In *In Infocom'07*, pages 1208–1216, May 2007.

[29] R Eustice, L Whitcomb, H Singh, and M Grund. Recent advances in synchronous-clock one-way-travel-time acoustic navigation. *OCEANS*, pages 1–6, 2006.

[30] Li Fan, Pei Cao, and Jussara Almeida. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. In *SIGCOMM'98*, Aug.-Sept. 1998.

[31] Roland Flury and Roger Wattenhofer. Randomized 3D Geographic Routing. In *INFOCOM'08*, Apr. 2008.

[32] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092 Vol. 2, Sept. 2005.

[33] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. In *Proc. ACM SIGCOMM 95*, Aug. 1995.

[34] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to Hidden Terminal Problems in Wireless Networks. In *Proc. ACM SIGCOMM 97*, pages 14–18, Sep. 1997.

[35] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Analysis of Aloha Protocols for Underwater Acoustic Sensor Networks. In *in Proc. ACM WUWNet 2006*, Sep. 2006.

[36] Stanley P. Y. Fung, Cao-An Wang, and Francis Y. L. Chin. Approximation Algorithms for Some Optimal 2D and 3D Triangulations. *Handbook of Approximation Algorithms and Metaheuristics, CRC Press*, 50, 2007.

[37] Holger Füßler, Jörg Widmer, Michael Käsemann, Martin Mauve, and Hannes Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks. *Elsevier Ad Hoc Networks*, 1(4):351–369, Nov. 2003.

[38] J. J. Garcia-Luna-Aceves, Marc Mosko, and Charles E. Perkins. A new approach to on-demand loop-free routing in ad hoc networks. In *PODC*, 2003.

[39] J.J. Garcia-Luna-Aceves and H. Rangarajan. A new framework for loop-free on-demand routing using destination sequence numbers. In *MASS*, 2004.

[40] David Goldenberg, Pascal Bihler, Ming Cao, Jia Fang, Brian D.O. Anderson, A. S. Morse, and Yang Richard Yang. Localization in Sparse Networks using Sweeps. In *MobiCom'06*, Sep. 2006.

[41] Xiaoxing Guo, M.R. Frater, and M.J. Ryan. A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks. In *OCEANS 2006 - Asia Pacific*, pages 1–6, May 2006.

[42] Reference has been removed for confidentiality.

[43] Tian He, John Stankovic, Chenyang Lu, and Tarek Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In *ICDCS'03*, May 2003.

[44] B.M. Howe and T. McGinnis. Sensor Networks for Cabled Ocean Observatories. In *Scientific Use of Submarine Cables and Related Technologies'03*, pages 216–221, June 2003.

[45] C. Hsu, K. Lai, C. Chou, and K. C. Lin. ST-MAC: Spatial-Temporal MAC Scheduling for Underwater Sensor Networks. In *In InfoCom'08*, May 2008.

[46] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *INMIC'01*, 2001.

[47] Jules Jaffe and Curt Schurgers. Sensor networks of freely drifting autonomous underwater explorers. In *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, pages 93–96, New York, NY, USA, 2006. ACM.

[48] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report TR-301, DEC Research, September 1984.

[49] Bjorn Jalving. Depth Accuracy in Seabed Mapping with Underwater Vehicles. In *Oceans'99 Riding the Crest into the 21st Century*, Sept. 1999.

[50] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, pages 153–181, 1996.

[51] Jonathan Friedman. Electrostatic Transconduction for Underwater Communication and Imaging. Ph.D. Thesis Prospectus TR-UCLA-NESL-200905-06, UCLA, May 2009. `http://nesl.ee.ucla.edu/document/show/309`.

[52] Josep Miquel Jornet, Milica Stojanovic, and Michele Zorzi. Focused Beam Routing Protocol for Underwater Acoustic Networks. In *WUWNet'08*, Sep. 2008.

[53] Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *MobiCom'00*, Aug. 2000.

[54] J. Kong, J. H. Cui, D. Wu, and M. Gerla. Building Underwater Ad-hoc Networks and Sensor Networks for Large Scale Real-time Aquatic Applications. In *IEE MILCOM'05*, Oct. 2005.

[55] K. Kredo, P. Djukic, and P. Mohapatra. STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol. In *In Infocom'09*, pages 2961–2965, April 2009.

[56] Seungjoon Lee, Bobby Bhattacharjee, and Suman Banerjee. Efficient Geographic Routing in Multihop Wireless Networks. In *MobiHoc'05*, May 2005.

[57] U. Lee, J. Kong, J. S. Park, E. Magistretti, and M. Gerla. Time-Critical Underwater Sensor Diffusion with No Proactive Exchanges and Negligible Reactive Floods. In *Proceedings of ISCC'06*, June 2006.

