Reliability in Underwater Inter-Vehicle Communications

Amrita A. Nimbalkar and Dario Pompili

Rutgers, The State University of New Jersey
Department of Electrical and Computer Engineering
94 Brett Road, Piscataway, NJ 08854

amritan@eden.rutgers.edu, pompili@ece.rutgers.edu

ABSTRACT

Underwater networks are envisioned to enable several applications for oceanographic data collection, environmental monitoring, navigation, and tactical surveillance. Underwater acoustic networking is the enabling technology for these applications. Most of these applications make use of underwater vehicles and rely on inter-vehicle communication capabilities for information exchange and coordination purposes. Reliable data delivery, especially in the case of mobile underwater vehicles, is therefore a major concern in many of these applications. In this paper, three versions of a reliable unicast protocol are proposed, which integrate MAC and routing functionalities and leverage different levels of neighbor knowledge for making optimum decisions for reliable data delivery. The different levels of neighbor knowledge used by the protocols are: (i) no neighbor knowledge, (ii) one-hop neighbor knowledge, and (iii) two-hop neighbor knowledge. The three versions of the protocol have been devised by considering the peculiar characteristics of underwater channel, in design as well as in performance simulation. The protocols have been compared in static as well as mobile scenarios in terms of different end-to-end networking metrics.

Categories and Subject Descriptors:
C.2.1 [Computer-Communication Networks]: Network Architecture and Design-*Network topology*

General Terms: Design, Performance, Reliability.

Keywords: Underwater Sensor Networks, Routing, MAC.

1. INTRODUCTION

Underwater sensor networks have the potential to enable unexplored applications and to enhance our ability to observe and predict the ocean. Unmanned or Autonomous Underwater Vehicles (UUVs, AUVs), equipped with underwater sensors, are also envisioned to find applications in exploration of natural undersea resources and gathering of scientific data in collaborative monitoring missions. These potential applications will be made viable by enabling communications among underwater devices. UnderWater Acoustic Sensor Networks (UW-ASNs) [1] consist of sensors and vehicles deployed underwater to perform collaborative monitoring tasks for scientific, environmental, commercial, safety, and military applications. UW-ASN communication links are based on *acoustic wireless technology*, which poses challenges due to the unique underwater environment such as limited bandwidth capacity [9], high and variable propagation delays [8], high bit error rates, and temporary losses of connectivity caused by multipath and fading phenomena [10].

Owing to the peculiar characteristics of the underwater environment, reliable communication is a fundamental primitive for underwater networks. 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 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 three nodes A, B, and 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 the packet delivery to B. After B receives packet from A, either of the following 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. Hence, even though the protocol ensures hop-by-hop reliability, because of the described situations, 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 window-based (e.g., TCP) and rate-based mechanisms [12]. Although transport solutions are crucial for reliable communication, in the underwater environment end-to-end retransmissions would result in large Round Trip Times (RTTs). This is due to the fact that the underwater acoustic propagation delay is five orders of magnitude higher as compared to the delay in radio frequency terrestrial sensor networks [1]. This, in turn, would delay the feedback from destination to the source leading to low end-to-end throughput and high end-to-end delays. For these reasons, in order to reduce the number of end-to-end retransmissions, in this paper we rely on lower-layer mechanisms to provide communication reliability. *Specifically, our proposed unicast protocol aims at maximizing the end-to-end reliability by providing high link-layer reliability.*

Ensuring link-layer reliability, especially in the case of UW-ASNs consisting of mobile 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

WUWNet'08, September 15, 2008, San Francisco, California, USA.
Copyright 2008 ACM 978-1-60558-185-9/08/09 ...$5.00.

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 leveraging cross-layer interactions between MAC and routing layers, while optimizing different levels of neighbor knowledge: (i) no neighbor knowledge, (ii) one-hop neighbor knowledge, and (iii) two-hop neighbor knowledge.

