DOTS: A Propagation Delay-aware Opportunistic MAC Protocol for Underwater Sensor Networks

Youngtae Noh, Paul Wang[†], Uichin Lee[‡], Dustin Torres[§], Mario Gerla
UCLA, CS   [†]JPL, Caltech   [‡]KAIST, KSE   [§]UCLA, EE
{ytnoh, gerla}@cs.ucla.edu, [†]Paul.Wang@jpl.nasa.gov, [‡]uclee@kaist.edu, [§]dustintorres@ucla.edu

Abstract—Underwater Acoustic Sensor Networks (UW-ASNs) use acoustic links as a means of communications and are accordingly confronted with long propagation delays, low bandwidth, and high transmission power consumption. This unique situation, however, permits multiple packets to concurrently propagate in the underwater channel, which must be exploited in order to improve the overall throughput. To this end, we propose the Delay-aware Opportunistic Transmission Scheduling (DOTS) algorithm that uses passively obtained local information (i.e., neighboring nodes' propagation delay map and their expected transmission schedules) to increase the chances of concurrent transmissions while reducing the likelihood of collisions. Our extensive simulation results document that DOTS outperforms existing solutions and provides fair medium access.

Index Terms—Underwater, Medium Access Control, Opportunistic Transmission, CSMA

Fig. 1.   SEA Swarm Architecture

I. INTRODUCTION

Underwater Acoustic Sensor Networks (UW-ASNs) have recently been proposed as a way to explore and observe the ocean, which covers two-thirds of the Earth's surface [1], [2], [3]. In particular, we consider a SEA Swarm (Sensor Equipped Aquatic Swarm) architecture illustrated in Fig. 1 for short-term *ad hoc* real-time aquatic exploration such as oil and chemical spill monitoring, submarine detection, and surveillance. A swarm of drifting sensor nodes such as UCSD Drogues [4] is deployed to the venue of interest and moves as a group with the ocean current [5], [6]. Each sensor monitors local underwater activities and reports critical events using acoustic multi-hop routing to a distant data collection center, e.g., surface buoys or Autonomous Underwater Vehicles (AUVs) [7].

Despite the technological advances of acoustic communications, we are still confronted with limitations that need to be addressed in order for UW-ASNs to be put into practical use, namely severely limited bandwidth, long propagation delays ($1.5km/s$, five orders of magnitude slower than radio signals), and relatively high transmission energy cost (reception to transmission power ratio of 1:125 [8]). Moreover, the unreliable nature of underwater wireless channels due to complex multipath fading and surface scattering further aggravates data communications [9].

Under these circumstances, Medium Access Control (MAC) protocols designed for terrestrial packet radio networks cannot be directly used because the propagation delay of acoustic signals is much greater than the packet transmission time (e.g., $0.5sec$ vs. $0.04sec$ to transmit a $256bytes$ data packet with the data rate of 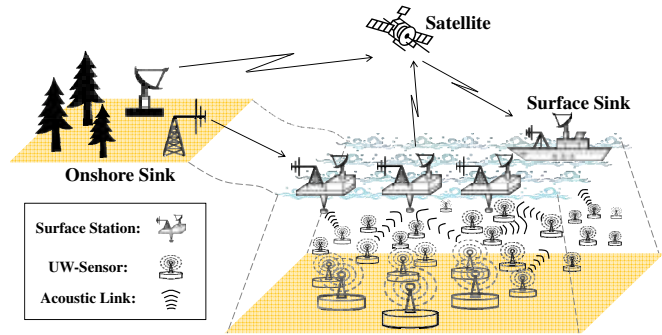$50kbps$ over a $750m$ range) — carrier sensing in Carrier Sense Multiple Access (CSMA) may not prevent packet collisions. This unique situation, however, permits multiple packets to concurrently propagate in an underwater channel, which must be exploited in order to improve the channel throughput. While this phenomenon is also observed in transatlantic wire lines or wireless satellite links, the main departure is that these are point-to-point links without any contention and that the large Bandwidth-Delay Product (BDP) is exploited at a higher layer, namely TCP. In general, long propagation latency in an underwater wireless network creates a unique opportunity for *temporal reuse* that allows for multiple concurrent packets propagating within the same contention domain. Note that temporal reuse is an additional opportunity on top of well-known *spatial reuse* in wireless networks which allows concurrent, non-colliding transmissions to different destinations if they are sufficiently removed from one another, solving the exposed terminal problem.

