RELIABLE UNICAST AND GEOCAST PROTOCOLS FOR UNDERWATER INTER-VEHICLE COMMUNICATIONS

BY AMRITA NIMBALKAR

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Dario Pompili

and approved by

____

____

____

New Brunswick, New Jersey

October, 2008

ABSTRACT OF THE THESIS

Reliable Unicast and Geocast Protocols for Underwater Inter-Vehicle Communications

by Amrita Nimbalkar

Thesis Director: Professor Dario Pompili

Underwater networks are envisioned to enable several applications for oceanographic data collection, environmental monitoring, navigation and tactical surveillance. Many applications make use of Autonomous Underwater Vehicles (AUVs) equipped with underwater sensors. Underwater communication links are based on *acoustic wireless technology*, which poses challenges due to the unique underwater environment such as high propagation delays, high bit error rates, and temporary losses of connectivity caused by multipath and fading phenomena. For data collection and monitoring tasks, underwater vehicles can either periodically send the measured data to the surface station (sink) or the sink can initiate a query to the sensors asking for the information of interest. The former case is reduced to unicasting, where the data is sent periodically by nodes to a specific destination, i.e., the surface station. In the later case, query dissemination can involve either broadcasting or geocasting technique, depending on whether the query is sent to all the nodes, or a subset of nodes based on location respectively. As broadcasting can be viewed as a special case of geocasting, geocast protocols provide a general routing scheme for query dissemination. In either of the cases, reliability is a crucial factor for underwater communications.

Reliability, especially in a mobile environment, is a major concern due to network

dynamics. Due to the high propagation delays involved in underwater communications, we do not consider transport solutions for reliable communications. Rather, we consider the lower layers for ensuring reliability. In this work, three versions of unicasting and geocasting protocols have been proposed, which integrate Medium Access Control (MAC) and routing functionalities and leverage different levels of neighbor knowledge for making optimum routing decisions. Performance evaluation has been done for unicast protocols in terms of different end-to-end metrics, for static and mobile scenarios, with an aim of finding an optimal level of neighbor knowledge required in either of these scenarios. It is observed that based on different end-to-end metrics considered, one version of unicast protocol outperforms the other. Thus, based on the application requirements and scenarios considered, an optimum level of neighbor knowledge can be utilized for periodic data delivery from nodes to the surface station.

Acknowledgements

At the end of this two-year journey leading to masters dissertation, I realize the importance of acknowledging several people I am indebted to. First and foremost, I would like to thank Prof. Dario Pompili for his consistent support, guidance and enthusiasm throughout the course of this thesis. His incredible perception into problems has taught me a lot on conducting research. Little can I forget the long hours of constructive meetings that I had with Prof. Pompili in tackling a particular problem, and finding a solution to it. In addition to this, his fruitful criticism and comments have greatly helped me enhance my technical writing skills.

I definitely cant miss this opportunity to thank my friend Ayesha, who had been my survivor in difficult times. I thank her for sparing time to listen to me, and give practical comments and suggestions. I cant forget the long hours of discussions we have had over my work that she used to sit through and listen, no matter how busy her schedule was. Closely following up, I would also like to thank my friend Tanuja for the enlightening discussions we have had on a daily basis during the course of my thesis, and for motivating me to survive the process of research. I also thank my friends Sumit and Salmaan for participating in useful discussions at certain times when I was stuck up with my work. Last but not the least, I remember my parents with gratitude for the way they have brought me up over the years and motivated me to do well in every walk of life.

Dedication

To my mother Ms. Sushma Nimbalkar

my late father Mr. Ashok Nimbalkar

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Underwater sensor networks have the potential to enable unexplored applications and to enhance our ability to observe and predict the ocean. Such applications include exploration of natural undersea resources, navigation, tactical surveillance and gathering of scientific data in collaborative monitoring missions. UnderWater Acoustic Sensor Networks (UW-ASNs), which consists of stationary sensor devices and underwater vehicles, enable these applications. Unmanned or Autonomous Underwater Vehicles (UUVs, AUVs), equipped with underwater sensors have multitude of applications in underwater resource study owing to their flexibility, i.e, they can function without tethers, cables or remote control and can reach any depth in the ocean [1]. The potential applications will be made viable by enabling communications among underwater devices. These communication links are based on *acoustic wireless technology.*

1.1 Underwater Autonomous Vehicles

One of the design objectives of AUVs is to make them rely on local intelligence and be less dependent on communications from online shores [2]. In general, control strategies are needed for autonomous coordination, obstacle avoidance, and steering strategies. Solar energy systems allow increasing the lifetime of AUVs, i.e., it is not necessary to recover and recharge the vehicle on a daily basis. Hence, solar powered AUVs can acquire continuous information for periods of time of the order of months. A reference architecture for 3D UW-ASNs with AUVs is shown in Fig. 1.1 [3]. Several types of AUVs exist as experimental platforms for underwater experiments. Some of them resemble small-scale submarines (such as the Odyssey-class AUVs developed at MIT). Others are simpler devices that do not encompass such sophisticated capabilities. For

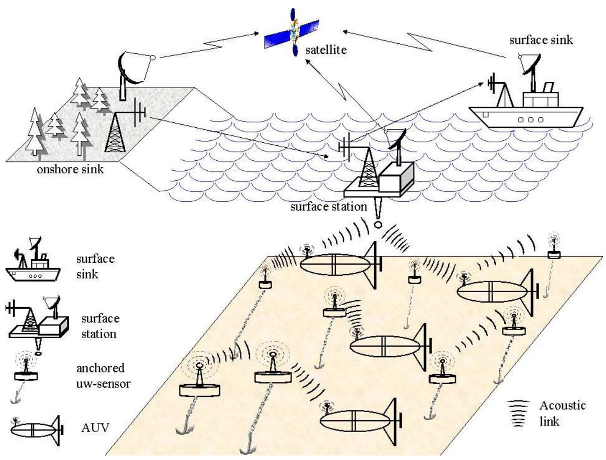

Figure 1.1: 3D Underwater Sensor Networks with AUVs

example, *drifters* and *gliders* are oceanographic instruments often used in underwater explorations. Drifter underwater vehicles drift with local current and have the ability to move vertically through the water column, and are used for taking measurements at preset depths [4]. Underwater gliders [5] are battery powered autonomous underwater vehicles that use hydraulic pumps to vary their volume by a few hundred cubic centimeters in order to generate the buoyancy changes that power their forward gliding. In our work, we consider conventional AUV's as well as gliders specifically, which differ in their motion. Underwater gliders alternately reduce and expand displaced volume to dive and climb through the ocean. Thus, they follow a sawtooth path as they glide from the ocean surface to the bottom of the ocean and back while sampling the ocean in space and time. Although gliders have lower velocities compared to conventional they AUVs, they permit longer-duration operations.

1.2 Underwater Acoustic Communication Challenges

UW-ASN communication links are based on *acoustic wireless technology*, which poses challenges due to the unique underwater environment such as limited bandwidth capacity [6], high and variable propagation delays [7], high bit error rates, and temporary losses of connectivity caused by multipath and fading phenomena [8]. The available bandwidth of the *UnderWater Acoustic Channel* (UW-A) depends on both range as well as frequency. Long-range systems that operate over several tens of kilometers may have a bandwidth of only few kHz, while a short-range system operating over several tens of meters may have more than a hundred kHz of bandwidth [3]. The challenges faced by underwater channel for underwater sensor networking are:

1. Path loss:

    - Attenuation. It is caused by absorption due to the conversion of acoustic energy into heat. The attenuation increases with increase in distance and frequency [3].

    - Geometric spreading. This refers to the spreading of sound energy as a result of the expansion of the wavefronts. It increases with the propagation distance and is independent of frequency. There are two common kinds of geometric spreading: spherical(omni-directional point source) and cylindrical(horizontal radiation only).

2. Noise:

    - Man made noise. This is mainly caused by machinery noise and shipping activity, especially in areas with heavy vessel traffic.

    - Ambient noise. It is related to movement of water including tides, current, storm, wind, and to seismic and biological phenomena.

3. High delay and delay variance:

    - The propagation speed in the UW-A channel is five orders of magnitude lower than in the radio channel. The large propagation delay (0.67s/Km) can reduce the throughput of the system considerably.

    - The very high delay variance is even more harmful for efficient protocol design, as it prevents from accurately estimating the round trip time (RTT), which is the key parameter for many communication protocols.

Owing to the peculiar characteristics of the underwater environment, reliable communication is a fundamental primitive for underwater networks. Reliability is critical for specific sensor network applications such as tactical surveillance and disaster prevention. Consider an application where an UW-ASN is deployed to detect seismic activity on the sea bed to provide tsunami warnings. In this network, the sensors can report the magnitude of seismic waves if the magnitude is above a certain threshold. Given the nature of the application, it is critical that the data packets from the sensors reach the surface station in a reliable manner. In multihop networks, reliability can be defined on a *hop-by-hop* and on an *end-to-end* basis. Hop-by-hop reliability ensures successful delivery of messages between each pair of nodes in a network, whereas, end-to-end reliability ensures successful delivery of messages between the source node and the destination node. However, a sequence of hop-by-hop guarantees does not necessarily add up to an end-to-end guarantee. For example, consider nodes A, B, C where A is the source, C is the destination and B is an intermediate node. On successful reception of packet from A, B sends an ACK to A. As A receives an ACK from B, it transfers the responsibility of forwarding the data packet to B. After B receives packet from A, either of the situations might arise: (i) B fails, (ii) B runs out of energy, (iii) B moves out of range because of mobility, (iv) B gets disconnected because of channel impairments. Even though the link from B to C is reliable in these cases, there is no guarantee of reliable delivery from A to C. Thus we cannot have 100% end-to-end reliability by providing only link-layer reliability.

In terrestrial wireless networks, end-to-end reliability is provided by the transport layer. The transport solutions mostly focus on reliable data transmission following end-to-end Transmission Control Protocol (TCP) semantics [9]. In TCP [10], providing an end-to-end guarantee requires an acknowledgement of the message by the final destination to the source. In underwater environment, the delay is five orders of magnitude higher as compared to the delay in terrestrial sensor networks [3]. Although transport solutions are crucial for reliable communication, end-to-end reliability would result in large Round Trip Times (RTTs). This in turn would severely affect the end-to-end throughput that can be achieved over a network, which decreases as RTT increases [11]. Thus we consider lower layers for reliable communication. In this work, we maximize end-to-end reliability by providing high link-layer reliability.

Ensuring link-layer reliability, especially in the case of UW-ASNs consisting of AUVs, is a challenging task due to the dynamic nature of the network topology posed by channel impairments and vehicle mobility. In case of mobility, some amount of neighborhood knowledge and topology information would potentially improve reliability. As the level of neighbor knowledge for a node increases, the reliability is expected to improve. However, the task of maintaining an updated topology becomes more challenging, as it requires more frequent exchange of control messages. Thus, even though the number of collisions between data packets are likely to decrease with increase in the neighbor knowledge, collisions between the data packets and the control messages are likely to increase. Our aim is to ensure link-layer reliability by using cross-layered interactions between MAC and routing layers and optimizing different levels of neighbor knowledge: (i) no neighbor knowledge, (ii) one-hop neighbor knowledge, and (iii) two-hop neighbor knowledge.

1.3 Thesis Overview

For data collection and monitoring tasks, the sensors can either periodically send the measured data to the surface station (sink) or the sink can initiate a query to the sensors asking for the information of interest. In the former case, we propose three versions of unicasting protocols for reliable data delivery, between the surface station and all the nodes or subset of nodes receiving the query, based on different levels of neighbor knowledge. In the later case we consider geocasting schemes for disseminating the query from the surface station. The protocols integrate MAC and routing layer functionalities, to route the packets from source to the destination. The protocols use random-access MAC, i.e., the AUVs access the channel in an uncoordinated manner. This is done to account for the fact that most of the state-of-the-art underwater acoustic modems use random access MAC scheme. We have used geographical routing mechanism, since underwater vehicles need to estimate their current position irrespective of the routing approach, as it is necessary to associate the sampled data with the 3D position of the device that generates the data and spatially reconstruct the characteristics of the event. We assume that the nodes in the network know their geographical co-ordinates.

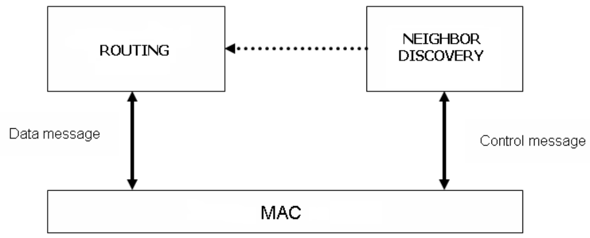

Figure 1.2: Interaction between different functionalities

Moreover, since the nodes in the underwater network are sparsely deployed, IDs are assigned to the nodes.

In this work, we study the operation regions in which one version of the protocol behaves better than the other. We compare the performances of the three protocols and determine which protocol outperforms the other in terms of a several end-to-end metrics. The features of our work lie in the following:

1. The channel is considered to be asymmetric, i.e., the channel conditions between two nodes in underwater environment are not considered to be the same in both the directions;

2. A node transmits a packet at maximum power level, but we do not fix up the range such that all the nodes lying within that range always receive the packet. Thus we do not use Unit Diskgraph Model in design and implementation. The reception of packets by the nodes is based on the channel dynamics;

3. We have accounted for spatial and temporal variations in the channel conditions, which is a peculiar characteristic of the underwater channel;

4. We have integrated the functionalities of MAC and routing layers and the neighbor discovery phase, to improve the performance of the cross-layered protocols by leveraging the neighbor knowledge for making optimum decisions.

Thus, the reliable performance of the protocols depend on the interaction between the following communication functionalities as shown in the Fig.1.2. Considering the pros and cons of each level of neighbor knowledge, the performance of the three protocols is evaluated in a mobile environment, to find an optimal level of neighbor knowledge that ensures reliability based on the application requirements. We have first defined hop-by-hop and end-to-end reliability, and motivated the use of the former in the proposed solutions, for underwater communications. The key features of the proposed solutions for unicasting phase:

- no neighbor knowledge based protocol:

  Provides fairness by synchronizing the hold-off timers of all the nodes that receive the data packet

- one-hop neighbor knowledge based protocol:

  Reduces collisions between the reception of transmissions of the neighboring nodes, by making use of the neighbor knowledge, and thus adjusting the hold-off timers dynamically

- two-hop neighbor knowledge based protocol:

  Reduces collisions of transmissions at the first and second hop distances, by making use of the neighbor knowledge and thus adjusting the timeout timers dynamically

Further, all the three protocols de-synchronize the transmission of ACK's with the transmission of data packets, on reception of duplicate data packets from the same sender. For this purpose, the unicast protocols with neighbor knowledge adjust the hold-off time of ACK packets dynamically. No neighbor knowledge based unicast protocol uses a sufficiently large value of hold-off time for ACK, to de-synchronize the propagation of ACK with data packets.

In this work, we have also accounted for the unique channel characteristics of underwater environment, in performing the performance evaluation of the protocols. In particular, we have considered channel asymmetry and its spatio-temporal variation to model statistical transmission losses in an underwater channel. The work is divided organized into different chapters. Since query dissemination (geocasting) and data collection (unicasting) both incur unicasting phase (geocast protocol requires unicast phase to reach up to the geocast region from the surface station), the different versions of unicast protocol have been analyzed before geocast protocol.

In chapter 2, we review the existing unicast and geocast protocols for mobile ad hoc networks. We consider the peculiar characteristics of underwater environment, and discuss the suitability of the existing protocols for underwater communications. We also review the existing work in UW-ASNs and discuss their pros and cons.