Our proposed unicast protocol integrates neighbor discovery, MAC, and routing layer functionalities to route the packets from source to the destination. Unlike the existing solutions, our protocol does not assume certain amount of neighborhood information but it integrates the functionality of neighbor discovery phase with the cross-layer design. The protocol uses 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, such as those developed by WHOI and Benthos, use a random access MAC scheme. Also, we rely on a geographical routing mechanism, since underwater vehicles need to estimate their current position irrespective of the routing approach. The surface station, in fact, needs to be able to associate the data sampled by the underwater vehicles with the 3D position of the device that generates the data and spatially reconstruct the characteristics of the event. For this reason, we assume that the vehicles (hereafter nodes) in the network know their geographical co-ordinates. Moreover, because nodes in the underwater network are sparsely deployed, we assume that a unique ID is assigned to each node.

In this paper, we compare the performances of three versions of our protocol and determine which outperforms the other in terms of three different end-to-end networking metrics, i.e., packet delivery ratio, packet delay, and energy per successfully received bit. The performance of the protocol is then evaluated for static and different mobility scenarios. The aim is to find an optimal level of neighbor knowledge for different applications and mobility scenarios. The main features of our communication solution are the following.

1. 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.

2. 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.

3. 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 the Unit Disk Graph Model in design and implementation. Rather, the reception of packets by the nodes is based on the channel dynamics.

4. In the simulations, we have accounted for spatial and temporal variations in the channel conditions, which is a peculiar characteristic of the underwater channel.

The remainder of the paper is organized as follows. In Sect. 2, we review the background and existing work in UW-ASNs and discuss their suitability for reliable communication in an underwater environment. In Sect. 3, we introduce the three versions of our protocol and explain their formulation in depth. Finally, in Sect. 4, we show the performance evaluation of the proposed solutions, while in Sect. 5 we draw the main conclusions.

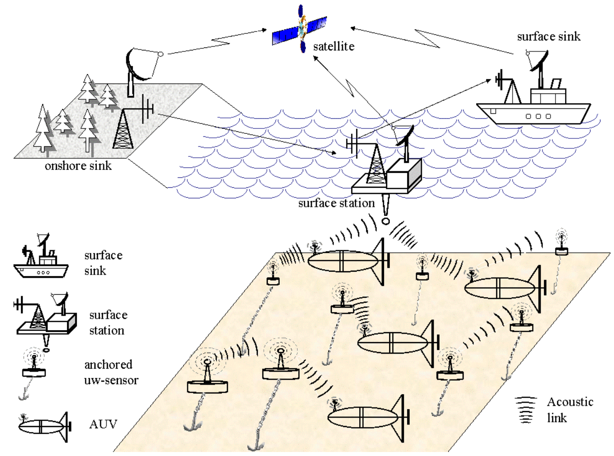


Figure 1: 3D Underwater Sensor Networks with AUVs.

2. BACKGROUND AND RELATED WORK

One of the design objectives of AUVs is to make them rely on local intelligence and be less dependent on communications from online shores [6]. 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].

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 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 [5]. Underwater gliders [4] 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 propeller-driven AUVs as well as gliders specifically, which differ in their motion. Gliders follow a sawtooth path though the water and have lower velocities compared to propelled-driven AUVs. Thus, gliders have a constrained motion, whereas AUVs are not constrained in mobility.

There has been an intensive work on routing protocols in the last few years for terrestrial ad hoc and wireless sensor networks. However, the different nature of the underwater environment poses several drawbacks with respect to the suitability of existing terrestrial routing solutions for underwater sensor networks. In fact, routing in UW-ASNs poses additional challenges due to the peculiarities of underwater channel such as limited bandwidth, very high and variable propagation delays, temporary losses of connectivity, channel asymmetry, and heavy multipath and fading phenomena.

Proactive protocols (e.g., DSDV and OLSR) 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. In this way, each node is able to establish a path to any other node in the network, which may not be needed in underwater networks. Reactive protocols (e.g., AODV and DSR) are more appropriate for dynamic environments but incur high latency. Thus, proactive and reactive protocols are not suitable for underwater environment. For these reasons, we consider a geographical routing scheme.

In [7], 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 paper. Moreover, the paper does not consider mobility in the work.