Recently a great deal of attention has been focused on exploiting temporal and/or spatial reuse of acoustic channels to improve the throughput. For instance, Slotted FAMA (S-FAMA) uses time slotting in order to lower the probability of collisions by aligning packet transmissions into slots (as in Slotted Aloha) while Propagation-delay-tolerant Collision Avoidance Protocol (PCAP) [10] allows a node to send multiple reservation requests for transmission time slots (i.e., request to transmit, RTS). In Underwater-FLASHR (UW-FLASHR) [11], time slots are divided into reservation and data transmission periods to realize efficient channel reservation and to minimize data packet losses caused by control packet exchanges. For better channel utilization, most protocols attempt to build a Time Division Multiple Access (TDMA) schedule using brute-force learning via repeated trial-and-errors [11] or solving computationally hard optimal scheduling

problems as in ST-MAC [12] and STUMP [13]. Distributed approximation algorithms for optimal scheduling were proposed in the literature [12], [13], but discovering a reasonable TDMA schedule requires a network-wide consensus, incurring a large number of packet exchanges and taking a considerable amount of time. In general, TDMA-based methods are not suitable for resource constrained underwater mobile sensor networks, because nodes must periodically perform expensive scheduling operations.

Nonetheless the key insights from TDMA-based scheduling methods allow us to enhance conventional CSMA-like random channel access protocols as follows. We need to ensure that transmissions are scheduled carefully such that they do not interfere with the reception of each others' packets by their intended receivers. To satisfy this requirement, each node must evaluate the collision conditions for neighboring packet receptions prior to transmitting a packet. Recall that a *collision* occurs when a receiver tries to decode a packet when more than one packet arrives from different senders simultaneously [14]. The key intuition is that each node can *predict* whether its upcoming packet transmission will collide with another's if it has the neighboring nodes' propagation delay information and their transmission schedules.

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

- DOTS can effectively exploit temporal and spatial reuse by using local information. In DOTS, each node learns neighboring nodes' propagation delay information and their expected transmission schedules by passively overhearing packet transmissions. Thus, DOTS can compensate for the long propagation latencies by increasing the chances of concurrent transmissions while reducing the likelihood of collisions. Our extensive simulation results confirm that DOTS can significantly improve the overall throughput. We also show that such opportunistic scheduling can effectively handle spatial-unfairness caused by physical location and propagation latency (i.e., the closer the distance between a pair of nodes, the higher the chance of capturing the channel [15]).
- One of the key assumptions of DOTS is clock synchronization, because nodes build local propagation delay maps by overhearing packets. Syed *et al.* proposed a protocol called Time Synchronization for High Latency (TSHL) and validated that TSHL can correct clock offset and skew in a reliable and efficient manner using simulations [16]. In this paper, we implement this protocol on the UANT platform that is composed of a software defined radio and a mix of custom and commercially available hardware for the acoustic transmitter and receiver [17]. We demonstrate that TSHL can effectively synchronize clock offset and skew. To the best of our knowledge, this is the first real implementation of its kind.
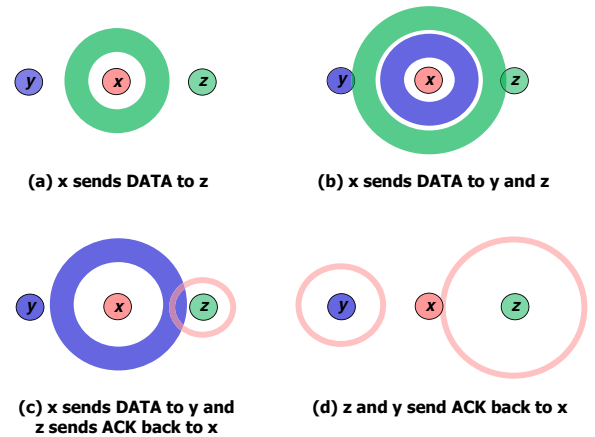

Fig. 2. Temporal Reuse

II. BACKGROUND AND RELATED WORK

In multi-hop wireless networks, it is important to efficiently utilize limited network resources and to provide fair access for competing data flows. It has been proven that CSMA provides reasonable performance and fairness [18]. Since CSMA does not require strict scheduling, it can support node mobility, which is also a major challenge in the SEA Swarm architecture (Fig. 1). However, the handshaking mechanism of CSMA leads to a severely degraded system throughput due to the presence of long propagation delay of acoustic signals in UW-ASNs, which is a well recognized problem. Moreover, carrier sensing may fail to detect an ongoing transmission due to the propagation delay, which impairs the performance of CSMA protocols [19].