In chapter 3, we discuss the three versions of our unicast protocol employed by all the nodes or subset of nodes in response to the query dissemination by the surface station. We explain how neighborhood information in each version of the protocol is used in the formulation of cross-layered scheme between the MAC and routing layers, to take optimum decisions. In the process, we state the key features of the three schemes, which enhance their reliability.

In chapter 4, we introduce three versions of geocast protocol employed by nodes inside a region defined by the surface station, for the purpose of disseminating a query initiated by the surface station. Like unicast schemes, geocasting also makes use of interactions between MAC and routing layers to ensure reliability. We also explain how the shape of the geocast region is leveraged for the purpose of query dissemination and how the schemes for geocasting the query are decoupled from the schemes for unicasting data, in response to the query.

In chapter 5, we present the simulation results of the unicast protocol. In Sect. 5.1, we model the behavior of underwater channel by taking its peculiar characteristics into account. Further in Sect. 5.2, we compare the simulation performances of the three versions of unicasting in terms of end-to-end metrics such as packet delivery ratio, delay and energy/bit. In addition to this, we also compare the performance of each version of the protocol with new schemes having equivalent amount of neighborhood information but not fully exploiting the information possessed, i.e., they do not employ the key features of our proposed schemes, which help the later take better routing decisions.

In chapter 6, we draw conclusions from the simulation results obtained in chapter 3 and discuss the scope of future work for chapter 4. From the simulation results, we observe that based on the end-to-end metric considered, one version of unicast protocol outperforms the others. Consequently, there is a trade-off involved in the performance of the three versions of protocol. Thus depending on specific classes of applications, optimum amount of neighborhood information and corresponding versions of unicast protocol can be utilized.

Chapter 2

Related Work

In this chapter, we discuss unicast and broadcast protocols available for the terrestrial sensor networks and mobile ad-hoc networks. In addition to these protocols, multicast solutions that send packets to group of destinations are also considered. Further, we evaluate the feasibility of extending these protocols in underwater environment.

2.1 Unicast Protocols

Proactive protocols (e.g., DSDV [12] and OLSR [13]) involve a large signaling overheard to establish routes for the first time and each time the network topology changes due to mobility or node failures. This is done to maintain an updated routing information at all times from each node to every other node. In the process, control packets are broadcasted that contain routing table information. In this way, each node is able to establish a path to any other node in the network, which may neither be needed, nor be feasible in underwater networks.

In reactive protocols (e.g., AODV [14] and DSR [15]), nodes initiate a route discovery process only when a route to the destination is needed. Thus reactive protocols are more appropriate for dynamic environments but incur high latency since they also require source-initiated flooding of control packets to establish path from source to the destination. Thus like proactive protocols, reactive protocols also involve excessive signaling overhead due to their extensive reliance on flooding.

The Geographical routing protocol GFG [16] models ad hoc wireless networks as *unit graphs* in which nodes are points in plane and two nodes can communicate if the distance between them is less than a fixed point. However, this model may not be apt for underwater environment due to the statistical nature of the channel, i.e, the underwater channel is temporally and spatially variable. Another geographical routing protocol PTKF [17] aims at finding an *optimal topology knowledge range* for each node to make energy efficient geographical routing decisions. However, energy is not the foremost objective of our work. We want to ensure reliability and compare the performance of different knowledge based solutions primarily in terms of reliability.

In [18], the authors propose a network layer protocol for UnderWater Acoustic Networking (UAN) that autonomously establishes the network topology and controls network resources and flows. The process is centralized and relies on network manager running on a surface station. The manager establishes optimum paths for data delivery by ensuring some form of quality of service. However, the solutions lack detailed analysis and the impact of node mobility on protocol convergence still needs to be determined.

The authors in [6] provide a simple example of shallow water network, where route establishment is a centralized process. It is done with the help of neighborhood information, which is obtained with the help of poll packets exchanged between the nodes. However, the criteria used to select the data paths is not discussed. In addition to this, the sensors are deployed linearly. Whereas in underwater environment, it is necessary to consider the 3D deployment of the sensors.

In [19], a long-term monitoring platform for static and mobile nodes is proposed, and hardware and software architectures are described. The nodes communicate point-to-point using optical communication system. The mobile nodes can hover and locate over the static nodes for data muling. However, communication is enabled only when static and mobile nodes are in close proximity due to the limitations of optical transmissions.

2.2 Geocast and Broadcast Protocols

In [20], solutions have been proposed for different application requirements in underwater sensor networks, with the objective of minimizing the energy consumption. A model characterizing the acoustic channel utilization efficiency was developed to investigate fundamental characteristics of underwater environment, and to set up an optimal packet size for underwater communications based on the applications. However, the proposed solutions do not ensure reliability, which itself is the objective of our work. Moreover, the paper does not cater mobility in the work.

In [21], the authors propose a scheme for reliable communication in UW-ASNs. To ensure reliability, the authors propose a separate control and data channel. A control packet is first transmitted on a low frequency, high power level, followed by data broadcast on a high frequency, low power level. Since the transmission loss is lower at lower frequencies, the control packet has a higher probability of being received by the nodes than the data message. The nodes contend for rebroadcasting the data message using random back-off timers. A node that has received the control message but fails to receive the data message broadcasts a NACK to inform the neighboring nodes. Neighboring nodes, in turn, then broadcast the message. However, in characterizing the channel, the authors have only taken into account the deterministic transmission losses and neglected the statistical nature of the channel. Further, in this work, the selection of nodes that broadcast the data is random. In our work, we have made use of the geographical location of the nodes as well as that of the destination, to give forwarding priority to the nodes closer to the destination (surface station) for unicasting data packets in response to the query. This promotes faster propagation of data packets towards the surface station. In geocasting schemes, forwarding priority of a node inside the geocast region is based on the position of node with respect to the sender of query packet as well as the node's position within the geocast region. The aim is to propagate query along the length of the geocast region with fewer number of transmissions.

In [22], the authors propose a directional propagation scheme for emergency messaging in the vehicular network. Nodes at a greater distance from the sender node are given a greater priority to transmit. A counter is kept by all the nodes that keeps track of the number of overhearings of the packet transmissions. If the counter goes beyond a certain threshold, the node decides to be a non-forwarding node. For dense networks, inter-arrival time of packets is also considered by the nodes. A node that has experienced a higher packet inter-arrival time at its receiver has a greater probability of forwarding a packet. This strategy is meant to reduce the number of collisions in a dense network. In a separate scheme, the paper addresses directional propagation by requiring each node to be a forwarding node if it does not overhear the packet from the forward direction. The hold-off timer is used in this scheme to delay packet forwarding, which is not distance dependent. In underwater environment this creates a greater redundancy in packet transmission and higher chances of collision because of large propagation delays.

In [23], the authors propose an optimized Broadcast Protocol for Sensor networks (BPS) that minimizes the number of retransmissions by maximizing each hop length. The algorithm divides the communication plane into hexagons as per [24]. The "broadcast storm problem" [25] is then addressed by proposing that only the nodes located near the vertices of the hexagons relay the message. Since the aim of the paper is to optimize the broadcast, it does not give priority to propagation in a certain direction. However, in geocasting, propagation in a certain direction can result in faster penetration of message in the geocast region. To achieve this, we have enhanced the idea of maximizing the hop length by introducing the directivity in propagation.

In [26], the authors discuss the location-aided multicast algorithms in terrestrial sensor networks. The paper proposes three heuristic algorithms: Single brAnch Regional Flooding (SARF), Single brAnch Multicast tree (SAM) and Cone based Forwarding Area Multicast tree (CoFAM) to construct a multicast tree rooted at source and deliver the packets to nodes in a geographic location. SARF and SAM construct the shortest path from sink to a node called Access Point (AP) that belongs to geographic region. The packet delivery to destinations is done by Access Point using flooding in the geographic region. The difference between SARF and SAM is in the criteria of selecting an AP node in the geographic region. In SARF, AP is the node that is in the center of the geographic region, whereas in SAM, AP is the node that is closest to the sink and belongs to the geographic region. On the other hand, CoFAM, instead of designating a node as AP, employs the concept of limited flooding within a cone shaped forwarding area as described in [27]. None of these algorithms ensure reliable delivery of packets that is necessary for query dissemination in underwater environment. Unlike terrestrial sensor networks, sensors in underwater are sparsely deployed and information sent by sensors to sink may not have correlation. Thus, in underwater scenario it becomes necessary that all the nodes receive the query and depending on the nature of the query, send information to the sink.

In [27], the authors describe two location-based multi-cast algorithms for mobile ad-hoc networks. The algorithms use a forwarding zone that surrounds the geocast area to reach nodes in a geographic region. In the paper, algorithm I uses the concept of restricted flooding, which means that the nodes outside the forwarding zone will not forward the packet. On the other hand, algorithm II uses the concept of positive advance to give priority to the nodes closer to the center of geocast region to forward the packet. Both the algorithms result in a high number of collisions that not only delays message propagation inside the forwarding zone but also makes the transmissions unreliable.

The concept of dominating sets has been used in [28], [29], and [30] for efficient broadcasting. Consider a graph G with nodes in a wireless network as vertices. A set is dominating if all the nodes in G are either in the set or neighbors of nodes in the set. The broadcast algorithm proposed in [28] applies the concept of localized dominating sets using one-hop information. A set of internal (forwarding) nodes is selected using the list of neighbors and their geographic positions maintained by each node. The number of internal nodes is further reduced by neighbor elimination algorithm. The algorithm thus achieves reachability with considerable reduction in rebroadcast messages. In [29], the authors propose a self pruning scheme where the aim is to form a Connected Dominating Set (CDS) in a distributed manner. A node chooses itself to be a non-forwarding node if its k-hop neighborhood has the nodes with higher priority or nodes that have already received the data packet. It is shown in the paper that an efficient decision can be taken with two to three hop neighborhood information and one-hop routing history. This paper leaves open the definition of priority which can be based on node-id, node-degree, or any other metric. In [30], the authors propose that each node determines the priority of its neighbor nodes. The idea is to find the shortest path between all the possible pairs of nodes in its two-hop neighborhood. Then the neighbor node that is involved in the highest number of paths is given the highest priority. Based on this metric, a node sorts its neighbors in descending order and determines which one-hop neighbors should be selected so as to cover the complete two-hop neighborhood.

In our work, we have used the idea of neighbor elimination for efficient broadcasting inside the geocast region for two-hop neighbor knowledge based protocol. In addition, we have enhanced our algorithm to ensure reliability and promote faster propagation of query packets.

Chapter 3

Reliable Unicast Protocol for Underwater Inter-Vehicle Communications

This chapter describes the different versions of unicasting protocol employed by nodes for reliable data delivery to the surface station. The three versions of protocol differ in the level of neighbor knowledge based on which they make MAC and routing decisions. The different levels of neighbor knowledge used are: (i) no neighbor knowledge, (ii) one-hop neighbor knowledge, (iii) two-hop neighbor knowledge. The concept of neighborhood is defined statistically. If a node is able to receive 85% of the packets from another node, the latter is defined as its one-hop neighbor. By two-hop neighbor knowledge, we mean that a node has information about the one-hop neighbors of its own one-hop neighbors. In the neighbor discovery phase, the size of control messages is fixed for one-hop neighbor knowledge. Whereas, it is variable for two hop neighbor knowledge, since a node includes the entire list of its neighbors, i.e, their ID's and geographical co-ordinates, in the data packet. Consequently, the size of control messages depends on the number of neighbors.

In the protocol with no neighbor knowledge as described in Sect. 4.0.4, the routing decisions are made by the receiving nodes themselves with the help of their own position and the destination's position. Since there is no neighbor knowledge, a node receiving a packet in turn broadcasts it using MAC scheme. In one-hop neighbor knowledge based protocol in 4.0.5, the receiver of data packet makes use of its one-hop neighborhood information to select the neighbor closest to the destination as its next hop and unicast the data packet. Whereas, in case of two-hop neighbor knowledge based protocol as described in 4.0.6, the next hop is designated by the sender and the receiver selects the next two-hop based on its two-hop neighbor knowledge to make an optimum routing decision. Thus the protocols with neighbor knowledge make use of the location information of their neighbors, which is exchanged during the neighbor discovery phase, to make routing decisions. Moreover, the MAC scheme is devised by taking into account neighbor knowledge so as to avoid collisions at the neighbors and decrease the number of retransmissions. Consequently, it improves the reliability of the protocols.

3.0.1 No Neighbor Knowledge

Because there is no neighbor discovery phase in this protocol, a node does not have information about its neighbors. The main idea is to reach the destination reliably using *limited-flooding*. This is done to de-synchronize the transmissions from different nodes and avoid collisions. A node receiving a packet decides itself whether it should be a forwarding or non-forwarding node. The MAC scheme, which is devised with an aim of de-synchronizing transmissions, reducing retransmissions and avoiding collisions is described below:

If a node receives the data packet for the first time from a farther node (i.e., the receiving node is closer to the destination compared to the sending node), it starts a hold-off timer. The hold-off timer T_{hold}, is a uniform random variable in [0, $2T_{hold}^{mean}$], where T_{hold}^{mean} is given by,

$$T_{hold}^{mean} = \frac{d_{id}}{d_{sd}} \cdot \tau + \frac{\phi_{si}}{c}, \tag{3.1}$$

$$\phi_{si} = \begin{cases} R_{max} - d_{si} & \text{if } R_{max} \geq d_{si} \\ 0 & \text{if } R_{max} < d_{si}, \end{cases} \tag{3.2}$$

where d_{si} is the distance between sender s and node i, d_{id} is the distance between node i and destination d, d_{sd} is the distance between sender and the destination, τ is a constant parameter whose optimum value is to be determined, c is the speed of the underwater acoustic signal [31], R_{max} is the maximum transmission range, which is taken as a constant parameter for simulation purposes.

During the hold-off period, if the node overhears a packet, it stops its timer and becomes a non-forwarding node. If the node does not overhear any packet transmission before the hold-off timer expires, it decides to be a forwarding node. On expiration of the hold-off timer, the forwarding node transmits the packet and starts a timer $T_{timeout}$ given by,

$$T_{timeout} = T_{hold}^{max} + \frac{R_{max}}{c} + T_t^D, \tag{3.3}$$

where T_t^D is the time required to transmit a data packet, T_{hold}^{max} is the maximum value of hold-off time taken as $T_{hold}^{max} = 1.8T_h^{mean}$. A node that is closest to the sender ($d_{id} \approx d_{sd}$) will have the maximum hold-off time with T_h^{mean} as its mean value is given by,

$$T_h^{mean} = T_{hold}^{mean}|_{d_{si}=0} = \tau + \frac{R_{max}}{c} \tag{3.4}$$

Since (3.1) already takes into account the delay to reach the node located at the maximum distance, T_{hold}^{max} includes the maximum delay that it takes for the transmission of a node to be overheard by the sender.

During the timeout period, the node stops the $T_{timeout}$ timer if it overhears the packet from a node that is closer to the destination than itself. Overhearing ensures that the packet was received successfully and has been propagated further. The packet is retransmitted if the node does not overhear before $T_{timeout}$ expires. A forwarding node starts an ACK hold-off timer, $T_{ACK-hold-off}$, which is uniformly distributed in $[\frac{R_{max}}{c}, 2\frac{R_{max}}{c}]$, when it receives the data packet from the same source, i.e., a duplicate packet. As ACK-hold-off timer is used to de-synchronize the transmission of ACK and data packets, uniform distribution is used since it gives the highest deviation. On expiration of ACK-hold-off timer, the forwarding node sends an explicit ACK.