In [2], the authors propose a scheme for reliable communication in UW-ASNs. To ensure reliability, the authors propose a separate control and data channel. 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, the selection of nodes that broadcast the data is random. Conversely, in our paper, 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. This promotes faster propagation of data packets to the destination.

3. RELIABLE COMMUNICATIONS

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 no neighbor knowledge protocol, described in Sect. 3.1, the routing decisions are made by the receiving nodes themselves with the help of their own position and the destination's position (*contention at the receivers*). Since there is no neighbor knowledge, a node receiving a packet in turn schedules a packet broadcast transmission according to the MAC rules described in the following. Conversely, in the one-hop neighbor knowledge protocol, described in Sect. 3.2, the receiver makes use of its neighbor knowledge and selects the neighbor closest to the destination as its next hop to unicast the packet. Finally, in case of the two-hop neighbor knowledge protocol, presented in Sect. 3.3, 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.

3.1 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 so to reduce retransmissions and avoid 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} [s], is a uniform random variable in [0, $2T_{hold}^{mean}$], where the mean T_{hold}^{mean} is given by,

$$T_{hold}^{mean} = \frac{d_{id}}{d_{sd}} \cdot \tau + \frac{\phi_{si}}{c}, \qquad (1)$$

$$\phi_{si} = \begin{cases} R_{max} - d_{si} & \text{if } R_{max} \geq d_{si} \\ 0 & \text{if } R_{max} < d_{si}, \end{cases} \qquad (2)$$

where d_{si} [m] 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 [s] is a constant parameter whose optimum value is determined in the following, c [m/s] is the speed of the underwater acoustic signal [11], and R_{max} [m] is taken as a constant parameter for simulation purposes.

The mean value of the hold-off timer, T_{hold}^{mean} in (1), 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 its hold-off timer. The factor ϕ_{si}/c in (1) represents an extra delay that a node should wait to allow all the nodes to receive the packet. Thus, this factor 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.

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. Hence, on expiration of the hold-off timer, the forwarding node transmits the packet and starts a timer $T_{timeout}$ [s] given by,

$$T_{timeout} = T_{hold}^{max} + \frac{R_{max}}{c} + T_t^D, \qquad (3)$$

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 $T_{hold}^{max} = 1.8 T_{hold}^{mean}$. A node that is closest to the sender ($d_{id} \approx d_{sd}$) will have the maximum hold-off time, whose mean value is given by (1),

$$T_{hold-max}^{mean} = T_{hold}^{mean}|_{d_{si}=0} = \tau + \frac{R_{max}}{c}. \qquad (4)$$

Note that, since (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, selection of uniform distribution serves this purpose as it gives the highest deviation. On expiration of ACK-hold-off timer, the forwarding node sends an explicit ACK.

Note that in (1), greater the value of τ, greater is the value of T_{hold}^{mean}. Thus, the selection of the constant parameter, τ, involves a trade-off as explained in the following conditions:

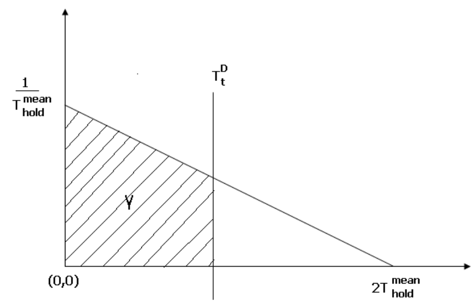


Figure 2: Probability Density Function of $\triangle T^{+}_{hold}$, as in (6).

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* 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. Also, a small value for τ will limit the medium access delay, which in turn has an impact on the overall end-to-end packet delay.

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. 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. Although collisions occur *at the receiver*, assuming uniform propagation delays for nodes i and j, i.e., the acoustic speed c does not significantly change spatially, gives rise to the following condition *at the transmitters*:

$$Pr(|T^i_{hold} - T^j_{hold}| \leq T_t^D) \leq \gamma, \tag{5}$$

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

Let $|T^i_{hold} - T^j_{hold}|$ be $\triangle T^{+}_{hold}$. The Probability Density Function (pdf) of $\triangle T^{+}_{hold}$, $P(\triangle T^{+}_{hold})$, is given as,

$$P(\triangle T^{+}_{hold}) = \begin{cases} \frac{1}{T^{mean}_{hold}} - \frac{\triangle T^{+}_{hold}}{2T^{mean\,2}_{hold}} & \text{if } 0 \leq \triangle T^{+}_{hold} \leq 2T^{mean}_{hold} \\ 0 & \text{if } \triangle T^{+}_{hold} > 2T^{mean}_{hold}. \end{cases} \tag{6}$$