A. Temporal Reuse

One potential solution for improving CSMA in UW-ASNs is to utilize *temporal reuse* that exploits the long propagation latencies of acoustic waves. Fig. 2 demonstrates the notion of temporal reuse. Node x sends a DATA packet to node z in Figure 2(a) and again at a later time another DATA packet to node y in Fig. 2(b). Node z sends an acknowledgment (ACK) back to node x as node y is about to receive the transmission from node x in Fig. 2(c). Finally, node y sends an ACK back to node x in Fig. 2(d). This case enables the data and ACKs to be transmitted and received without any collision.

To harness this temporal reuse, Yackoski *et al.* [11] proposed UW-FLASHR, a variant TDMA protocol that can achieve higher channel utilization than the maximum utilization possible in existing TDMA protocols. Hsu *et al.* [12] proposed ST-MAC, another underwater TDMA protocol that operates by constructing Spatial-Temporal Conflict Graph (ST-CG) to describe the conflict delays among transmission links and reduces the ST-CS model to a new vertex coloring problem. A heuristic, called the Traffic-based One-step Trial Approach (TOTA), is then proposed to solve the coloring problem. Kredo *et al.* [13] proposed a TDMA-like protocol called STUMP that uses propagation delay information and prioritizes conflicting packet transmissions based on certain metrics (e.g., random ordering and uplink delay ordering). However, TDMA scheduling is typically performed in a centralized

(a) Exclusive medium access: *exposed terminal*

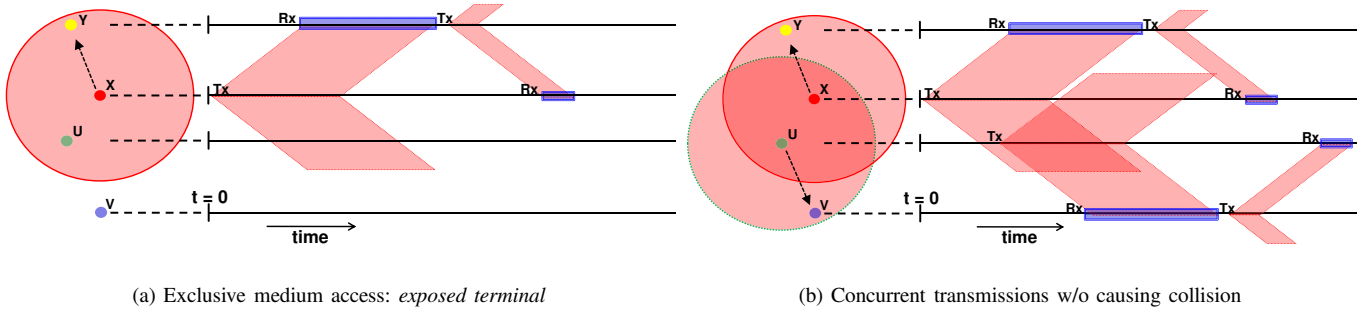(b) Concurrent transmissions w/o causing collision