The mean value of the hold-off timer, T_{hold}^{mean} is chosen such that it de-synchronizes a node's transmission from its neighbors transmissions and avoids collisions at the receiver. The factor $\frac{d_{id}}{d_{sd}} \cdot \tau$ de-synchronizes the transmission of the nodes. Closer the node to the destination, smaller is the value of $\frac{d_{id}}{d_{sd}}$. Consequently, smaller is its hold-off timer. The factor ϕ_{si}/c in 3.1 represents an extra delay that a node should wait to allow all the nodes to receive the packet. Thus, it gives *fairness*, by providing synchronization in starting the hold-off timers of all the nodes that receive the data packet. There can arise a condition where two nodes i and j have equal distances from the sender and the destination, i.e., $d_{id} = d_{jd}$ and $d_{si} = d_{sj}$. Such nodes have equal values of T_{hold}^{mean}, which motivates the need of selecting random hold-off timers in order to de-synchronize the transmissions from nodes i and j. Greater the value of τ, greater is the value of T_{hold}^{mean}. Thus, the selection of the constant parameter, τ, in 3.1 involves a trade-off as explained in the following conditions:

1. τ should be *large enough* to de-synchronize the transmissions of nodes close to each other and avoid collision of their transmissions at the receiver,

2. τ should be *small enough* in order to retain the priorities of nodes i and j based on their positions from the source and the destination and still de-synchronize their transmissions.

If the difference in the instants at which nodes i and j transmit is greater than the transmission time of data packet T_t^D, collision can be avoided at the receiver. In other words, if the probability of difference in the hold-off timers being less than T_t^D is kept very low, collisions can be reduced to a great extent at the receiver. Assuming uniform propagation delays for nodes i and j gives rise to the following condition at the transmitters:

$$Pr(|T_{hold}^i - T_{hold}^j| \leq T_t^D) \leq \gamma, \tag{3.5}$$

where T_{hold}^i and T_{hold}^j are the hold-off times of i and j respectively, γ is taken as the probability of collision. The optimum value of τ can be derived as follows:

Let $|T_{hold}^i - T_{hold}^j|$ be $\triangle T_{hold}^+$. The pdf of $\triangle T_{hold}^+$ is given as,

$$P(\triangle T_{hold}^+) = \begin{cases} \frac{1}{T_{hold}^{mean}} - \frac{\triangle T_{hold}^+}{2T_{hold}^{2mean}} & \text{if } 0 \leq (\triangle T_{hold}^+) \leq 2T_{hold}^{mean} \\ 0 & \text{if } \triangle T_{hold}^+ > 2T_{hold}^{mean}, \end{cases} \tag{3.6}$$

which can be expressed as,

$$Pr(\triangle T_{hold}^+ \leq T_t^D) = \int_0^{T_t^D} \left(\frac{1}{T_{hold}^{mean}} - \frac{\triangle T_{hold}^+}{2 \cdot T_{hold}^{2mean}} \right) d\triangle T_{hold}^+ \leq \gamma \tag{3.7}$$

$$= \frac{T_t^D}{T_{hold}^{mean}} - \frac{T_t^{D^2}}{4 \cdot T_{hold}^{2mean}} \leq \gamma \tag{3.8}$$

Solving (3.8) and using the fact that $T_t^D \leq 2T_{hold}^{mean}$ we get,

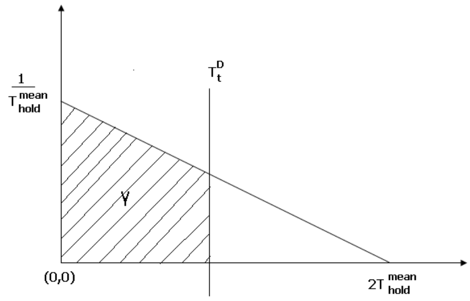

Figure 3.1: Probability Density Function of $\triangle T_{hold}^{+}$

$$T_{hold}^{mean} \geq \frac{T_t^D}{2\gamma}(1+\sqrt{1-\gamma}) = \Psi. \tag{3.9}$$

From (3.1) and (3.9), an optimum value of τ is found out by formulating an optimization problem as follows:

$\mathbf{P_{desync}^{opt}}$: **De-synchronization Optimization Problem**

$$\begin{aligned} \textbf{Given}: &\quad d_{si} > 0,\, d_{id} > 0,\, d_{sd} > 0 \\ \textbf{Find}: &\quad \tau^{*} \\ \textbf{Minimize}: &\quad \tau = \frac{d_{sd}\cdot(\Psi + \frac{d_{si}}{c} - \frac{R_{max}}{c})}{d_{id}} \\ \textbf{Subject to}: & \end{aligned} \tag{3.10}$$

$$d_{id} \leq d_{sd}; \tag{3.11}$$

$$d_{si} \leq d_{sd} + d_{id}. \tag{3.12}$$

The value of d_{sd} is taken as a constant parameter. Since hold-off timers are always started by the nodes closer to the destination than the sender, $d_{id} \leq d_{sd}$. Also, by considering nodes s, i and d as vertices of triangle, by triangle inequality, we have $d_{si} \leq (d_{sd} + d_{id})$.For different values of d_{sd} and γ, the values of τ obtained are shown in Table 3.1. To have a better understanding of the algorithm, we present a finite state machine (FSM) for nodes in the unicasting phase as shown in fig. 3.2. The change in the state of a node is represented by $\frac{event}{action}$. A node can be in either of the four states: 'idle', 'wait', 'transmitted packet' or 'retransmitted packet' state. In 'idle' state, a node does not hold any packet and none of its timers are active. In other words, a node is in

Table 3.1: Parameter τ in seconds

	$d_{sd} = 0.1$Km	$d_{sd} = 1$Km	$d_{sd} = 10$Km
$\gamma = 0.02$	4.1740	4.1700	4.200
$\gamma = 0.05$	1.5746	1.1746	1.000
$\gamma = 0.1$	0.9800	0.8130	0.8000

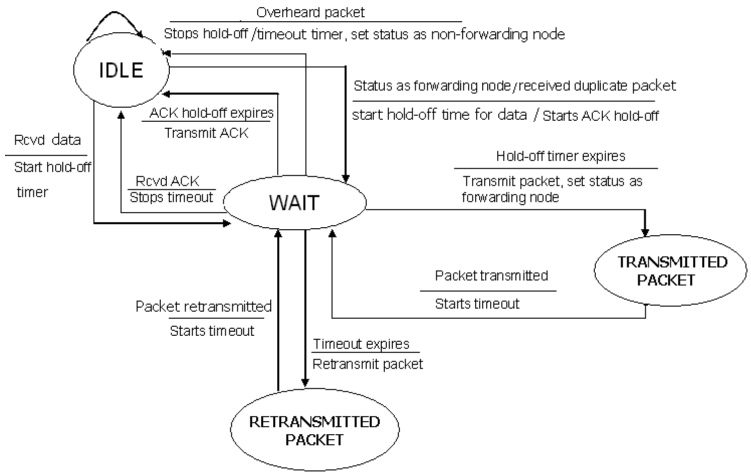

Figure 3.2: FSM for a node with no neighbor knowledge for unicasting phase

'idle' state either before reception of any packet, after successful transmission of received data packet, or transmission of an ACK packet after duplicate reception from the same sender. A node is in 'wait' state in either of the cases: (i) its hold-off timer for data packet is active, (ii) its timeout timer is active, (iii) its hold-off timer for ACK packet is active. On the expiry of hold-off timer, a node enters 'transmitted packet' state and starts a timeout timer. On the expiry of timeout timer, a node enters 'retransmitted packet' state and resets the timeout timer. On the expiry of ACK hold-off timer, a node enters 'idle' state.

3.0.2 One-Hop Neighbor Knowledge

Similar to no neighbor knowledge based protocol, a node with one-hop neighborhood knowledge ensures reliability by requiring each node receiving the data packet to send an implicit ACK (overhearing) or an explicit ACK. However, unlike the previous case, *limited flooding* scheme is not employed in this case. The nodes do not decide for themselves whether to set the status as forwarding or non-forwarding. Rather, the sender of the data packet designates the one-hop neighbor closest to the destination as the next forwarding node. Thus the sender of the data packet inserts the ID of the designated node in one of the fields of the packet and transmits it. When any node receives this packet, it checks for this field and discards the packet if it is not meant for it. Thus, the routing decisions are based on neighbor knowledge, which reduces the number of redundant transmissions as compared to no neighbor knowledge based protocol, since only the designated nodes transmit in this case. The MAC scheme is designed by making use of one-hop neighbor knowledge, in order to reduce collisions of the transmissions at a node from the neighboring nodes. This, in turn, reduces the number of retransmissions of a data packet. Thus the MAC scheme is designed as follows:

Consider the case where three nodes k-1, k and k+1 select k, k+1 and k+2 as their next best hops respectively, as shown in fig 3.3. When node k-1 transmits the data packet to node k, it starts a $T_{timeout}^{k-1}$ timer for retransmission of packet. During the timeout period, it waits to overhear the transmission of node k. The value for this timeout period is deterministic and is given by:

$$T_{timeout}^{k-1} = T_t^D + 2T_p^{k-1,k} + T_{hold}^{max} \tag{3.13}$$

where $T_p^{k-1,k}$ is the propagation delay from node k-1 to k, T_{hold}^{max} is the maximum value of the holding time, which is varied in [0, $2T_t^D$], T_t^D is the time required to transmit the data packet.

On receiving the data packet successfully and with the knowledge of its one-hop neighbors, node k knows the time at which it would receive the retransmission (if any) from node k-1. It adjusts the value of hold-off timer T_{hold}^k in [0, T_{hold}^{max}] in order to transmit the packet to its next-hop neighbor k+1. The value of T_{hold}^k is chosen such that there is no collision of retransmission from node k-1 and overhearing from k+1 at node k. The expressions for the estimated reception times of retransmission and overhearing at node k are given by:

$$T_{k-1} = 2T_t^D + 2T_p^{k-1,k} + T_{hold}^{max} \tag{3.14}$$

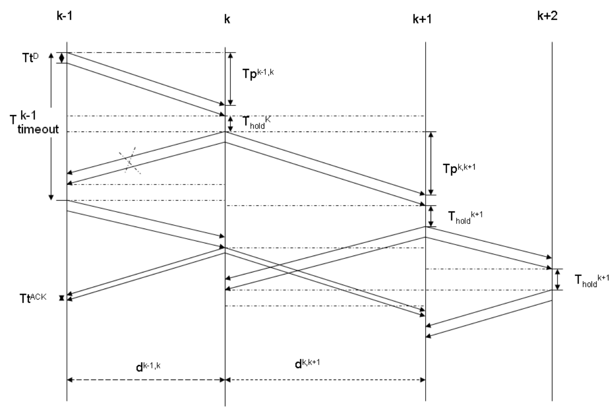

Figure 3.3: Timing diagram for one-hop neighborhood knowledge based protocol

$$T_{k+1} = 2T_t^D + 2T_p^{k,k+1} + T_{hold}^k + T_{hold}^{max} \tag{3.15}$$

where T_{k-1} and T_{k+1} are the estimated times of reception of the retransmission from node k-1 at node k and overhearing of packet transmitted by node k+1 at node k respectively.

From (3.14) and (3.15), it can be seen that the node k leverages the distances from nodes k-1 and k+1 to determine the hold-off time of data packet and avoid collisions. On expiration of hold-off timer, node k transmits the data packet. If node k-1 does not overhear the transmission of node k before timeout, it retransmits. Node k sends an explicit ACK on hearing each successive retransmission. Transmission of explicit ACK by k is scheduled by taking the time instants T_{k-1} and T_{k+1} into consideration. There are three cases:

1. $T_{k+1} > T_{k-1}$ and $(T_{k+1} - T_{k-1}) > (T_t^D + T_t^{ACK})$.

   In this case, k transmits ACK immediately after the reception of retransmission from k-1, i.e., at time $(T_{k-1} + T_t^D)$

2. $T_{k+1} > T_{k-1}$, $(T_{k+1} - T_{k-1}) < (T_t^D + T_t^{ACK})$ and k overhears by time T_{k+1}.

   k schedules the transmission of ACK after the completion of overhearing from k+1, i.e., at $(T_{k+1} + T_t^D)$

3. $T_{k+1} > T_{k-1}$, $(T_{k+1} - T_{k-1}) < (T_t^D + T_t^{ACK})$ and k does not overhear by time T_{k+1}. Since k has to retransmit the data packet in this case, it does not transmit the ACK. Retransmission of data packet provides implicit ACK to k-1.

Thus the hold-off time for ACK, which is started at the reception of retransmission is determined such that the transmission of ACK is de-synchronized from the transmission of data packet so as to avoid collisions between the ACK and data packets, thereby reducing the number of retransmissions. Accordingly the hold-off timer for ACK $T_{ACK-hold-off}$, which is started at the reception of retransmission, is determined based on the algorithm 1.

Algorithm 1 Selection of $T_{ACK-hold-off}$ for one-hop neighbor knowledge

```
if T_{k+1} > T_{k-1} then
  if (T_{k+1} - T_{k-1}) < (T_t^D + T_t^{ACK}) then
    if k overhears from k+1 then
      T_{ACK-hold-off} = T_{k+1} - T_{k-1}
    else
      discard ACK and retransmit the packet after timeout
    end if
  else
    T_{ACK-hold-off} = T_t^D
  end if
else
  T_{ACK-hold-off} = T_t^D
end if
```

FSM for a node with one-hop neighbor knowledge is shown in fig 3.4. Similar to the protocol with no neighbor knowledge, the four states of a node in this case are: 'idle', 'wait', 'transmitted packet' and 'retransmitted packet' state. In 'idle' state, a node does not have any active timer and it does not hold any packet. A node is in 'wait' state if its hold-off timer for data packet, timeout timer, or hold-off timer for ACK packet is active. On the expiry of hold-off timer, a node enters 'transmitted packet' state and starts a timeout timer. On the expiry of timeout timer, a node enters 'retransmitted packet' state and resets the timeout timer. Consequently, a node enters the 'wait' after starting/resetting the timeout timer.

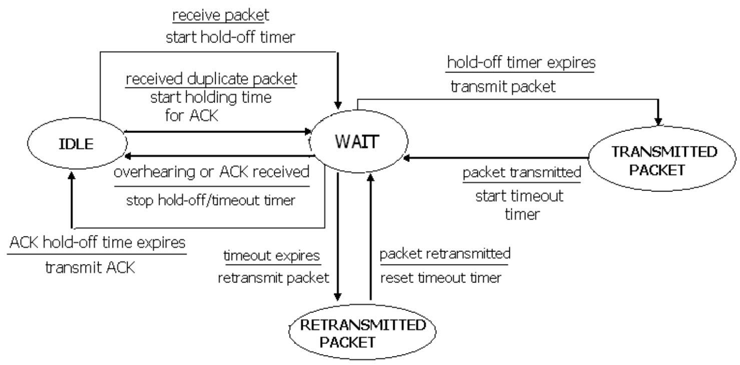

Figure 3.4: FSM for a node with one-hop neighbor knowledge

3.0.3 Two-Hop Neighbor Knowledge

Two-hop neighbor knowledge based protocol differs from the previous versions of the protocol in the following ways: (i) the receiver of the data packet does not have the freedom to choose the next forwarding hop based on its own neighbor knowledge. It is designated by the sender of the data packet. This is done to make optimum routing decisions as explained later. (ii) Owing to the two-hop neighborhood information, the MAC scheme is designed to avoid collisions at the first hop and second hop neighbors. Consequently, transmissions can be scheduled deterministically with the help of first and second hop distances. Thus unlike no neighbor knowledge and one-hop neighbor knowledge based protocols, this protocol makes it possible to avoid collisions at second hop neighbor also. Hence for routing purposes, the sender of the data packet inserts the ID's of the next two forwarding hops in the packet. The MAC scheme for the two-hop neighbor knowledge based protocol is designed as explained below.