Hence, (5) can be rewritten as,

$$Pr(\triangle T^{+}_{hold} \leq T_t^D) = \int_0^{T_t^D} \Big(\frac{1}{T^{mean}_{hold}} - \frac{\triangle T^{+}_{hold}}{2T^{mean\,2}_{hold}}\Big) d\triangle T^{+}_{hold} \leq \gamma$$

$$= \frac{T_t^D}{T^{mean}_{hold}} - \frac{T_t^{D^2}}{4T^{mean\,2}_{hold}} \leq \gamma. \tag{7}$$

Solving (7) for (1), and considering that $T_t^D \leq 2T^{mean}_{hold}$, we obtain,

$$T^{mean}_{hold} \geq \frac{T_t^D}{2\gamma}(1 + \sqrt{1-\gamma}) = \Psi. \tag{8}$$

From (1) and (8), an optimum value of τ is found out by formulating an optimization problem. The value of d_{sd} is taken as a constant parameter, and d_{si} and d_{sd}, being distances, are constrained to be positive. 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 $d_{si} \leq d_{sd} + d_{id}$. Thus, the optimization problem can be stated as follows.

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

Given : $d_{si} > 0,\ d_{id} > 0$

Find : τ^*

Minimize : $\tau = \frac{d_{sd}\cdot(\Psi + \frac{d_{si}}{c} - \frac{R_{max}}{c})}{d_{id}}$ (9)

Subject to :

$$d_{id} \leq d_{sd}; \tag{10}$$

$$d_{si} \leq d_{sd} + d_{id}. \tag{11}$$

For different values of d_{sd} and γ, the optimal τ^* is shown in Table 1. Considering the values of τ obtained, and the conditions involved in its selection, τ is taken as 1 s in the simulations.

Table 1: τ in seconds for different values of γ

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

3.2 One-Hop Neighbor Knowledge

Similar to no neighbor knowledge based protocol, a node with one-hop neighborhood knowledge ensures reliability by overhearing packet forwarding transmitted by receivers (*implicit ACK through overhearing*), and by requiring each node receiving a duplicated data packet to send an explicit ACK. However, unlike the previous case, the sender of the data packet inserts the ID of its next hop in the packet, i.e., the ID of the neighbor closest to the destination, whom it designates to be the next forwarding node. When a node receives the 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. 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. When node $k-1$ transmits the data packet to node k, it starts a $T^{k-1}_{timeout}$ timer. 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,

$$T^{k-1}_{timeout} = T_t^D + 2T_p^{k-1,k} + T^{max}_{hold}, \tag{12}$$

where $T_p^{k-1,k}$ is the propagation delay from node $k-1$ to k, T^{max}_{hold} is the maximum value of the holding time, which may vary in $[0, 2T_t^D]$, and 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^k_{hold} in [0, T^{max}_{hold}] in order to transmit the packet to its next-hop neighbor $k+1$. The value of T^k_{hold} 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,

$$T_{k-1} = 2T_t^D + 2T_p^{k-1,k} + T^{max}_{hold}, \tag{13}$$

$$T_{k+1} = 2T_t^D + 2T_p^{k,k+1} + T^{k}_{hold} + T^{max}_{hold}, \tag{14}$$

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

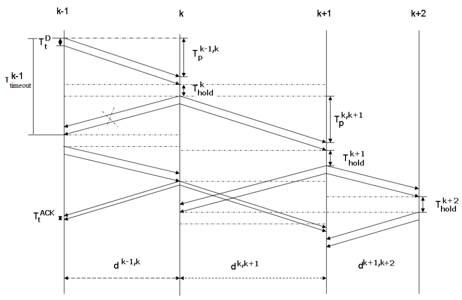


Figure 3: One-hop Neighborhood Knowledge Protocol.