Fig. 3. Spatial Reuse

way which is not resilient to failure; moreover, discovering a reasonable TDMA schedule using distributed algorithms for optimized transmission scheduling requires a network-wide consensus. TDMA-like protocols are not suitable for resource constrained mobile sensor networks.
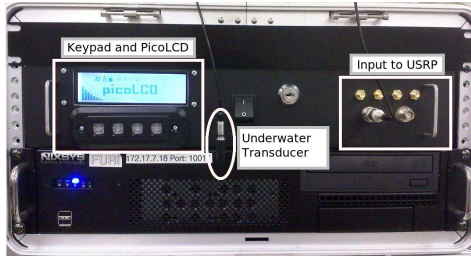
CSMA-like protocols (or reservation-based protocols) have been proposed to exploit temporal reuse in several ways. Given that channel reservation takes long time (i.e., RTS/CTS), Guo *et al.* proposed Adaptive Propagation-delay-tolerant Collision Avoidance Protocol (APCAP) that allows a node to transmit packets in out-of-order during this period (i.e., multiple reservations concurrently) [10], but it does not detail scheduling strategies for out-of-order packet delivery. To reduce the control overhead (e.g., reservation, acknowledgement), R-MAC [20] delivers a burst of packets (or a packet train) and delayed ACKs, thereby improving the channel throughput. Chen *et al.* proposed Ordered CSMA that transmits each data packet in a fixed order [21]. Given the fact that two sequential carriers traveling in the same direction will not collide, each station transmits immediately after receiving a data frame from the previous station sequentially, instead of waiting for a period of maximum propagation delay. Yet, ordered CSMA is not appropriate for large-scale multi-hop networks because generating collision free transmission order requires relative positions of all nodes in the network and a large number of packet exchanges. Chirdchoo et al. [22] proposed a receiver initiated reservation protocol called Receiver-Initiated Packet Train (RIPT) where after initiating packet transfers, the receiver accepts the packet transmission requests from its neighboring nodes and builds a transmission schedule for its neighboring nodes by considering the propagation delay to its neighbors. In RIPT, the receivers need to periodically initiate packet transfers, which is very expensive, and under varying traffic demands, it is non-trivial to determine when to initiate packet transmissions. Unlike existing underwater CSMA solutions, DOTS neither requires an additional phase for reservation scheduling nor restricts transmission schedules to a specific order. DOTS is a sender initiated protocol that relies solely on passively overhearing neighboring transmissions to make intelligent local decisions based upon its own transmission schedule that does not interfere with neighboring receptions, thereby aggressively exploiting temporal reuse.
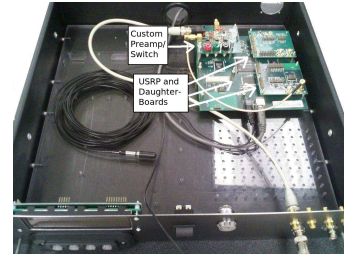
B. Spatial Reuse

Spatial reuse in UW-ASNs also improves the channel utilization by allowing concurrent transmissions. In Fig. 3(a), a network topology consisting of four nodes is depicted and its corresponding signal propagation in time is drawn on the side. Node x gains the exclusive access of the channel in its collision domain, preventing node u from transmitting to node v, since node u's transmission will interfere with node x's reception of an ACK from node y, known as the *exposed terminal problem*. However, Fig. 3(b) shows that it is still possible for node u to transmit concurrently without affecting x's transmission, enabling spatial reuse of the medium. While spatial reuse is well-investigated in terrestrial wireless communications [23], [24], [25], [26], [27], to the best of our knowledge, none of the existing work has been done in UW-ASNs. Henceforth, we follow with a short discussion of related works in terrestrial networks and its applicability to UW-ASNs.

MACA-P [28] detects an expose terminal from *Request-To-Send/Clear-To-Send* (RTS/CTS) exchanges such that a node overhears an RTS without overhearing the corresponding CTS. MACA-P introduces a control gap (or delay) between RTS/CTS and DATA/ACK to allow neighboring nodes to schedule their transmissions (via explicit RTS/CTS). Given that control gap incurs an extra overhead, Shukla *et al.* proposed to use direct data transmissions without RTS/CTS during the exposed period [26]. Alternative method to this approach is to build local conflict maps by empirically detecting expose link pairs via off-line methods (e.g., broadcast collision based interference estimation as in RTSS/CTSS [24]), or on-line methods (e.g., unicast collision based interference estimation as in CMAP [29]). In RTSS/CTSS, nodes coordinate simultaneous transmissions using new control messages, namely the Request-To-Send-Simultaneously (RTSS) and the Clear-To-Send-Simultaneously (CTSS), whereas in CMAP, nodes monitor neighboring nodes' transmissions to opportunistically schedule simultaneous transmissions.