The knowledge of next two forwarding hops helps a node to schedule its retransmissions (when it does not receive implicit or explicit ACK from the earlier transmission), so as to avoid collision at the next two hops. To achieve this, we use different schemes for node (source) originating the data and for the subsequent forwarding nodes in the unicast chain. The source say s, will first choose its second hop neighbor j, that is closest to the destination and then select the best next hop i as the one via which j is the next hop. The source will then include the selected first and second hops (i and j

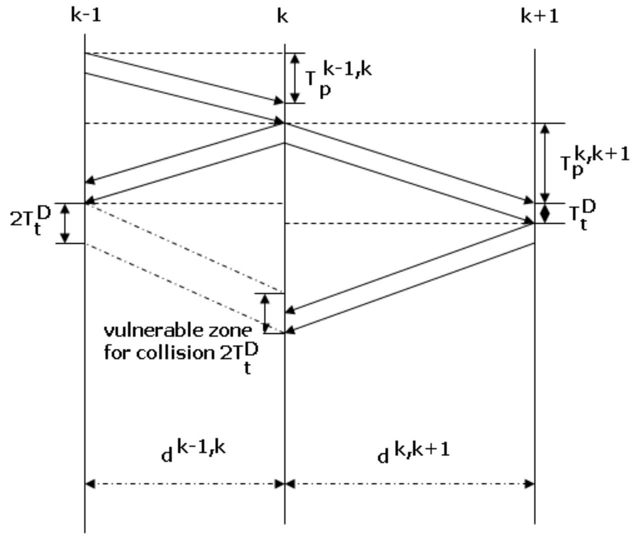

Figure 3.5: Collision due to retransmission, at one-hop neighbor

respectively) in the data packet and transmit the data packet. The same scheme will, however, not be followed by the next hop i. The node i will not select the best next hop but will rather take j as its next hop. Nevertheless, it will select the best second hop k that is reachable via j. After this, i will include the designated first and second hops (j and k respectively) in the data packet and transmit the data packet. Thus, the source gets to choose the next two hops, whereas, subsequent forwarding nodes only choose the next second hop.

Consider nodes k-1, k, k+1. As shown in the fig 3.5, node k-1 sends a packet to node k. After receiving the data packet, node k transmits it immediately. Node k-1 starts a timer $T_{timeout}$, given by,

$$T_{timeout} = 2T_p^{k-1,k} + T_t^D, \tag{3.16}$$

where $T_p^{k-1,k}$ is the propagation delay between node k-1 and k, T_t^D is the data packet transmission time.

The timer is stopped if either node k-1 is able to overhear node k's transmission of the data packet or if node k-1 receives an explicit ACK from node k. The timeout expiry implies two possible cases.

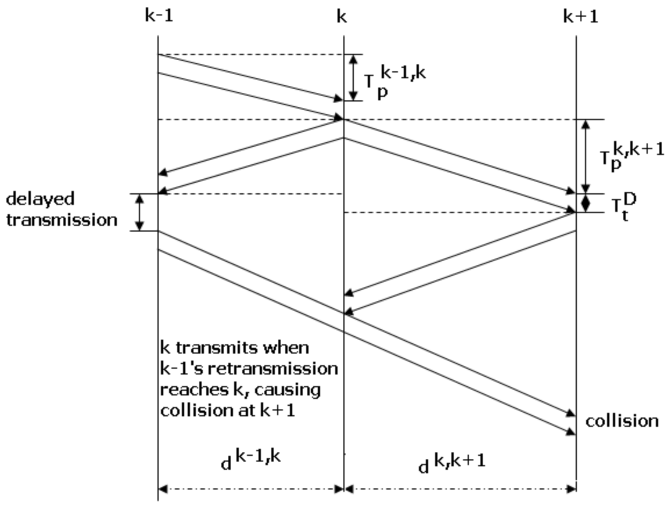

Figure 3.6: Collision due to retransmission, at two-hop neighbor

1. Data packet transmission of node k-1 to node k got lost.

2. Node k-1 was not able to overhear node k's transmission or receive an explicit ACK from node k.

In case 1, node k-1 needs to retransmit immediately. On the other hand, in case 2, node k-1 needs to time its retransmission so that the retransmitted packet does not collide with the overhearing of data packet transmitted by k+1 at k . The knowledge of next two forwarding hops, i.e. k and k+1, can be used to accurately time the retransmission at k-1 to avoid collision at k. If $(2T_p^{k,k+1} - T_t^D < 2T_p^{k-1,k} < 2T_p^{k+1,k} + T_t^D)$, then k-1's retransmission will collide with the overhearing of the packet transmitted by k+1, at k. Thus, collision can be avoided if k-1 delays its retransmission by $(2(T_p^{k,k+1} - T_p^{k-1,k}) + T_t^D)$ to reach node k exactly after it finishes its overhearing from node k+1. If the above inequality does not hold, then there are no chances of collision and k-1 can immediately retransmit the data packet.

Consider also the scenario depicted in fig 3.6. Here even after k-1 has delayed its retransmission, collision between retransmissions of k-1 and k can occur at k+1. This collision can obstruct the reception of data packet at node k+1, thus delaying the data propagation. To avoid this kind of collision, we make, as a rule, node k always delay its retransmission by T_t^D. This means that node k-1 will also delay its retransmission by T_t^D before making the comparison defined by the inequality above. There can still be another scenario when even after k delays its retransmission, node k-1's retransmission overlaps with node k's retransmission causing collision at node k+1. In this case we would need to define some more conditions for node k-1 to time its retransmission. Combining all these scenarios with the above inequality we can formulate a rule as shown in algorithm 2. In the above algorithm, in case $(2T_p^{k,k+1} < 2T_p^{k-1,k} < 2T_p^{k,k+1} + 4T_t^D)$,

Algorithm 2 Selection of timeout period for retransmission

```
if (2T_p^{k,k+1} - 2T_t^D < 2T_p^{k-1,k} < 2T_p^{k+1,k}) then
  delay retransmission by (2T_p^{k,k+1} - 2T_p^{k-1,k})
else
  if (2T_p^{k+1,k} < 2T_p^{k-1,k} < 2T_p^{k+1,k} + 4T_t^D) then
    delay retransmission by (2T_p^{k+1,k} + 4T_t^D - 2T_p^{k-1,k})
  else
    delay retransmission by mandatory period of T_t^D
  end if
end if
```

node k-1 delays its retransmission by $(2T_p^{k,k+1} + 4T_t^D - 2T_p^{k-1,k})$. This is done to give room to node k to adjust its retransmission to avoid collision at node k+1. The overall scenario is depicted in fig 3.7. Node k-1 is not able to overhear from node k, therefore, it retransmits the data packet after waiting for T_t^D amount of time. Node k determines that if it transmits at its scheduled time (T_t^D after timeout) it will interfere with node k+1's retransmission at node k+2. Therefore, it delays its retransmission by $4T_t^D$ amount of time. Node k+1 determines that if it transmits at its scheduled time it will interfere with the overhearing at node k+2. Therefore it delay its retransmission by T_t^D amount of time.

Now, consider that node k receives a duplicate packet from node k-1. We can make four cases:

1. Duplicate packet arrives at k more than $T_t^D + T_t^{ACK}$ time units before k's timeout. In this case node k immediately sends an ACK.

2. Duplicate packet arrives less than $T_t^D + T_t^{ACK}$ before k's timeout. In this case k will hold the ACK and will wait for the overhearing from k+1. If the data packet

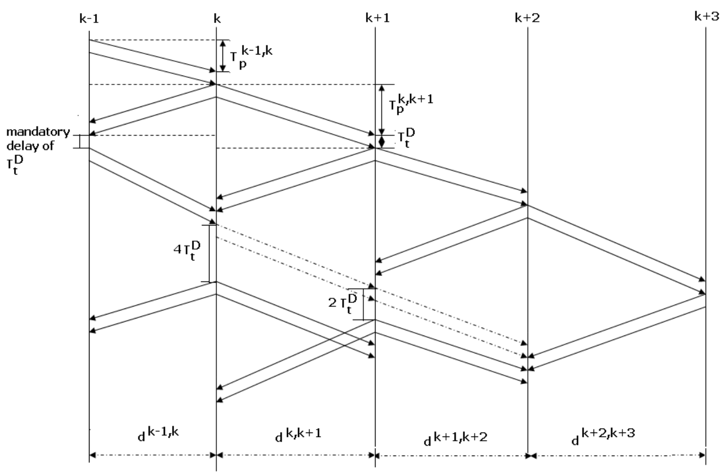

Figure 3.7: Timing diagram for two-hop neighborhood knowledge based protocol

is overheard from k+1, an ACK is transmitted immediately after overhearing. If the data packet is not overheard from k+1, only the data packet is retransmitted.

3. Duplicate packet arrives less than $2T_p^{k-1,k}$ after timeout and subsequent retransmission. In this case k will not take any action as this duplicate packet was transmitted by k-1 before it could have received k's retransmission.

4. Duplicate packet arrives more than $2T_p^{k-1,k}$ after timeout and subsequent retransmission and also more than $T_t^D + T_t^{ACK}$ time units before k's next timeout. In this case node k immediately sends an ACK.

Let T_{k-1} be the time at which the duplicate packet arrives at k, T_{k+1} be the time when k expects to overhear the transmission by k+1 and T_k be the retransmission time of k. The following algorithm determines the hold-off time for ACK corresponding to the reception of the duplicate packet.

The FSM for a node with two-hop neighbor knowledge is shown in fig 3.8. Similar to no neighbor knowledge and one-hop neighbor knowledge based protocols, a node with two-hop neighbor knowledge enters four states. However, unlike the previous cases, a node after reception of packet for the first time does not go into the 'wait' state. It immediately transmits the received packet and goes into 'transmitted packet' state since there is no hold-off timer used for data packet transmission.

Algorithm 3 Algorithm for selecting $T_{ACK-hold-off}$ for two-hop neighbor knowledge

if $(T_{k+1} - T_{k-1} \geq 2T_t^D)$ **then**
  send ACK immedialtely.
else
  if $(0 < T_{k+1} - T_{k-1} < 2T_t^D)$ **then**
    hold ACK for $T_{k+1} - T_{k-1}$
    if data packet overheard at T_{k+1} **then**
      send ACK immediately
    else
      retransmit the packet after T_t^D
    end if
  else
    if $(T_{k-1} - T_k < 2T_p^{k-1,k})$ **then**
      do nothing
    else
      if $(2T_p^{k-1,k} < T_{k-1} - T_k < T_t^D + T_t^{ACK})$ **then**
        send ACK immediately
      end if
    end if
  end if
end if

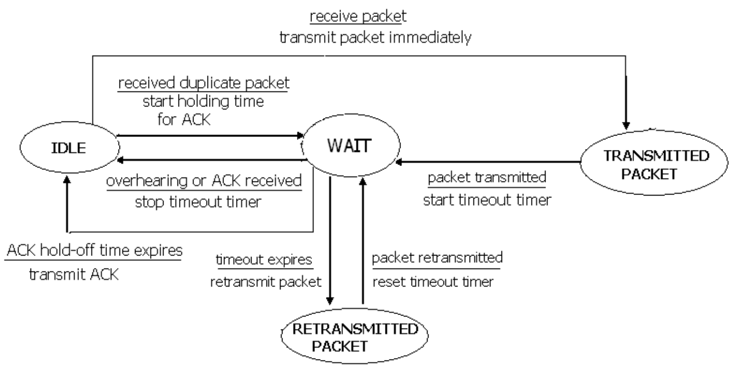

Figure 3.8: FSM for a node with two-hop neighbor knowledge

Chapter 4

Reliable Geocast Protocol for Underwater Inter-Vehicle Communications

Geocast protocols are used for disseminating the query inside a region, which is defined by the sender, i.e, the surface station. In unicasting, the nodes choose a fixed destination, i.e, the surface station. Geocasting does not have a fixed destination. The idea is to reach all the nodes in the geocast region. In this sense, geocast is decoupled from unicast. In geocasting protocol, we leverage the shape of the geocast region using the information included in the transmitted query packet. Assuming the geocast region to be a cylindrical, the surface station includes the following information about the geocast region in the query packet:

- The geographical co-ordinates of the center 'C' of the geocast region,
- Vector $\vec{v}$ along the longest side of the geocast region and passing through the center,
- Radius vector $\vec{r}$ of the cylindrical region.

If $\vec{si}$ is the vector from the sender s to the receiver i of the query packet, the projection of the vector $\vec{si}$ along vectors $\vec{v}$ and $\vec{r}$ are given by,

$$d_{si_v} = \vec{si} \cdot \frac{\vec{v}}{\|\vec{v}\|} \tag{4.1}$$

$$d_{si_r} = \vec{si} \cdot \frac{\vec{r}}{\|\vec{r}\|} \tag{4.2}$$

A node receiving the query packet determines if it lies in the geocast region, with the help of the above information transmitted in the query packet, and knowledge of its own geographical location. Every node lying inside the geocast region and receiving the packet follows the protocol for query dissemination inside the region. If a node lying outside the geocast region receives a packet, it discards it.

4.0.4 No Neighbor Knowledge

Consider the shape of geocast region as shown in the figure 4.1. Let's say the query packet enters the geocast region through the node A. Node A then broadcasts the query packet. Immediately after broadcasting the query packet, node A also transmits a short packet called ENFORCE packet. The ENFORCE packet is sent to cater for the nodes that might not have received the query packet in the first attempt but are able to receive the ENFORCE packet. On receiving the query packet for the first time, the nodes E, D, C and B start a hold-off timer, T_{hold}. T_{hold} is a uniform random variable in in $[0, 2T_{hold}^{mean}]$ where T_{hold}^{mean} is given by,

$$T_{hold}^{mean} = (1 - \frac{d_{si_v} - d_{si_r}}{R_{max}}) \cdot \tau + \frac{\phi_{si}}{c}, \tag{4.3}$$

$$\phi_{si} = \begin{cases} R_{max} - d_{si} & \text{if } R_{max} \geq d_{si} \\ 0 & \text{if } R_{max} < d_{si}, \end{cases} \tag{4.4}$$

where d_{si_v} and d_{si_r} are the projection of the vector $\vec{si}$ along the vector $\vec{v}$, and along the radius vector $\vec{r}$ passing through s respectively. The idea is to give higher priority to nodes having a greater value of d_{si_v} and smaller value of d_{si_r} so that the query packet quickly penetrates the length of the geocast region. From the figure 4.1 one can see that node C will have the smallest hold-off timer as compared to other receiving nodes. Once node C's hold-off timer expires, it broadcasts the packet. If node E, D or B is able to overhear C's transmission during the hold-off period, it will stop the timer with probability P(NF)= $1 - \frac{d_{si_v} - d_{si_r}}{R_{max}}$. Once a node overhears during hold-off period, it statistically decides to be a forwarding node or a non-forwarding node. A non-forwarding node simply stops its hold-off timer whereas forwarding node broadcasts the packet once the hold-off timer expires. However, if the node does not overhear during hold-off period, it becomes a forwarding node.