From (13) and (14), it can be seen that node k leverages the knowledge of the distances from nodes $k-1$ and $k+1$ to determine the hold-off time of data packet and avoid collisions. On expiration of the hold-off timer, node k transmits the data packet. If node $k-1$ does not overhear the transmission of 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}. Node k schedules the transmission of ACK after the completion of overhearing from $k+1$.

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. Data packet retransmission 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. Also, the transmission of ACK is scheduled such that it reaches the sender before it timeouts and retransmits the packet.

3.3 Two-Hop Neighbor Knowledge

In two-hop neighbor knowledge based protocol, transmissions can be scheduled deterministically, with the help of *first and second hop distances*, to avoid collisions. Similar to the one-hop neighbor knowledge protocol described in Sect. 3.2, this protocol tries to avoid collisions at one-hop neighbor. In addition to this, two-hop neighborhood knowledge enables it to *avoid collisions at the second hop neighbor also*. In this protocol, the sender of the data packet inserts the IDs of the next two forwarding hops in the packet. Thus, the next hop for the intended receiver is designated by the sender, and the former chooses the second forwarding hop. The MAC scheme for the two-hop neighbor knowledge based protocol is designed as explained below.

By utilizing the two-hop topology information, each node is aware of its next two forwarding hops. This helps a node 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, to reach the chosen second hop. The best next hop A is so selected that it minimizes the time $T_p^{s,i} + T_p^{i,j}$, where $T_p^{s,i}$ is the propagation delay from s to i and $T_p^{i,j}$ is the propagation delay from i to j.

The source will then include the selected first and second hops (i and j, respectively) in the data packet and will 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 chooses the next two hops, whereas subsequent forwarding nodes only choose the next second hop.*

Consider nodes $k-1$, k, and $k+1$. As shown in Fig. 4(a), $k-1$ sends a packet to k. After receiving the data packet, k transmits it immediately. Node $k-1$ starts a timer, $T_{timeout}$, given by

$$T_{timeout} = 2T_p^{k-1,k} + T_t^D, \tag{15}$$

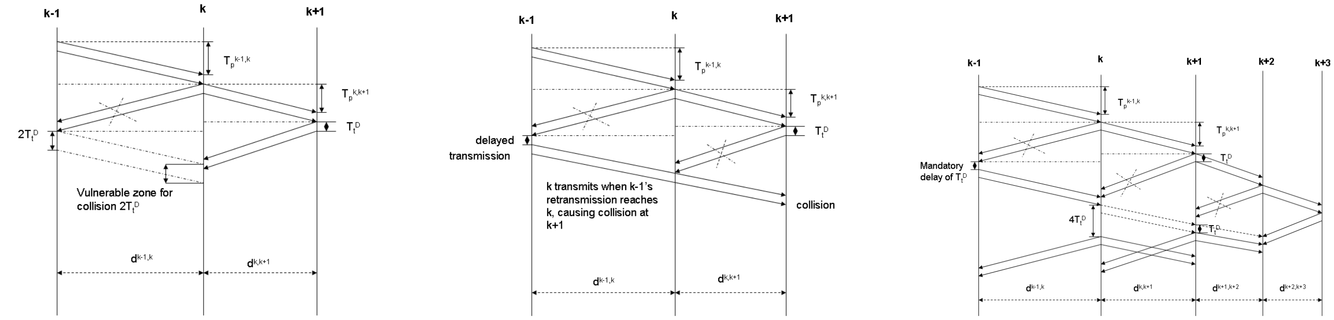
where $T_p^{k-1,k}$ is the propagation delay between node $k-1$ and k, and T_t^D is the data packet transmission time. The timer is stopped if either node $k-1$ is able to overhear node k's transmission of the data packet or if $k-1$ receives an explicit ACK from k. The timeout expiry implies two possible cases.

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 k.