In underwater acoustic networks, we note that building such conflict maps is very expensive (due to power-hungry packet transmissions and mobility of sensor nodes), and moreover, they fail to take the large propagation delay into account. In this paper, we propose to use delay maps which can be built by passively observing transmissions. Given such delay maps can be used to predict potential collisions, our approach

(a) Front view of UANT system

(b) Internal view of UANT system

Fig. 4.   A deployable UANT system

utilizes delay maps to opportunistically schedule simultaneous transmissions (which can be done without extra control packet exchanges). Note that since we only use delay maps, our approach cannot exploit the capture effect where a receiver can correctly decode a packet even in the presence of other concurrent transmissions. Yet, our approach can be extended to exploit the capture effect as in Interference Aware (IA) MAC [26], which is the part of our future work.

III. DOTS PREREQUISITE

It has been shown that observed information obtained from passively overhearing neighboring transmissions can be useful in estimating collisions at the intended receivers [30]. DOTS uses the passively obtained information by building a *delay map* to achieve both temporal and spatial reuse by making intelligent transmission scheduling decisions. DOTS therefore is able to compensate for the long propagation latencies and severely limited bandwidth of the acoustic medium by using passively observed information to increase the chances of concurrent transmissions while reducing the likelihood of collisions. However, the lack of clock synchronization could make it difficult for an overhearing node of a transmission to gauge the propagation delay between itself and the transmitting node. Thus, the DOTS protocol makes the assumption of time synchronization amongst all nodes in the network, similar to existing underwater CSMA solutions proposed in [31], [32], [15]. This assumption is necessary in order to accurately enable estimation of the transmission delay between nodes in a passively promiscuous mechanism.

Syed *et al.* showed that clock offset and skew can be corrected in a reliable and efficient manner to achieve time synchronization for underwater acoustic networks using the Time Synchronization for High Latency (TSHL) protocol [16]. Using this protocol a leading transmitter will send out multiple time-stamped beacons. All receiving nodes will calculate the difference between the received timestamp and the local time, compute a linear regression over all these values, and find the slope of the line. Finally in the second phase offset is found using the skew compensated time. We have implemented this protocol on the UANT platform (see Fig. 4), which uses a software defined radio and a mix of custom and commercially available hardware for the transmitter and receiver [17]. Fig. 5 shows that after enough beacons are sent the skew between
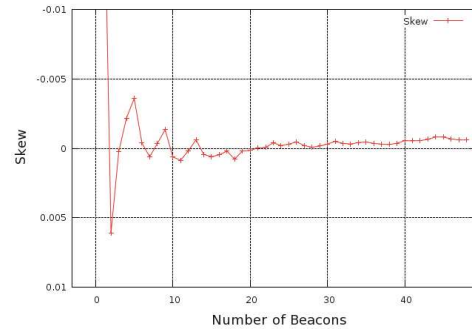

Fig. 5.   Number of beacons used in TSHL vs. skew estimate

nodes converges, and the nodes share the same notion of time.

Due to clock drift that appears in all oscillators, even after nodes have been synchronized, their clocks will eventually drift apart. This fact leads to the need for periodic resynchronization. The rate at which a synchronization protocol should run largely depends on the properties of the crystals used in oscillators. While inexpensive oscillators tend to have a drift of 30-50 parts per million (ppm), many underwater ranging solutions use more precise clocks (that are temperature compensated) and can achieve accuracies of less than 1 ppm [33]. Two nodes with 50 ppm clocks can accumulate a maximum error of 50ms in approximately 8.3 minutes, while the clock used by Eustice *et al.* [33] will accumulate 2ms of error in just under 14 hours. Therefore, depending on the nodes' hardware, the required resynchronization rate can vary dramatically, but it is still feasible with limited overhead.

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

IV. DOTS DESIGN

We now describe our underwater transmission scheduling algorithm, DOTS that exploits long propagation delays by using passively observed one-hop neighboring nodes' transmissions to improve channel utilization. The design of DOTS

is based on MACA-like random channel access with RTS/CTS. Because of this design choice, it is confronted with the problem that data transmission between two nearby nodes after RTS/CTS handshaking can be collided with RTS control frames of a distant node due to relatively long propagation delays [34]. Recall that this will happen more frequently and be more expensive in underwater acoustic networks than in terrestrial radio networks due to the high latency and transmission costs. Fullmer *et al.* [35] identified the problem and provided the following two conditions for collision free transmission:

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

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

Whenever a node has a frame to send, it runs a transmission scheduling decision algorithm based on its delay map database to make a decision as to whether or not to begin its transmission, which will be further discussed in Section IV-B. If no conflicts are detected, it begins its transmission; otherwise, it backs off for a random amount of time. It is important to note that unlike traditional CSMA-like protocols, DOTS allows each node to have multiple outstanding packets to receive. Since each node may miss a neighbor's RTS or CTS transmission due to channel fading in underwater, conflict detection schedules may still cause collisions. Thus, to reduce the damage and to avoid deadlock, DOTS provides for a recovery scheme, the details of which will be discussed in Section IV-C. Finally, since deployed nodes are moving along with the ocean current, it requires a guard time to avoid invalid transmission scheduling caused by the node mobility, which will be further discussed in Section IV-D.

A. Delay Map Management

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

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

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

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

B. Transmission Scheduling

Based on the delay map, a node decides whether or not it can transmit without interfering with a neighbor's reception. Fig. 6 provides an example of the transmission scheduling decision process. Node x sends an RTS to node y. When node u receives this RTS and has data to send, it can begin its own transmission to node v concurrently if the following two conditions hold:

- *Neighboring non-interference*: Its current transmission (RTS) and future transmission (DATA) must not interfere with neighbors' ongoing and prospective receptions.
- *Prospective non-interference*: Its future receptions (CTS and ACK) must not be interfered with by neighbors' prospective transmissions.
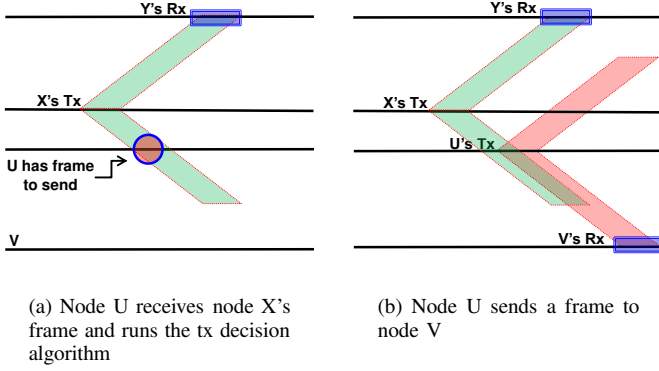
(a) Node U receives node X's frame and runs the tx decision algorithm

(b) Node U sends a frame to node V

Fig. 6.   Example of a transmission decision

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

$$r_{CTS(y)} = t_{RTS(x)} + \Delta_{CtlProp} + \tau_{CTS(y)} + \Delta_{y \to x} \quad (1)$$

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

$$\Delta_{CTS(y)} = [r_{CTS(y)}, \ r_{CTS(y)} + \frac{\ell_{CTS}}{\lambda_{DATA}}] \quad (2)$$

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

$$r_{ACK(y)} = r_{CTS(y)} + \Delta_{DataProp} + \tau_{DATA(x)} + \tau_{ACK(y)} + \Delta_{y \to x} \quad (3)$$

where $\Delta_{DataProp}$ denote the sum of the maximum propagation delay between two nodes and the reception duration for the DATA frame, $\tau_{DATA(x)}$ denotes DATA frame processing time, and $\tau_{ACK(y)}$ denotes ACK frame processing time. Reception duration of y's ACK at node x can be calculated as below:

$$\Delta_{ACK(y)} = [r_{ACK(y)}, \ r_{ACK(y)} + \frac{\ell_{ACK}}{\lambda_{DATA}}] \quad (4)$$

Finally, node u makes a decision to launch its RTS transmission when its *current time + delay from node u to x* is not in the time ranges of (2) and (4).