A node that does not receive the query packet but receives the ENFORCE packet decides to inform other nodes that it did not receive the query packet by sending a

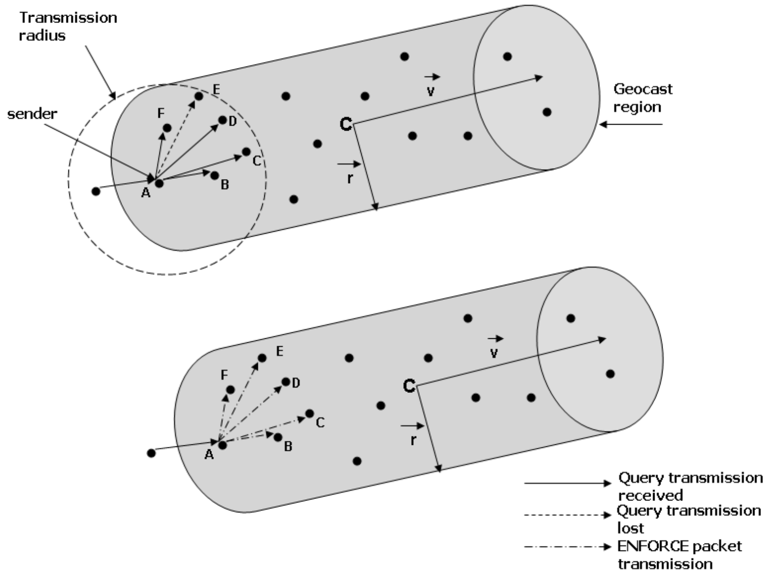

Figure 4.1: Broadcast in the geocast region

NACK. Before transmitting a NACK, the node waits for a duration of NACK-hold-off timer given by,

$$T_{NACK-hold-off} = T_{hold}^{max} + \frac{R_{max}}{c} + T_t^Q. \tag{4.5}$$

In the figure 4.2, it can be seen that node F will transmit a NACK. The NACK hold-off timer ensures that F waits long enough to overhear the transmission from a forwarding node in the neighborhood if any. If F is not able to overhear before NACK-hold off timer expires, it will transmit the NACK and start a NACK-timeout timer given by,

$$T_{NACK-timeout} = 2\frac{R_{max}}{c} + T_t^Q. \tag{4.6}$$

A node that receives the NACK will respond with probability P(N)= $\frac{n}{n+2}$ where n is the number of NACK's received. A node that receives the highest number of NACKs will have a higher probability to respond. If a node does not get the packet during NACK timeout period, it will retransmit the NACK.

The FSM for a node with no neighbor knowledge is as shown in Fig. 4.3. A node has five states: 'idle', 'wait', 'transmitted query', 'transmitted ENFORCE' and 'NACK received' state. On receiving a NACK, a node enters 'NACK received' state and goes back to 'idle' state if it statistically decides not to transmit the query. A node is in 'wait'

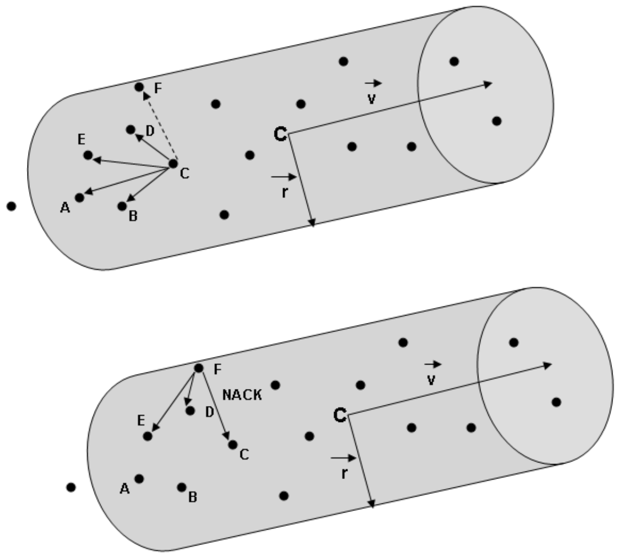

Figure 4.2: NACK transmission in the geocast region

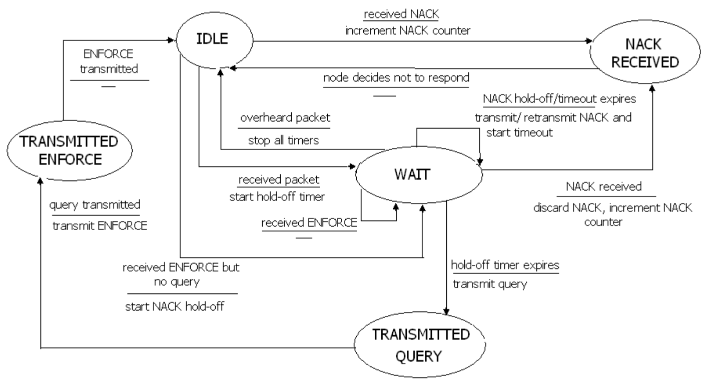

Figure 4.3: FSM for a node inside the geocast region with no neighbor knowledge

state when either of its timers are active. When hold-off timer for query packet expires, a node enters ‘transmitted query’ state, transmits query and then enters ‘transmitted ENFORCE’ state. Consequently, it transmits the ENFORCE packet and enters an ‘idle’ state.

4.0.5 One-Hop Neighbor Knowledge

Unlike zero-hop case, a node with one-hop neighborhood knowledge ensures reliability by requiring each neighboring node receiving the query to send an implicit ACK (overhearing) or explicit ACK. A node after receiving the query packet starts a hold-off timer. The expression for hold-off timer of a node receiving the query packet, which is uniformly distributed in $[0,\ 2T_{hold}^{mean}]$, is same as that of zero-hop case, given by,

$$T_{hold}^{mean} = (1 - \frac{d_{si_v} - d_{si_r}}{R_{max}}) \cdot \tau + \frac{\phi_{si}}{c}, \tag{4.7}$$

$$\phi_{si} = \begin{cases} R_{max} - d_{si} & \text{if } R_{max} \geq d_{si} \\ 0 & \text{if } R_{max} < d_{si}, \end{cases} \tag{4.8}$$

The neighbors (nodes inside the geocast region only) of the sender can be categorized as follows:

- Forwarding nodes (nodes which decide to broadcast the received packet).
- Non-forwarding nodes (nodes which decide not to broadcast the received packet).
- Nodes which fail to receive the transmission.

The transmission patterns are considered as spheres of radius R_{max} in underwater 3D environment. During the hold-off period, a node might overhear the transmission from the other nodes. As a node knows the co-ordinates of its neighbors, it determines whether any of its neighbors are covered by the overhearing. This is done by computing the distance d between each of its neighbors and the sender and determining if $d < R_{max}$. If all its one-hop neighbors satisfy this condition, it stops the timer, discards the packet and becomes a non-forwarding node. On the other hand, even if one neighbor does not satisfy $d < R_{max}$, the node becomes a forwarding node and decides to transmit after the hold-off timer expires. A non-forwarding node sends an explicit ACK to the sender after waiting for an ACK-hold-off period, $T_{ACK-hold-off}$. Explicit ACK is also sent by forwarding nodes on every duplicate reception of a packet after waiting for $T_{ACK-hold-off}$. To de-synchronize explicit ACKs with the query traffic, $T_{ACK-hold-off}$ is uniformly distributed in $[\frac{R_{max}}{c},\ 2\frac{R_{max}}{c}]$. The sender will keep track of all the ACKs and overhearings it receives, and will retransmit the packet if there is even a single neighbor that does not reply implicitly or explicitly.

The timeout timer has been designed so that a node waits long enough to hear from all its neighbors before it retransmits. Thus,

$$T_{timeout} = T_{hold}^{max} + \frac{R_{max}}{c} + T_t^Q \tag{4.9}$$

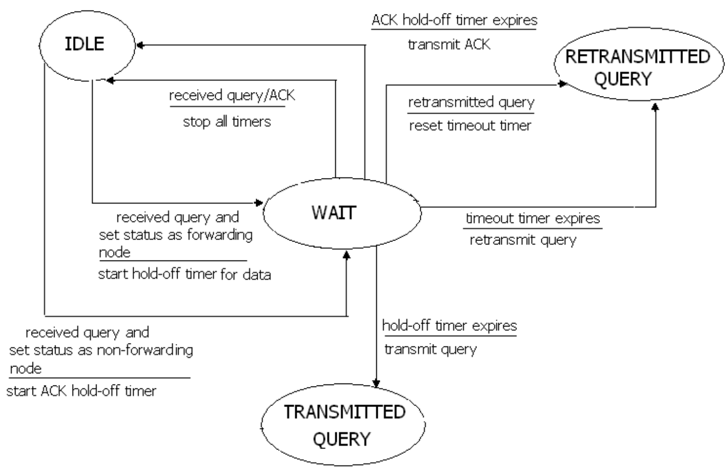

Figure 4.4: FSM for a node inside the geocast region with one-hop neighbor knowledge

Fig. 4.4 shows the FSM for a node with one-hop neighbor knowledge inside the geocast region. The four states of the node are: 'idle', 'wait', 'transmitted query' and 'retransmitted query' state. Similar to the previous case, a node is in 'wait' when either its hold-off timer or timeout timer is active. When a node transmits query for the first time, it enters 'transmitted' query state. With any subsequent retransmission of query, it enters 'retransmitted query' state. In 'idle' state, none of its timers are active.

4.0.6 Two-Hop Neighbor Knowledge

Similar to one-hop neighbor knowledge, this version of the protocol ensures reliability with the help of implicit and explicit ACKs. However, the two-hop neighbor knowledge enables the sender to schedule the transmission of all its neighbors. When the query reaches the last node outside the geocast region, the border node, the broadcast process starts. Let the degree, D, of a node be defined as the number of neighbors of a node that are two-hop neighbors of the sender, s (border node in this case). Let d_{si_v} and d_{si_r} be the projections of vector $\vec{si}$ along the vector $\vec{v}$, and along the radius vector $\vec{r}$ passing through s respectively. Let R_{max} be the maximum transmission range of the sender s. We define a priority metric, $D + \frac{d_{si_v} - d_{si_r}}{R_{max}}$, for each node. Greater the value of the priority metric, higher is the priority of a node. The procedure to select forwarding nodes is described below:

1. The border node arranges its next hop neighbors in descending order in terms of their priority metric.

2. Node with the highest metric is chosen as a forwarding node.

3. After choosing a node as the forwarding node, degree of the remaining nodes is reduced according to the number of 2-hop neighbors of the sender s already covered by the chosen forwarding nodes. The priority of remaining nodes is recalculated. These remaining nodes are again arranged in descending order w.r.t their revised priority.

4. Steps 2-3 are repeated until all the two hop neighbors of the sender have been covered.

The nodes in the one-hop neighborhood who were not designated as forwarding nodes are given the status of non-forwarding nodes. After this, the border node schedules query transmission and ACK transmission for forwarding nodes and non-forwarding nodes respectively, and transmits the schedule in the query packet. The schedule is required so that all the nodes do not transmit the query packets or ACKs simultaneously and cause collisions at the intended receivers. Since the query packet transmission time is T_t^Q, collision between packets can be avoided if the time difference between reception of two packets at any receiver is at least T_t^Q. Therefore, the forwarding nodes are directed to delay their transmission in terms of some integral multiple, say I, of T_t^Q to avoid collisions at all receivers. For the forwarding node we consider T_t^Q to be an appropriate minimum time resolution in schedule calculation. Similarly, non-forwarding nodes are directed to delay their ACK transmission in terms of some integral multiple of T_t^{ACK} to avoid collisions at the sender.

To decide upon a schedule, the sender s divides all of the forwarding nodes, A, B and C in figure 4.5, into two disjoint sets, the adjacent node set and the non-adjacent node set. Both the sets, collectively called as node sets, are initialized as empty sets at the beginning of the scheduling. Two nodes are defined to be non-adjacent nodes, if their common neighbor nodes only consist of either the sender or it's one-hop neighbors. In figure 4.5, nodes B and C and nodes A and C are non-adjacent nodes as all of their

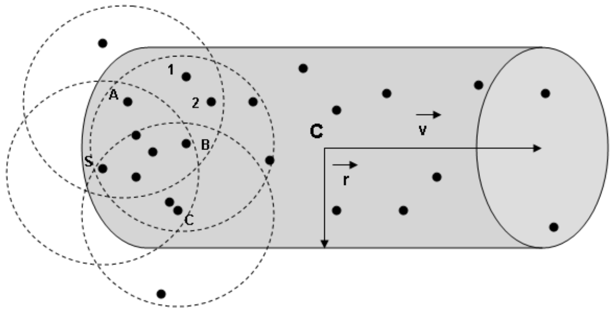

Figure 4.5: Adjacent and non-adjacent nodes

common one-hop neighbors are also the one-hop neighbors of the sender S. Two nodes are defined as adjacent if they have some common neighbors that are two-hop neighbors of the sender s. In Fig. 4.5 nodes A and B are adjacent nodes.

When non-adjacent nodes transmit without a schedule, collisions may occur at the sender and at the common neighbors that are also one-hop neighbors of the sender. Collisions occurring at one-hop neighbors of the sender are not of interest as these neighbor nodes are expected to have received the query packet from sender s before they receive query packet from the forwarding nodes. In case they have not received the packet, it is the responsibility of the sender s to make sure that they receive the packet. Thus, a schedule is required only to de-synchronize receptions at the sender. When adjacent nodes transmit without a schedule, collision can be caused at the sender as well as their common neighbors. As mentioned earlier, we are not interested in collisions occurring at common neighbors that are also one-hop neighbors of the sender s. Therefore, a schedule is required to de-synchronize receptions at the sender and at those common neighbors that are one-hop neighbors of the sender.

Scheduling of forwarding nodes: All the forwarding nodes are first arranged in descending order w.r.t. to the metric $d_{si_v} - d_{si_r}$. A node with the maximum value of this metric is scheduled to transmit immediately after the reception of the query packet, i.e. $I = 0$ and is added as the first node of non-adjacent node set. Note that the idea is to give higher priority to the nodes along the length of the geocast region, and lower priority to the nodes towards the edges. Then going down the list of forwarding nodes, a node say N, that is non-adjacent to all the nodes in the non-adjacent node set is initially given a transmission schedule with $I = 0$. It is determined whether collision will occur at the sender should all the nodes in the non-adjacent set and the node N transmit query packets at their scheduled time. If the calculation shows collision, the node N is scheduled to delay its transmission by an integral multiple I of T_t^Q. I is the minimum integer that is required so that if node N delays its transmission by IT_t^Q collision of query packet at the sender s can be avoided. This collision avoidance is necessary since the sender needs to overhear transmissions from all the forwarding nodes to ensure reliability. After this, node N is added to the non-adjacent node set. The procedure is continued till no more nodes can be added to the non-adjacent node set. Then for each node say L, in the remaining list, nodes adjacent to L in the non-adjacent node set and the adjacent node set (which is initially empty) are determined. Node L is initially given a transmission schedule with $I = 0$. Considering that all the nodes in the non-adjacent and adjacent node set transmit at their scheduled times, collision calculation is done at sender s and at those common neighbors of L and its adjacent nodes that are two-hop neighbors of the sender s. If the calculation shows collision the node L is scheduled to delay its transmission by an integral multiple I, of T_t^Q. After this, node L is added to the adjacent node set and the procedure is repeated until all the remaining nodes have been added to the adjacent node set.