In case 1, $k-1$ needs to retransmit immediately. On the other hand, in case 2, $k-1$ 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 $k+1$. 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. 4(b). 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 $k+1$, 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 to keep the delay incurred by $k-1$ small. This means that $k-1$ 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 k's retransmission causing collision at $k+1$. In this case, we



(a) (b) (c)

Figure 4: Two-hop neighbor knowledge protocol. (a): collision due to retransmission at one-hop neighbor; (b): collision due to retransmission at two-hop neighbor; (c): timing diagram.

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, summarized in Algorithm 1.

Algorithm 1 Two-hop knowledge: delaying the transmission

---

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 transmission by $2T_p^{k+1,k} + 4T_t^D - 2T_p^{k-1,k}$
  else
    do not delay retransmission after timeout
  end if
end if

In case $2T_p^{k,k+1} < 2T_p^{k-1,k} < 2T_p^{k,k+1} + 4T_t^D$, $k-1$ delays its retransmission by $2T_p^{k,k+1} + 4T_t^D - 2T_p^{k-1,k}$. This is done to give room to k to adjust its retransmission and avoid collision at $k+1$. The overall scenario is depicted in Fig. 4(c). Node $k-1$ is not able to overhear from 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 $k+1$'s retransmission at $k+2$. 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 $k+2$. Therefore, it delay its retransmission by T_t^D amount of time. Now, consider that node k receives a duplicate packet from $k-1$. 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, 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 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 $2T_t^D$ time units before k's next timeout. In this case, k immediately sends an ACK.

4. PERFORMANCE EVALUATION

4.1 Underwater Acoustic 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 [11],

$$TL^D(d, f_0) = 20 \cdot log_{10}(d) + \alpha(f_0) \cdot d. \qquad (16)$$

In (16), $\alpha(f)$ [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 [s] is the coherence time of the channel (0.5 s in the simulations). The entire 3D body of water, which is assumed to be a giant parallelepiped, is divided into cubes, w,h, etc., each with side S_c [m], which is taken as the coherence distance (1 m in the simulations). 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 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 (depth less than 100 m) [3][8]. Thus, the statistical transmission loss is modeled as,

$$TL_{ij}(t_k) = TL_{ij}^D \cdot \rho_{wh}^2, \qquad (17)$$

where i and j are the sending and the receiving nodes respectively, w and h are the cubes where i and j are located, respectively, ρ is an element in the matrix R at time t_k, which is recomputed every T_c seconds. Properties of matrix R are: (i) ρ_{ww}=1 (transmission within the coherence distance), (ii) $\rho_{wh} \neq \rho_{hw}$ (link asymmetry), and (iii) R does not have memory, i.e., $R(t_{k+1})$ does not depend on $R(t_k)$. In this way, we have implemented spatio-temporal variation in the characterization of the underwater channel. Also, we have introduced link asymmetry, which is often the case underwater.

4.2 Simulation Scenarios and Results

The simulations were performed in ns-2, which we modified to account for the channel characteristics described in Sect. 4.1. We consider randomly deployed sensors in a 3D volume of $6x6x0.1$ Km^3. The bandwidth is set to $30\ \mathrm{KHz}$, data rate to $40\ \mathrm{Kbps}$, transmission power to $10\ \mathrm{W}$, R_{max} to $3\ \mathrm{Km}$, packet size to $500\ \mathrm{Bytes}$ and τ to $1\ \mathrm{s}$. We compare the performances of three versions of the proposed protocol with three competing protocols that do not employ certain key features of the proposed solutions. These protocols represent existing solutions, which do not fully exploit neighbor knowledge in designing the MAC and routing schemes.

- Protocol A: no neighbor knowledge based protocol with no synchronization in starting the hold-off timers.
- Protocol B: one-hop neighbor knowledge based protocol with hold-off time taken as zero, i.e., packets are transmitted as received, thereby, not catering potential collisions.
- 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., next hop is not designated by the sender.

The comparison is made in the following three scenarios:

- *Static environment.* Nodes are fixed sensors (no mobility).
- *Mobility in case of gliders.* Nodes follow sawtooth trajectory in the given 3D region with a maximum speed of $1\ \mathrm{m/s}$.
- *Mobility in case of AUVs.* Nodes follow Random Waypoint motion in the 3D region with a maximum speed of $2\ \mathrm{m/s}$.