[58] Uichin Lee, Youngtae Noh, Paul Wang, Luiz F. M. Vieira, Mario Gerla, and Jun-Hong Cui. Pressure Routing for Underwater Sensor Networks. Technical report, UCLA, 2009.

[59] Uichin Lee, Paul Wang, Youngtae Noh, Luiz F. M. Vieira, Mario Gerla, and Jun-Hong Cui. Pressure routing for underwater sensor networks. In *INFOCOM*, 2010.

[60] Ben Leong, Barbara Liskov, and Robert Morris. Geographic Routing without Planarization. In *NSDI'06*, May 2006.

[61] Ke Liu and Nael B. Abu-Ghazaleh. Virtual Coordinate Backtracking for Void Traversal in Geographic Routing. In *ADHOC-NOW'06*, Aug. 2006.

[62] P. Lysanov L.M. Brekhovskikh, Yu. *Fundamentals of Ocean Acoustics 3rd Edition*. Springer, 2003.

[63] K. Mittal and E.M. Belding. RTSS/CTSS: mitigation of exposed terminals in static 802.11-based mesh networks. In *Wireless Mesh Networks, 2006. WiMesh'06*, pages 3–12, Sept. 2006.

[64] M. Molins and M. Stojanovic. Slotted FAMA: A MAC Protocol for Underwater Acoustic Networks. In *In Proc. of MTS/IEEE OCEANS06*, pages 1–7, Setember 2006.

[65] N. Nicolaou, A. See, Peng Xie, Jun-Hong Cui, and D. Maggiorini. Improving the robustness of location-based routing for underwater sensor networks. In *OCEANS*, 2007.

[66] P. Nicopolitidis, G.I. Papadimitriou, and A.S. Pomportsis. Adaptive data broadcasting in underwater wireless networks. *Oceanic Engineering, IEEE Journal of*, 2010.

[67] Youngtae Noh, Paul Wang, Uichin Lee, and Mario Gerla. Vapr: Void aware pressure routing protocol. In *WUWNet Work-In-Progress Poster*, 2010.

[68] J. Opatrny, A.E. Abdallah, and T. Fevens. Randomized 3d position-based routing algorithms for ad-hoc networks. In *Mobile and Ubiquitous Systems: Networking Services*, 2006.

[69] Vincent D. Park and Joseph P. Macker. Anycast routing for mobile services. In *Conf. Info. Sci. and Sys*, 1999.

[70] J. Partan, J. Kurose, and B. N. Levine. A Survey of Practical issues in Underwater Networks. In *WUWNet'06*, pages 17–4, September 2006.

[71] B. Peleato and M. Stojanovic. Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks. *Communications Letters, IEEE*, 11(12):1025 –1027, december 2007.

[72] Zheng Peng, Jun-Hong Cui, Bing Wang, Keenan Ball, and Lee Freitag. An underwater network testbed: design, implementation and measurement. In *WuWNet'07*, pages 65–72, New York, NY, USA, 2007. ACM.

[73] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM*, 1994.

[74] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Computing Sys. and Apps.*, 1997.

[75] Sameera Poduri, Sundeep Pattem, Bhaskar Krishnamachari, and Gaurav S. Sukhatme. Sensor Network Configuration and the Curse of Dimensionality. In *EmNets'06*, May 2006.

[76] D. Pompili, T. Melodia, and I. F. Akyildiz. A CDMA-based Medium Access Control Protocol for Underwater Acoustic Sensor Networks. *IEEE Transactions on Wireless Communications*, 8(4):1899–1909, April 2009.

[77] Dario Pompili, Tommaso Melodia, and Ian F. Akyildiz. Deployment analysis in underwater acoustic wireless sensor networks. In *WUWNet'06*, pages 48–55, New York, NY, USA, 2006. ACM.

[78] J. Preisig. Acoustics Propagation Considerations for Acoustics Communications Network Development. In *In Proc. of ACM WUWNet'06*, 2006.

[79] Theodore Rappaport. *Wireless Communications: Principles and Practice*. *Prentice Hall PTR*, Upper Saddle River, NJ, USA, 2001.

[80] V. Ravelomanana. Extremal properties of three-dimensional sensor networks with applications. *Mobile Computing, IEEE Transactions on*, 3(3):246–257, July-Aug. 2004.

[81] Lawrence G. Roberts. ALOHA packet system with and without slots and capture. *SIGCOMM Comput. Commun. Rev.*, 5(2):28–42, 1975.

[82] J. Ruppert and R. Seidel. On the Difficulty of Tetrahedralizing 3-Dimensional Non-convex Polyhedra. In *ACM Symposium on Computational Geometry*, June 1989.