To better understand the scheduling algorithm consider the arrangement given in Fig. 4.5. The ordered list of forwarding nodes is given as B, C and A according to the priority metric $d_{si_v} - d_{si_r}$ defined by us. Since node B has the highest priority, it is assigned a transmission schedule with $I = 0$. Node B becomes the first node to be added to the non-adjacent node set. After this node C is considered. Node C is non-adjacent to node B, therefore, it is first given a tentative transmission schedule with $I = 0$. Since nodes B and C are at the same distance from the sender s, they will receive the query packet at the same time, and if they transmit immediately a collision will occur at the sender who will not be able to overhear their transmissions. Therefore, node C is given a schedule with $I = 1$ so that node C delays its transmission of the query packet by T_t^Q to avoid the collision at the sender s. Node C will also be added to the non-adjacent node set. The sender s goes on to node A. Node A is adjacent to node B and there are

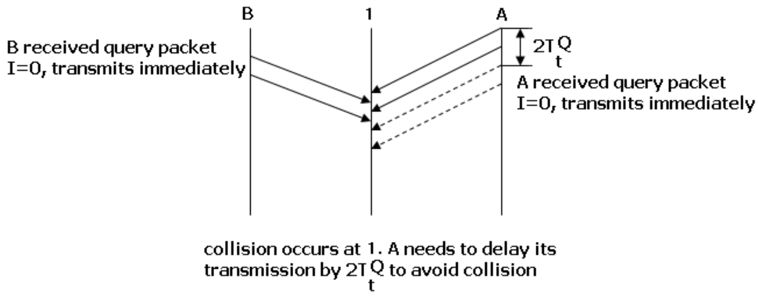

Figure 4.6: Scenario at node 1

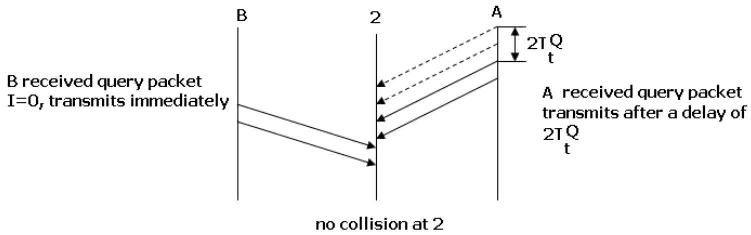

Figure 4.7: Scenario at node 2

no more nodes left that are non-adjacent to all the nodes (B and C) in the non-adjacent node set. Initially, node A is given a transmission schedule with $I = 0$. For calculation, it is considered that the nodes A, B and C transmit at their scheduled times, i.e., $I = 0$ for A, $I = 0$ for B and $I = 1$ for C. It is then calculated whether the collision will occur at the common neighbors 1 and 2 of A and B and at the sender S. Figures 4.6, 4.7, and 4.8 show that the collision occurs at node 1 due to the transmissions of A and B. If A delays its transmission by $2T_t^Q$, collision can be avoided at nodes 1, 2 and S. Thus, A is given a transmission schedule with $I = 2$ and added to the adjacent node set.

When a forwarding node inside the geocast region receives a query packet for the first time, it uses neighbor elimination [28][29] to reduce its two-hop neighborhood set to a set of nodes that have not been covered by the sender and the other forwarding nodes with higher IDs selected by the sender (information from the query packet). The

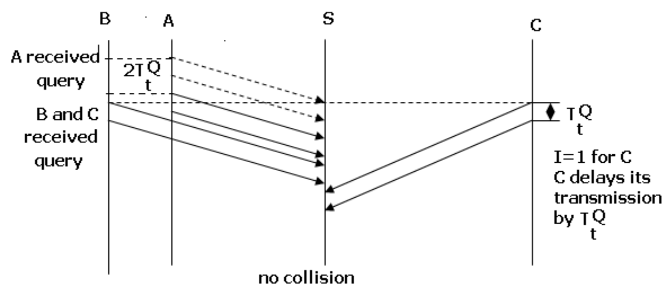

Figure 4.8: Scenario at sender s

nodes in the reduced two-hop neighborhood set are then assigned schedules using the scheduling algorithm described above.

Scheduling of non-forwarding nodes: The sender of the query packet inside the geocast region directs its non-forwarding nodes to wait for $2T_p{+}I_{max}T_t^Q + T_t^Q$ before sending an ACK, where I_{max} is the largest integral number assigned to a forwarding node during scheduling and T_p is the propagation delay required to reach this node. This is done to allow the query packet to propagate further before the non-forwarding nodes start sending their ACKs to the sender.

$2T_p{+}I_{max}T_t^Q + T_t^Q$ is called the *default holding time* for the non-forwarding nodes.

The sender arranges the non-forwarding nodes in ascending order w.r.t their distance from itself. Beginning from the closest node, the sender calculates the time at which an ACK will be received at the sender if the node transmit it after waiting only for the default holding time. For the sender s and non-forwarding node r the ACK will reach the sender s $(2T_p^{s,r} + 2T_p + T_t^Q * (I_{max}{+}1) + T_t^{ACK})$ time units after sender s broadcasts the query packet. If a node seems to be causing a collision at the sender, it is directed by sender to delay its ACK transmission by an integral multiple, I_a, of T_t^{ACK} on top of the default holding time. Transmission delay resolution is kept in multiples of T_t^{ACK}, because if two ACKs collide, the maximum collision duration is T_t^{ACK}. Hence if one ACK transmission is delayed by a T_t^{ACK} amount of time, the collision of two ACKs can be avoided.

Timers and Timeouts: After broadcasting the query packet, the sender starts two timers, timer for forwarding nodes $T^{F}_{timeout}$ and timer for non-forwarding nodes $T^{NF}_{timeout}$. The sender expects to overhear all transmissions by the forwarding nodes in $T^{F}_{timeout}$ and therefore $T^{F}_{timeout}$ is set to $2T_p{+}I_{max}T^{Q}_{t} + T^{Q}_{t}$, where I_{max} is the largest integral number assigned to a forwarding node during scheduling. If the sender did not overhear transmissions from all the forwarding nodes before $T^{F}_{timeout}$ expires, it retransmits the query packet. In the retransmitted packet, the scheduling integer I is set to -1 for those nodes whose transmissions have been overheard by the sender. A forwarding node, whose transmission was not overheard by the sender, transmits an explicit ACK on receipt of a duplicate packet. A forwarding node, whose transmission was overheard by the sender, ignores the packet. All the non-forwarding nodes, whose ACK timers are active, reset their default holding timers when they receive a duplicate query.

The sender expects to receive all ACKs from the non-forwarding nodes in $T^{NF}_{timeout}$ amount of time and therefore $T^{NF}_{timeout}$ is set to $(2T_{p'} + 2T_p + T^{Q}_{t} * (I_{max}{+}1) + T^{ACK}_{t} * (I_{max'}{+}1))$ where $I_{max'}$ is the largest integral number assigned to a non-forwarding node and $T_{p'}$ is the propagation delay involved in reaching that node . If the sender has not received ACKs from all the non-forwarding nodes before $T^{NF}_{timeout}$ expires, it reschedules transmissions among the non-forwarding nodes whose ACKs it has not received.

Since for geocasting scheme, the nodes have been categorized into forwarding and non-forwarding nodes, we present separate FSM for forwarding and non-forwarding nodes shown in Figs. 4.9 and 4.10. The FSM for a forwarding node is same as that of a node with one-hop neighbor knowledge. After receiving all the overhearings from forwarding nodes in the 'wait' state, a node continues to be in that state until it receives explicit ACKs. A node in this case enters from 'wait' state to 'idle' state only after all the overhearings and ACKs have been received from forwarding and non-forwarding nodes respectively. When a node enters the 'wait' state after 'transmitted packet' or 'retransmitted packet' state, it starts timeout timer for forwarding as well as non-forwarding nodes. A non-forwarding node has only two states: 'idle' and 'wait'. Transition from 'idle' state to 'wait' state is made after reception of query packet and hold-off timer for ACK is started. On the expiry of this hold-off timer, the node

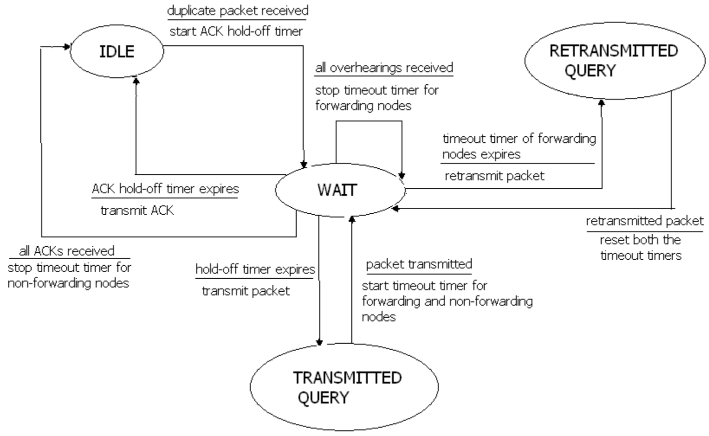

Figure 4.9: FSM for a forwarding node inside the geocast region with two-hop neighbor knowledge

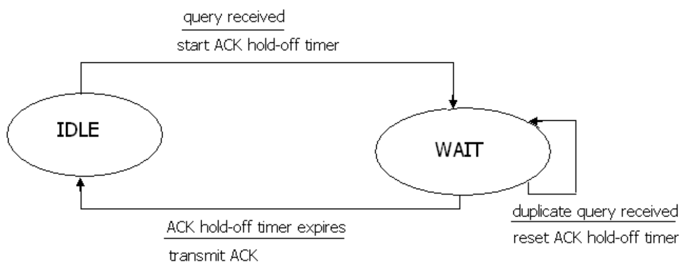

Figure 4.10: FSM for a non-forwarding node inside the geocast region with two-hop neighbor knowledge

transmits an ACK and goes back to the 'idle' state. The ACK hold-off timer is reset if a node receives duplicate query during the 'wait' state.

Chapter 5

Performance Evaluation

In this chapter, we discuss the simulation performance of different versions of protocol, presented in chapter 3. The simulations are performed in ns-2 [32]. The underwater channel is modeled in ns-2 to characterize the statistical behaviour of the environment, in order to evaluate the performance of the proposed protocols. Thus ns-2 is modified in order to account for the propagation delay and transmission loss incurred in underwater channel, thereby adapting it to underwater environment.

5.1 Underwater Channel Model

The underwater transmission loss describes how the acoustic intensity decreases as an acoustic pressure wave propagates outwards from a sound source. The deterministic transmission loss $TL^D(d, f_0)$ [dB] that a narrow-band acoustic signal centered at frequency f_0[kHz] experiences along a distance d[m] can be described by Urick propagation model [31],

$$TL^D(d, f_0) = 20 \cdot log_{10}(d) + \alpha(f_0) \cdot d \tag{5.1}$$

Equation 5.1 models how the acoustic intensity decreases as the pressure wave propagates outwards from the sound source. The first term accounts for *geometric spreading*, which increases with propagation distance and is independent of frequency. The second term accounts for *medium absorption*, where α(f)[dB/m] represents the medium absorption coefficient and quantifies the dependency of transmission loss on frequency band. The above equation only characterizes the deterministic nature of the channel. In this paper, we have also modeled the statistical behavior of the underwater channel as explained below:

Assuming that the time is discreet, $t_{k+1} = t_k + T_c$, where T_c is the coherence time of

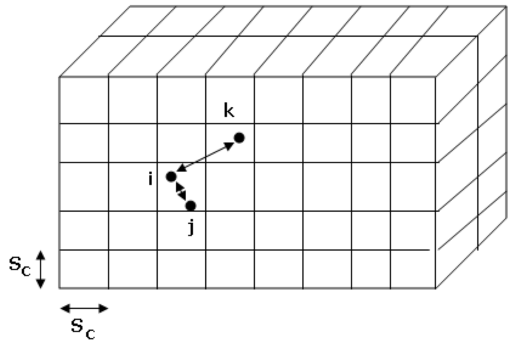

Figure 5.1: Implementation of coherence distance)

the channel. The entire 3D body of water, which is assumed to be a giant parallelopiped, is divided into cubes, w, h, etc., each with side S_c, which is taken as the coherence distance. We have implemented a matrix $R(t_k) = [\rho_{wh}]$, which stores random variables ρ, with a unit-mean Rayleigh distribution, to account for the statistical attenuation in the channel from cube w and cube h. The use of Rayleigh random variable gives the worst case behavior of the channel (saturation condition), as it is often the case in shallow water environment [7][33]. As shown in Fig. 5.1, nodes i and j are within the coherence distance. Whereas, k is not within the coherence distance, since its in different cube.

Thus, the statistical transmission loss is modeled as,

$$TL_{ij}(t_k) = TL_{ij}^{D} \cdot \rho_{wh}^{2} \tag{5.2}$$

where i and j are the sending and the receiving nodes respectively, w and h are the cubes where i and j are, respectively, ρ is an element in the matrix R at time t_k, which is recomputed every T_c seconds. Properties of matrix R: (i) ρ_{ww}=1 (transmission within the coherence distance), (ii) $\rho_{wh} \neq \rho_{hw}$ (link asymmetry), (iii) R does not have memory, i.e., $R(t_{k+1})$ does not depend on $R(t_k)$.

Thus, we have implemented spatio-temporal variation in the characterization of the underwater channel. Also, we have introduced link asymmetry, which is often the case in such an environment.

5.2 Simulation Scenarios and Results

We consider randomly deployed nodes in a 3D volume. The nodes transmit at a maximum power at all times. The value of R_{max} is taken as the maximum distance at which, 85% packets are received from a node. The simulation parameters are set as shown in Table 5.1.

Table 5.1: Simulation parameters

Parameter	value
3D volume dimensions	6x6x0.1 Km^3
bandwidth	30 KHz
data rate	40 Kbps
maximum transmission power	10 W
R_{max} (statistical)	3 Km
packet size	500 Bytes
τ	1 s
T_c	0.5 s
S_c	1 m

The performance evaluation is done in terms of the end-to end metrics: (i) packet delivery ratio, (ii) delay, (iii) energy/bit. The end-to-end metrics are considered by: (i) varying the packet inter-arrival times for the nodes, (ii) varying the number of nodes deployed in the 3D environment. In addition to comparing the performance of different knowledge based proposed protocols, comparison is also made between the proposed protocols and new protocols, which have equal amount of neighborhood information but do not fully exploit the information possessed in making MAC and routing decisions. Thus, protocols A,B and C, which are the modified versions of the protocols in sections 4.0.4, 4.0.5 and 4.0.6 respectively, are described as below:

- Protocol A: no neighbor knowledge based protocol with no synchronization in starting the hold-off timers. Thus this version of the protocol does not provide fairness among the transmitting nodes.

- Protocol B: one-hop neighbor knowledge based protocol with hold-off time taken as zero i.e., data packet is transmitted as soon as it is received, thereby, not catering the potential collisions. Thus one-hop neighborhood information is not fully exploited in designing the MAC scheme.

- Protocol C: two-hop neighbor knowledge based protocol with deterministic timeout period, which is taken as the minimum time required to overhear transmission from the next hop. This does not cater the potential collisions at the next two forwarding hops. Further, a node chooses its own next hop, i.e., the next hop is not designated by the sender.

The scenarios considered for comparison are:

- Static environment: the nodes are fixed sensors, i.e, there is no mobility;
- Mobility in case of gliders: the nodes follow sawtooth motion in the given 3D region. This is to account for the fact that gliders are designed to glide from ocean surface to bottom of the ocean and back, while measuring parameters along a sawtooth trajectory through water. Since gliders are slower than propelled vehicles, the maximum speed is taken as 1m/s. Thus it is a low mobility, low velocity environment;
- Mobility in case of AUVs: the nodes follow random waypoint motion in the given 3D region. Unlike buoyancy driven gliders, AUVs do not follow a sawtooth motion underwater. Hence, their motion is described using random waypoint model. To cater the higher speeds of these vehicles as compared to gliders, their maximum speed is taken to be 2m/s. This corresponds to high mobility, high velocity environment.