Figures 5, 6, and 7 lead to the following conclusions:

1. The 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.
2. 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, we notice that two-hop neighbor knowledge performs the best in terms of packet delivery ratio. Greater amount of neighborhood information helps the protocol to 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 less retransmissions and smaller 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. The size of control messages in one-hop neighbor knowledge is fixed and smaller than the former case. No neighbor knowledge performs the best in terms of energy consumption as energy is only consumed in the transmission of data packets.
3. Higher the mobility, less is the amount of information needed for making optimum decisions. Because of the mobility, information gets outdated. This is evident from Figs. 6 and 7. The packet delivery ratio decreases as mobility increases. One-hop knowledge performs the best for gliders in terms of packet delivery ratio whereas no neighbor knowledge outperforms the others for AUVs.

The optimal neighbor knowledge required in different mobility scenarios for different end-to-end metrics is shown in Table 2.

Table 2: Optimum neighbor knowledge

	static	glider motion	AUV motion
Max. delivery ratio	two-hop	one-hop	no knowledge
Min. e2e delay	one-hop	one-hop	one-hop
Min. energy/bit	no knowledge	no knowledge	no knowledge

5. CONCLUSIONS AND FUTURE WORK

We proposed different versions of a reliable unicast protocol, which integrate cross-layered functionalities between MAC and routing layers while optimizing different levels of neighbor knowledge for making reliable decisions. For performance evaluation of the three versions, we implemented the underwater characteristics and the three versions were compared based on different networking end-to-end metrics. We showed that different levels of neighbor knowledge can be used for different classes of applications. Thus, based on a particular application, we can leverage the necessary neighbor knowledge. As future work, the three versions of the proposed protocol will be implemented on WHOI Micro-modems.

6. REFERENCES

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

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

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

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

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

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

[7] D. Pompili, T. Melodia, and I. 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, Sept. 2006.

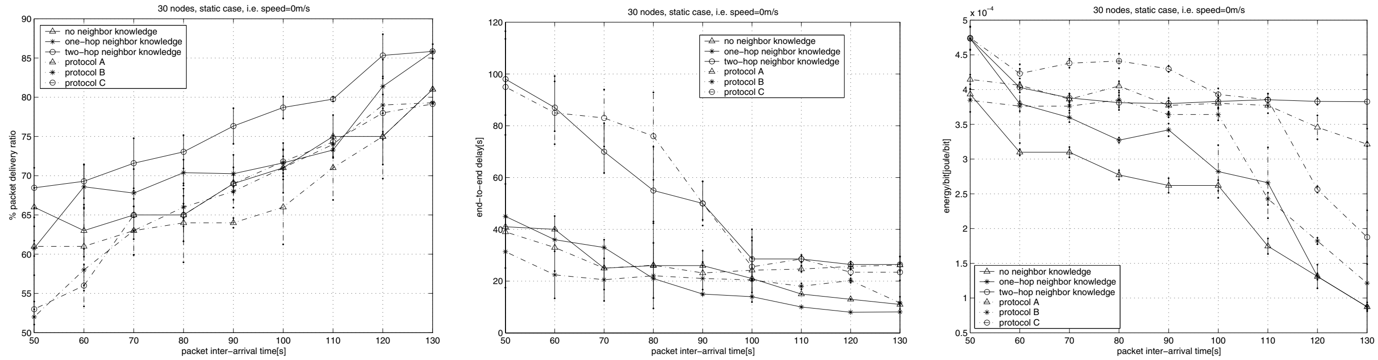
[8] J. Proakis, E. Sozer, J. Rice, and M. Stojanovic. Shallow Water Acoustic Networks. *IEEE Communications Magazine*, pages 114–119, Nov. 2001.