[83] Prince Samar and Stephen B. Wicker. On the behavior of communication links of a node in a multi-hop mobile environment. In *MobiHoc*, 2004.

[84] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic Fine-Grained Localization in Ad-Hoc Wireless Sensor Networks. In *MobiCom'01*, July 2001.

[85] Deepanshu Shukla, Leena Chandran-Wadia, and Sridhar Iyer. Mitigating the exposed node problem in IEEE 802.11 ad hoc networks. In *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*, pages 157–162, Oct. 2003.

[86] E. Sozer, M. Stojanovic, and J. Proakis. Underwater Acoustic Networks. *IEEE Journal of Oceanic Engineering*, 25(1):72–83, Jan. 2000.

[87] M. Stojanovic. Acoustic (Underwater) Communications. in J. G. Proakis, Editor, Encyclopedia of Telecommunications, *John Wiley and Sons*, 2003.

[88] Milica Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(4):34–43, 2007.

[89] Milica Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. *in SIGMOBILE*, 2007.

[90] A. Syed, Wei Ye, and J. Heidemann. Comparison and Evaluation of the T-Lohi MAC for Underwater Acoustic Sensor Networks. *Selected Areas in Communications, IEEE Journal on*, 26(9):1731–1743, December 2008.

[91] A. A. Syed and J. Heidemann. Time Synchronization for High Latency Acoustic Networks. In *In Infocom'06*, pages 1–12, April 2006.

[92] Affan A. Syed, Wei Ye, and John Heidemann. T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks. In *In Infocom'08*, pages 231–235, April 2008.

[93] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data Collection, Storage, and Retrieval with an Underwater Network. In *Sensys'05*, Nov. 2005.

[94] L.F.M. Vieira, Uichin Lee, and M. Gerla. Phero-trail: a bio-inspired location service for mobile underwater sensor networks. *Selected Areas in Communications, IEEE Journal on*, 2010.

[95] Jack Wills, Wei Ye, and John Heidemann. Low-power acoustic modem for dense underwater sensor networks. In *WUWNet'06*, pages 79–85, New York, NY, USA, 2006. ACM.

[96] P. Xie and J. H. Cui. R-MAC An Energy-Efficient MAC Protocol for Underwater Sensor Networks. In *In Proc. of the IEEE WASA'07*, pages 187 – 198, August 2007.

[97] Peng Xie, Jun-Hong Cui, and Li Lao. Vbf: Vector-based forwarding protocol for underwater sensor networks. In *Networking*, 2006.

[98] Justin Yackoski and Chien-Chung Shen. UW-FLASHR: Achieving High Channel Utilization in a Time-based Acoustic MAC Protocol. In *WUWNet'08*, pages 59–66, New York, NY, USA, 2008. ACM.

[99] Hai Yan, Zhijie Shi, and Jun-Hong Cui. DBR: Depth-Based Routing for Underwater Sensor Networks. In *IFIP Networking'08*, May 2008.

[100] Kai Zeng, Wenjing Lou, Jie Yang, and D. Richard Brown. On Geographic Collaborative Forwarding in Wireless Ad Hoc and Sensor Networks. In *WASA'07*, Aug. 2007.

[101] Wei Zeng, Rik Sarkar, Feng Luo, Xianfeng Gu, and Jie Gao. Resilient routing for sensor networks using hyperbolic embedding of universal covering space. In *INFOCOM*, 2010.

[102] Bin Zhang, G.S. Sukhatme, and A.A. Requicha. Adaptive sampling for marine microorganism monitoring. In *Intelligent Robots and Systems (IROS'04)*, volume 2, pages 1115–1122 vol.2, Sept.-2 Oct. 2004.

[103] Biao Zhou, Yeng Z. Lee, M. Gerla, and F. de Rango. Geo-LANMAR: A Scalable Routing Protocol for Ad Hoc Networks with Group Motion. *Wireless Communications and Mobile Computing*, 6(7):989–1002, 2006.

[104] Jiangwei Zhou, Yu Chen, Ben Leong, and Pratibha Sundar Sundaramoorthy. Practical 3d geographic routing for wireless sensor networks. In *SenSys*, 2010.

[105] Zhong Zhou and Jun-Hong Cui. Energy Efficient Multi-Path Communication for Time-Critical Applications in Underwater Sensor Networks. In *MobiHoc'08*, May 2008.

[106] Zhong Zhou, Jun-Hong Cui, and Amvrossios Bagtzoglou. Scalable Localization with Mobility Prediction for Underwater Sensor Networks. In *INFOCOM'08*, Apr. 2008.

[107] Michele Zorzi and Ramesh R. Rao. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003.