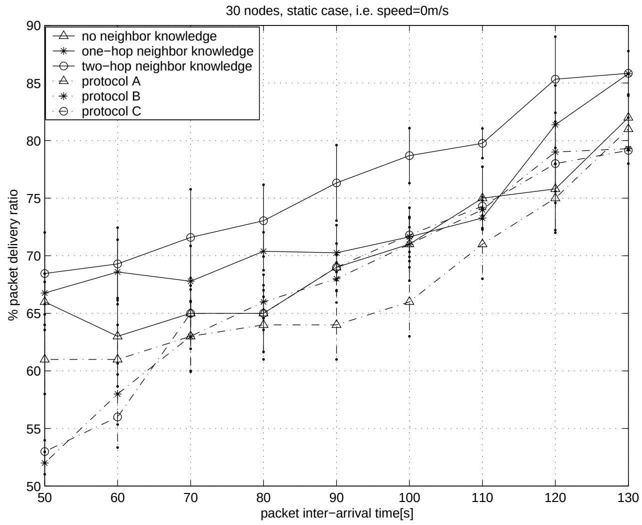

Figure 5.2: Packet delivery ratio vs. packet inter-arrival time (no mobility)

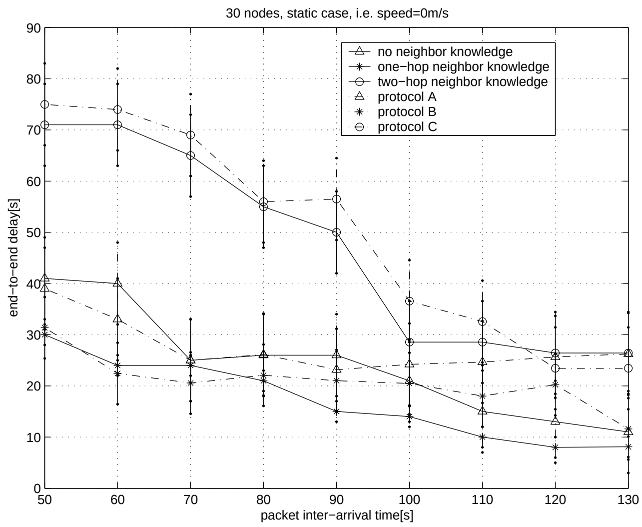

Figure 5.3: Delay vs. packet inter-arrival time (no mobility)

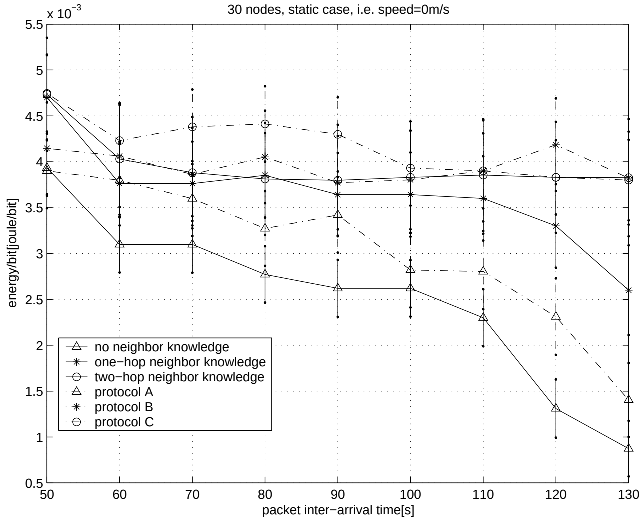

Figure 5.4: Energy/bit vs. packet inter-arrival time (no mobility)

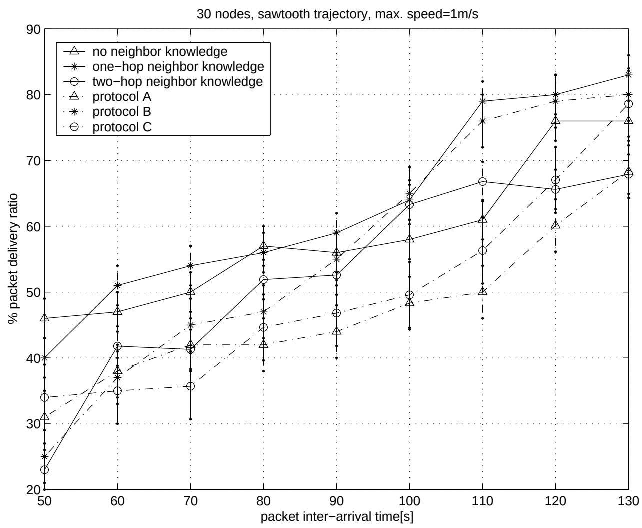

Figure 5.5: Packet delivery ratio vs. packet inter-arrival time (sawtooth trajectory)

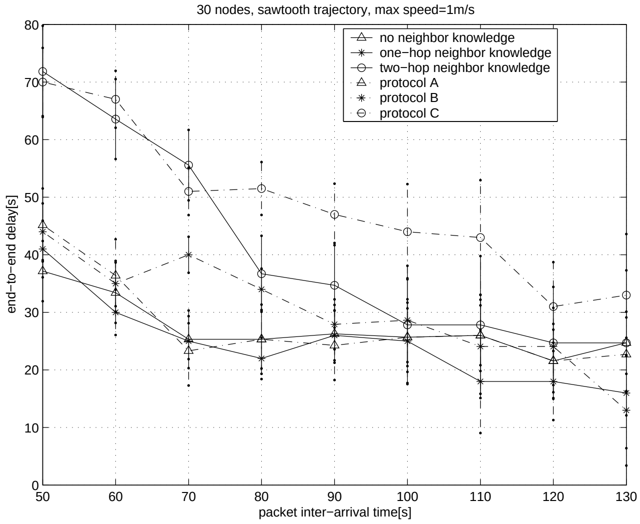

Figure 5.6: Delay vs. packet inter-arrival time (sawtooth trajectory)

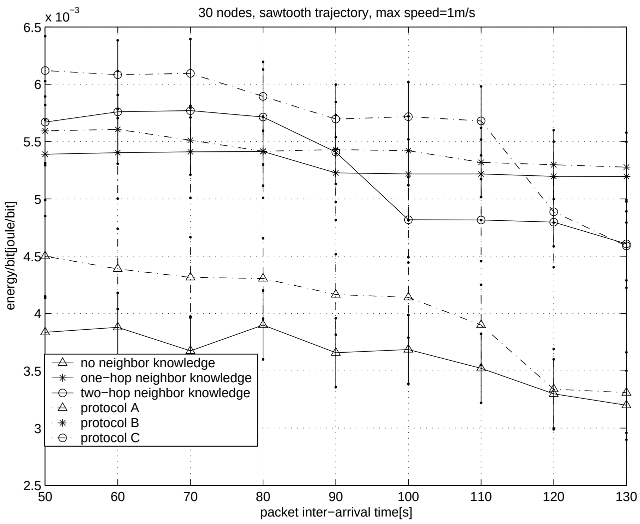

Figure 5.7: Energy/bit vs. packet inter-arrival time (sawtooth trajectory)

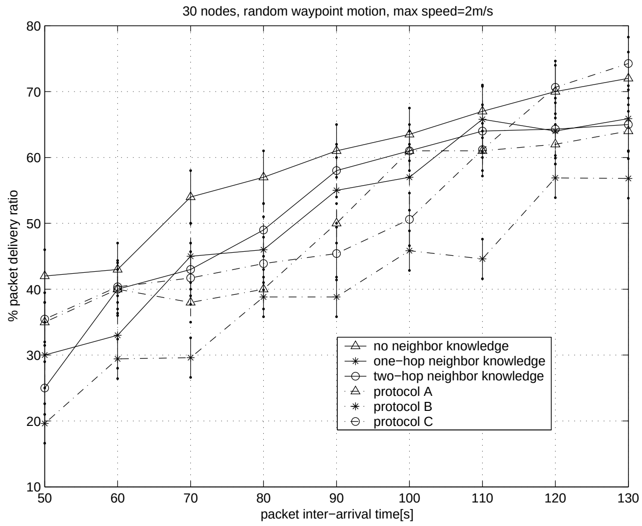

Figure 5.8: Packet delivery ratio vs. packet inter-arrival time (random waypoint motion)

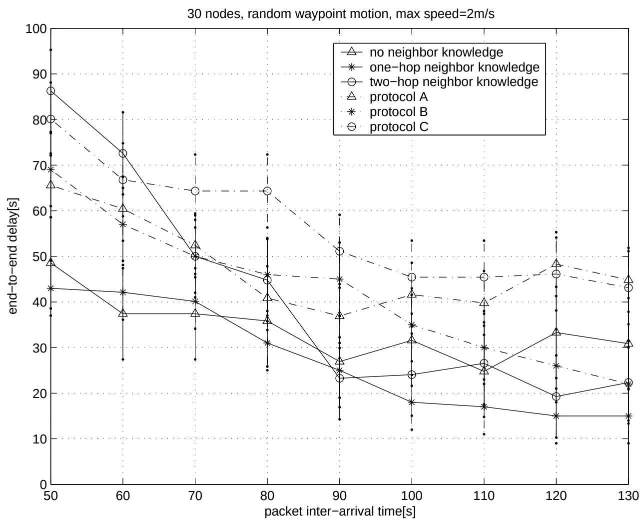

Figure 5.9: Delay vs. packet inter-arrival time (random waypoint motion)

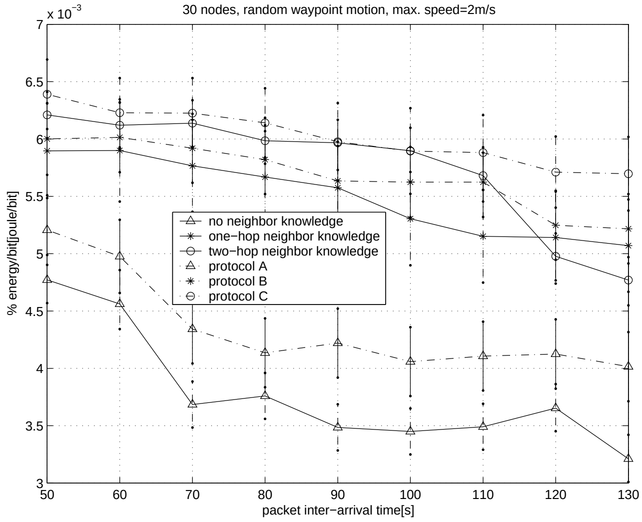

Figure 5.10: Energy/bit vs. packet inter-arrival time (random waypoint motion)

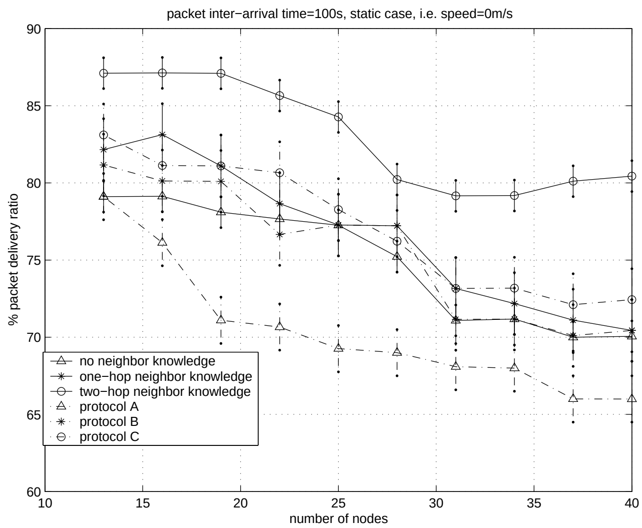

Figure 5.11: Packet delivery ratio vs. number of nodes (no mobility)

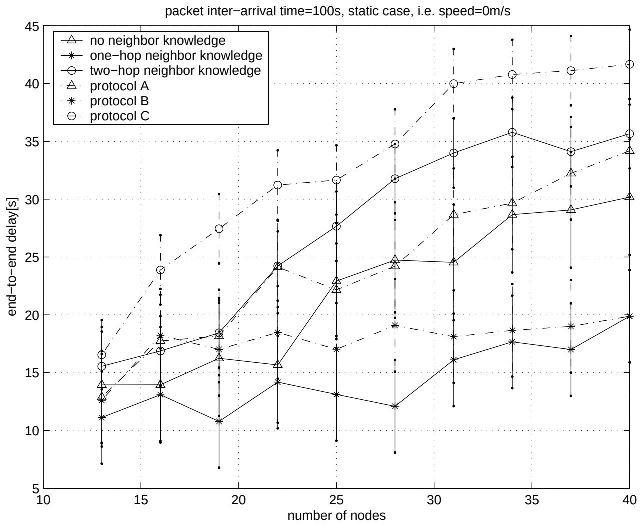

Figure 5.12: Delay vs. number of nodes (no mobility)

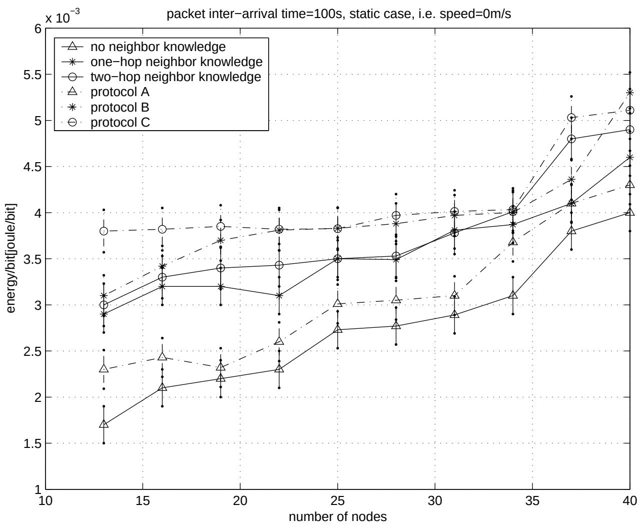

Figure 5.13: Energy/bit vs. number of nodes (no mobility)

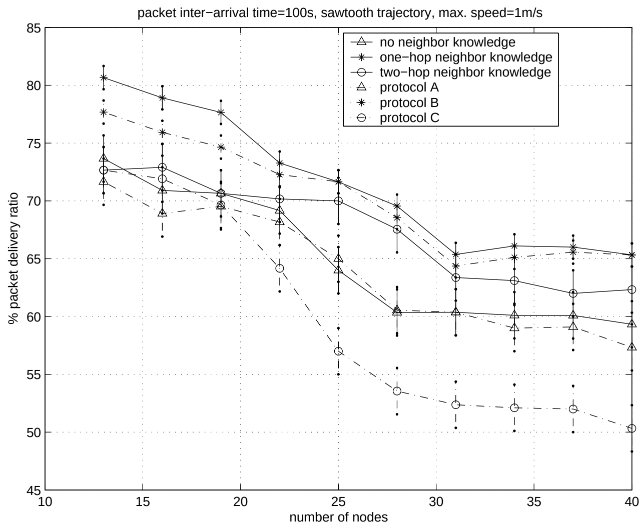

Figure 5.14: Packet delivery ratio vs. number of nodes (sawtooth trajectory)