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

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

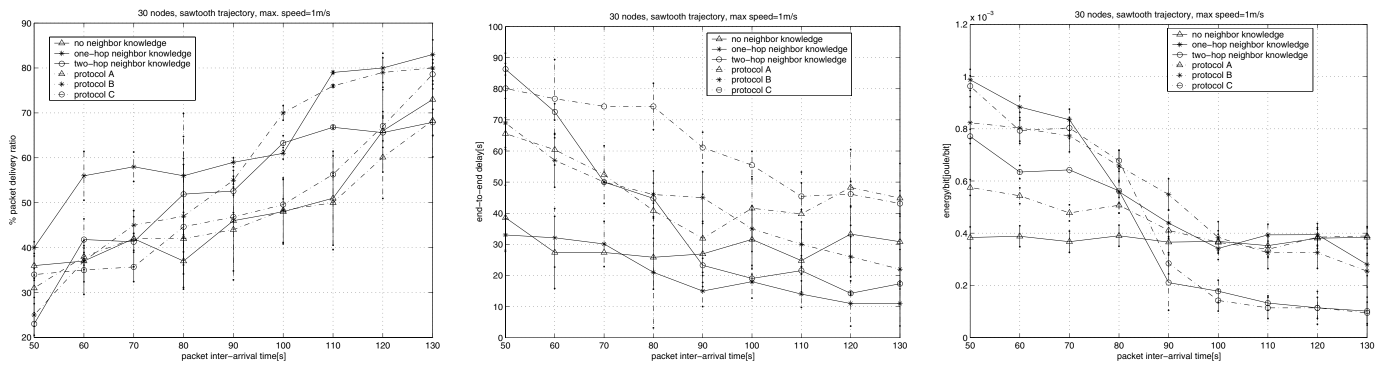
[11] R. J. Urick. *Principles of Underwater Sound*. McGraw-Hill, 1983.

[12] H. Zhai, X. Chen, and Y. Fang. Rate-based transport control for mobile ad hoc networks. In *Proc. of Wireless Communications and Networking Conference)*, volume 4, pages 2164–2169, Mar. 2005.



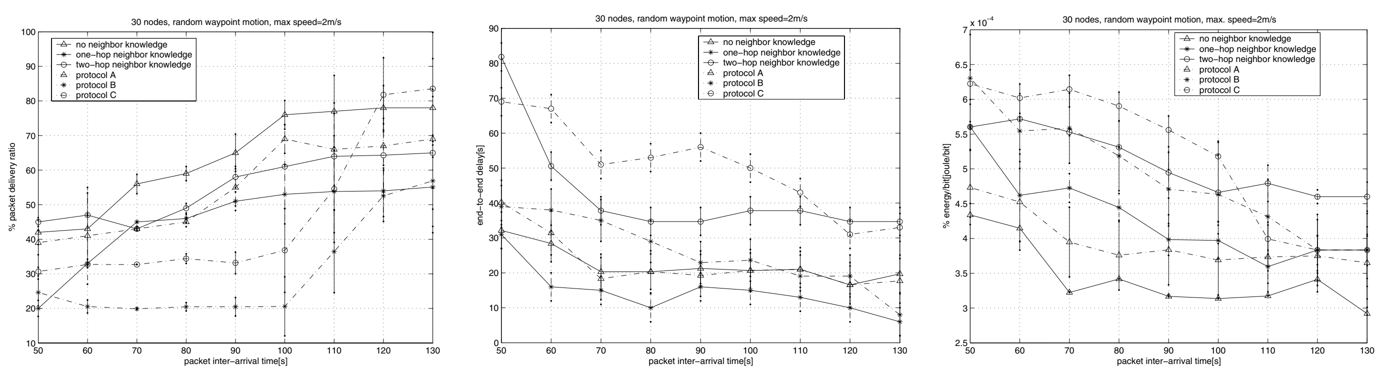
(a) (b) (c)

Figure 5: no mobility. (a): packet delivery ratio vs. packet inter-arrival time; (b): delay vs. packet inter-arrival time; (c): energy/bit vs. packet inter-arrival time.



(a) (b) (c)

Figure 6: mobility with gliders. (a): packet delivery ratio vs. packet inter-arrival time; (b): delay vs. packet inter-arrival time; (c): energy/bit vs. packet inter-arrival time.



(a) (b) (c)

Figure 7: mobility with AUVs. (a): packet delivery ratio vs. packet inter-arrival time; (b): delay vs. packet inter-arrival time; (c): energy/bit vs. packet inter-arrival time.