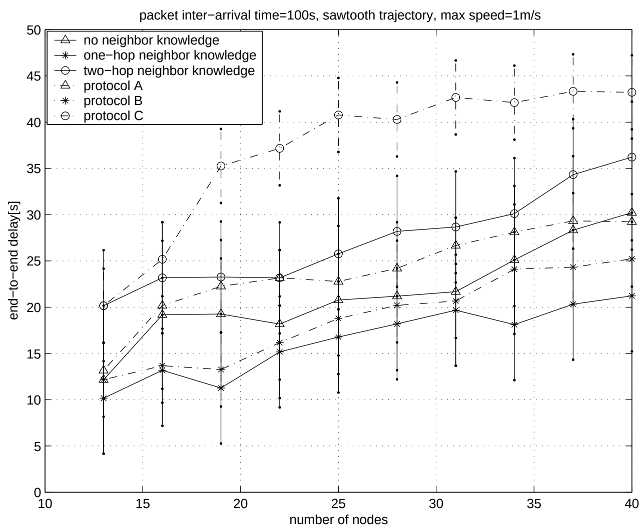

Figure 5.15: Delay vs. number of nodes (sawtooth trajectory)

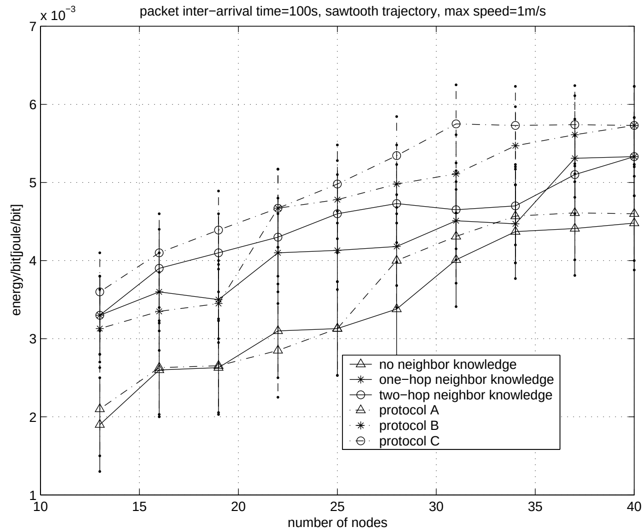

Figure 5.16: Energy/bit vs. number of nodes (sawtooth trajectory)

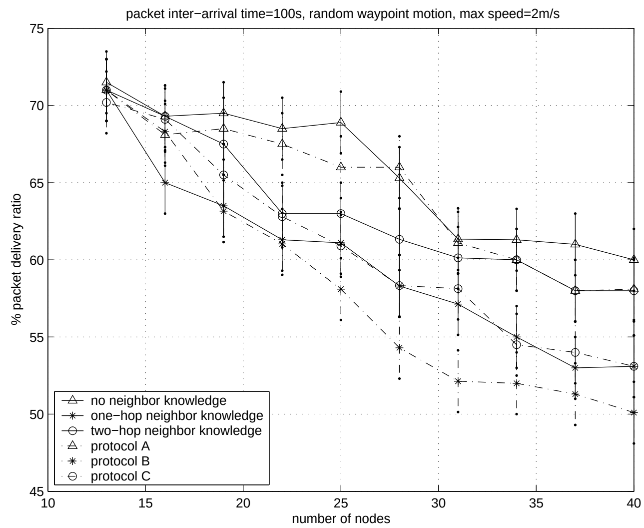

Figure 5.17: Packet delivery ratio vs. number of nodes (random waypoint motion)

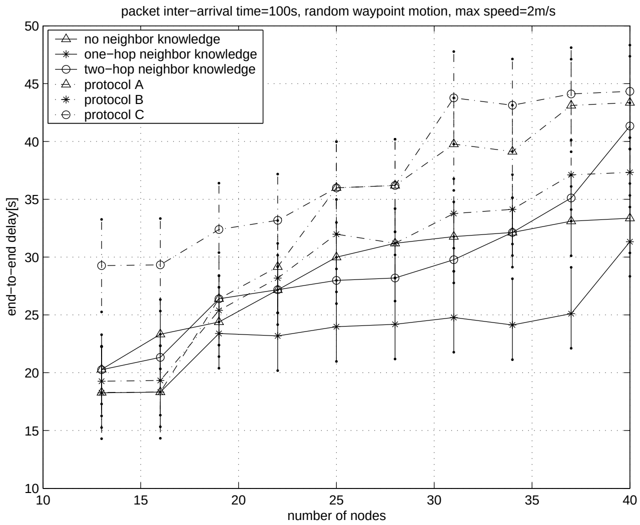

Figure 5.18: Delay vs. number of nodes (random waypoint motion)

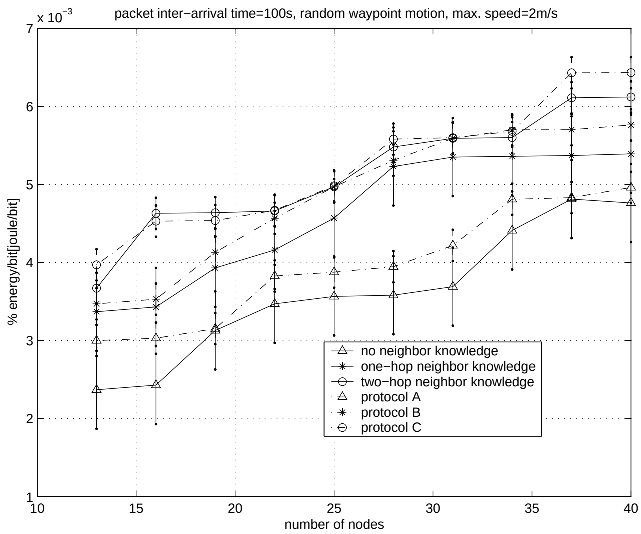

Figure 5.19: Energy/bit vs. packet inter-arrival time (random waypoint motion)

Chapter 6

Conclusions and Future Work

We have proposed three versions of unicast and geocast protocols, which integrate MAC and routing functionalities, and optimize different levels of neighbor knowledge for making routing decisions. The different levels of neighbor knowledge used are no neighbor knowledge, one-hop neighbor knowledge and two-hop neighbor knowledge. Performance evaluation of unicast protocol has been done in ns-2, and comparisons between different versions have been performed in terms of the end-to-end metrics: (i) packet delivery ratio, (ii) delay, (iii) energy/bit for static as well as mobile scenarios. Mobility has been considered for AUVs and gliders. From the simulation results, we observe the following:

As the packet inter-arrival time increases, the number of collisions decrease. This gives rise to decrease in end-to-end delay, decrease in energy/bit and improved reliability. On the other hand, as the number of nodes increases, it leads to greater collisions and affects the reliability. This is because, with an increase in the number of nodes, traffic increases.

The proposed three versions of the protocol based on no neighbor knowledge, one-hop neighbor knowledge and two-hop neighbor knowledge outperform protocols A, B and C respectively. The MAC schemes in protocols A, B and C result in more collisions than the proposed versions. This leads to greater retransmission tries for a data packet. Consequently, it results in increased end-to-end delays and energy/bit and decreased packet delivery ratio. Thus, the proposed protocols that fully exploit the neighbor knowledge possessed, in designing the MAC and the routing schemes, give an improved performance in terms of the end-to-end metrics considered. Consider the static scenario. In Fig. 5.2, the packet delivery ratio for two-hop neighbor knowledge based protocol is approximately 10% greater in low traffic environment and approximately 20% greater in high traffic environment as compared to protocol C. Similarly, one-hop neighbor knowledge shows approximately 5% greater packet delivery ratio than protocol B in low traffic scenarios, and 20% greater value in high traffic case. On the other hand, no neighbor knowledge shows an increase of 5% in packet delivery ratio than protocol A. Fig. 5.3 shows a better delay performance for no neighbor knowledge, and one-hop knowledge versions than protocols A and B respectively, especially in low traffic environment and consistently better performance for two-hop neighbor knowledge than protocol C. Similarly, Fig. 5.4 shows better performance throughout in terms of energy/bit for the three versions of protocol as compared to their counterparts. Figs. 5.11, 5.12 and 5.13 show similar results by varying the number of nodes.

For static environment, one version does not always outperform the other versions, but it depends on the end-to-end metric considered. From Fig. 5.2, we see that two-hop neighbor knowledge performs the best in terms of packet delivery ratio. Greater amount of neighborhood information helps the protocol make better routing decisions. One-hop neighbor outperforms the other two versions in terms of end-to-end delay. This is because, one-hop neighborhood information helps the protocol reduce collisions between the transmission of neighbors, resulting in lesser retransmissions and less delays. Although two-hop neighbor knowledge also reduces collisions at neighbors, there are delays involved in the transmission of variable length control messages whose size depends on the number of neighbors of a node. Thus the improved reliability in this case is at the cost of higher end-to-end delays involved in data delivery. The size of control messages in one-hop neighbor knowledge is fixed and smaller than the former case, which results in smaller amount of overhead involved. In terms of energy consumption, no neighbor knowledge performs the best. This is because energy is only consumed in transmission/reception of data packets since there is no control messaging.

Under the assumption that there is no mobility prediction, higher the mobility, lesser is the amount of information needed for making optimum routing decisions. Because of the mobility, neighborhood information gets outdated. This is evident from Fig. 5.14 and Fig. 5.17. On comparing with the static case, the packet delivery ratio for two-hop neighbor knowledge drops by approximately 20% for mobile scenarios (glider and AUV motion) for low traffic environment and by approximately 30% in case of high traffic. The corresponding drop in packet delivery ratio is 5%-10% for one-hop neighbor knowledge in case of high traffic, and 15%-25% for low traffic case. Similar performance is shown by no neighbor-knowledge based protocol in high as well as low traffic cases. Thus two-hop neighbor knowledge affects the worst in terms of mobility. With increase in mobility from gliders to AUVs, there is a less drop in packet delivery ratio for the three protocol versions as compared to the static case. Thus, mobility decreases the reliable performance of the protocols. Higher the mobility, smaller is the packet delivery ratio, and greater is the delay and energy/bit involved. One-hop knowledge performs the best for gliders in terms of packet delivery ratio, whereas no neighbor knowledge outperforms the others for AUVs.

In future, performance evaluation of the proposed versions of geocast protocol needs to be carried out. Consequently, an optimal level of neighbor knowledge required for query dissemination can be determined, based on the application requirements. Thus depending on application requirements, the necessary amount of neighbor knowledge can be exploited. Further, the performance of geocast protocol can be evaluated by considering different shapes of the geocast region in addition to the cylindrical region considered in the work.

References

[1] AUV Laboratory at MIT Sea Grant.

[2] B.M. Howe, T. McGinnis, and H. Kirkham. Sensor Networks for Cabled Ocean Observatories. In *Geophysical Research Abstracts*, volume 5, 2003.

[3] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater Acoustic Sensor Networks: Research Challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257–279, May 2005.

[4] Michael Hinchey. Development of a Small Autonomous Underwater Drifter. In *Proc. of IEEE Newfoundland Electrical and Computer Engineering Conference (NECEC)*, Chicago, IL, USA, November 2004.

[5] R.E. Davis, C.E. Eriksen, and C.P. Jones. Autonomous Buoyancy-driven Underwater Gliders. In G. Griffiths, editor, *The Technology and Applications of Autonomous Underwater Vehicles*. Taylor and Francis, London, 2002.

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

[7] J.G. Proakis, E.M. Sozer, J.A. Rice, and M. Stojanovic. Shallow Water Acoustic Networks. *IEEE Communications Magazine*, pages 114–119, November 2001.

[8] M. Stojanovic. Acoustic (Underwater) Communications. In John G. Proakis, editor, *Encyclopedia of Telecommunications*. John Wiley and Sons, 2003.

[9] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Trans. Networking*, 5(6):756–769, December 1997.

[10] J.B. Postel. Transmission Control Protocol. Information Sciences Institue, Marina del Rey, CA, RFC-793, Sept. 1981.

[11] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, 2000.

[12] C.E. Perkins and P. Bhagwat. Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In *Proc. of ACM Special Interest Group on Data Communications (SIGCOMM)*, London, UK, 1994.

[13] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *Proc. of IEEE Multi Topic Conference (INMIC)*, pages 62–68, Pakistan, December 2001.

[14] C.E. Perkins, E.M. Belding-Royer, and S. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. IETF RFC 3561.

[15] David B. Johnson, David A. Maltz, and Josh Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. In Charles E. Perkins, editor, *Ad Hoc Networking*, pages 139–172. Addison-Wesley, 2001.

[16] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *ACM-Kluver Wireless Networks (Springer)*, 7(6):609–616, November 2001.

[17] Tommaso Melodia, Dario Pompili, and Ian F. Akyildiz. On the Interdependence of Distributed Topology Control and Geographical Routing in Ad Hoc and Sensor Networks. *IEEE Journal of Selected Areas in Communications*, 23(3):520–532, March 2005.

[18] G. Xie and J.H. Gibson. A Network Layer Protocol for UANs to Address Propagation Delay Induced Performance Limitations. In *Proc. of MTS/IEEE Conference and Exhibition for Ocean Engineering, Science and Technology (OCEANS)*, volume 4, pages 2087–2094, Honolulu, HI, November 2001.

[19] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data Collection, Storage, and Retrieval with an Underwater Sensor Network. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, November 2005.

[20] Dario Pompili, Tommaso Melodia, and Ian F. Akyildiz. Routing Algorithms for Delay-insensitive and Delay-sensitive Applications in Underwater Sensor Networks. In *Proc. of ACM Conference on Mobile Computing and Networking (MobiCom)*, Los Angeles, LA, September 2006.

[21] Paolo Casari and Albert F. Harris. Energy-Efficient Reliable Broadcast in Underwater Acoustic Network. In *Proc. of ACM International Workshop on Underwater Networks (WUWNet)*, Montreal, Canada, September 2007.

[22] Sangho Oh, Jaewon Kang, and Marco Gruteser. Location-Based Flooding Techniques for Vehicular Emergency Messages. In *Proc. of IEEE International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS)*, San Jose, CA, July 2006.

[23] Arjan Durresi, Vamsi Paruchuri, S. Sitharama Iyengar, and Rajgopal Kannan. Optimized Broadcast Protocol for Sensor Networks. *IEEE Transactions on Computers (TC)*, 54(8):1013–1024, August 2005.

[24] Richard Kershner. The Number of Circles Covering a Set. *American Journal of Mathematics*, 61(3):665–671, July 1939.

[25] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, WA, August 1999.

[26] Wentao Zhang, Xiaohua Jia, and Chuanhe Huang. Energy-Aware Location-Aided Multicast Routing in Sensor Networks. In *Proc. of IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WCNM)*, Wuhan, China, September 2005.

[27] Young-Bae Ko and Nitin H. Vaidya. Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms. In *Proc. of IEEE Workshop on Mobile Computer Systems and Applications (WMCSA)*, Washington DC, February 1999.

[28] Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic. Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks. *IEEE Transactions on Parallel and Distributed Systems (PDS)*, 13(1):14–25, January 2002.

[29] Jie Wu and Fei Dai. Broadcasting in Ad Hoc Networks Based on Self-Pruning. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, April 2003.

[30] Dimitrios Katsaros and Yannis Manolopoulos. The Geodesic Broadcast Scheme for Wireless Ad Hoc Networks. In *Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Buffalo, NY, August 2006.

[31] Robert J. Urick. *Principles of Underwater Sound*. McGraw-Hill, 1983.

[32] LBNL UCB and VINT. Network simulator (ns), version 2.

[33] J. Catipovic. Performance Limitations in Underwater Acoustic Telemetry. *IEEE Journal of Oceanic Engineering*, 15:205–216, July 1990.