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

$$r_{CTS(v)} = t_{RTS(u)} + \Delta_{CtlProp} + \tau_{CTS(v)} + \Delta_{v \to u} \quad (5)$$

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

$$\Delta_{CTS(v)} = [r_{CTS(v)}, \ r_{CTS(v)} + \frac{\ell_{CTS}}{\lambda_{DATA}}] \quad (6)$$

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

$$r_{ACK(v)} = r_{CTS(v)} + \tau_{DATA(u)} + \Delta_{DataProp} + \tau_{ACK(v)} + \Delta_{v \to u} \quad (7)$$

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

$$\Delta_{ACK(u)} = [r_{ACK(v)}, \ r_{ACK(v)} + \frac{\ell_{ACK}}{\lambda_{DATA}}] \quad (8)$$

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

$$r_{DATA(x)} = t_{RTS(x)} + 2 \times \Delta_{CtlProp} + \tau_{CTS(y)} + \tau_{DATA(x)} + \Delta_{x \to u} \quad (9)$$

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

$$\Delta_{DATA(x)} = [r_{DATA(x)}, \ r_{DATA(x)} + \frac{\ell_{DATA}}{\lambda_{DATA}}] \quad (10)$$

Now, node u makes a decision to launch its RTS transmission when the time ranges of (6) and (8) is not in (10). Algorithm 1[1] provides a simplified general description of the transmission decision algorithm for the *neighboring non-interference* case. The algorithm for the *prospective non-interference* case can be implemented like the same way of Algorithm 1 based on (6) and (8).

C. Schedule Recovery

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

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

[1]To simplify the pseudo code, $[arrival_{CTS/DATA/ACK} + $ reception duration of CTS/DATA/ACK $\pm t_{GUARD}]$ is abbreviated to $[arrival_{CTS/DATA/ACK} \pm t_{GUARD}]$.

Algorithm 1 Transmission Scheduling Algorithm

1: **procedure** neighboring non-interference($Message\ m$)
2:   **for all** $e \in$ delay map entries **do**
3:     $arrival_{m \to e.src} \leftarrow prop_{delay} + tx_{delay}$
4:     **if** $e.frame\_type == RTS$ **then**
5:       $arrival_{CTS} \leftarrow e.timestamp + delay_{e.src \to e.dest} +$ $prop_{ctrl\_delay}$
6:       **if** $arrival_{m \to e.src} \in [arrival_{CTS} \pm t_{GUARD}]$ **then**
7:         **return** collision detected
8:       **end if**
9:       $arrival_{ACK} \leftarrow e.timestamp + delay_{e.src \to e.dest} +$ $(2 \times prop_{ctrl\_delay}) + prop_{data\_delay}$
10:       **if** $arrival_{m \to e.src} \in [arrival_{ACK} \pm t_{GUARD}]$ **then**
11:         **return** collision detected
12:       **end if**
13:     **else if** $e.frame\_type == CTS$ **then**
14:       $arrival_{DATA} \leftarrow e.timestamp + delay_{e.src \to e.dest} +$ $prop_{ctrl\_delay}$
15:       **if** $arrival_{m \to e.src} \in [arrival_{DATA} \pm t_{GUARD}]$ **then**
16:         **return** collision detected
17:       **end if**
18:     **else if** $e.frame\_type == DATA$ **then**
19:       $arrival_{ACK} \leftarrow e.timestamp + delay_{e.src \to e.dest} +$ $prop_{data\_delay}$
20:       **if** $arriaval_{m \to e.src} \in [arrival_{ACK} \pm t_{GUARD}]$ **then**
21:         **return** collision detected
22:       **end if**
23:     **else if** $e.frame\_type == ACK$ **then**
24:       no check necessary, continue processing
25:     **end if**
26:   **end for**
27:   **return** no collision detected
28: **end procedure**

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

D. Guard Time

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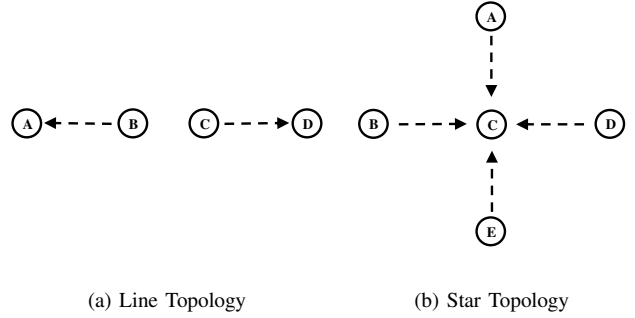

(a) Line Topology   (b) Star Topology

Fig. 7. Simulation Topologies

V. SIMULATION & EVALUATION

A. Simulation setup

1) Simulation parameters: For acoustic communications, the channel model described in [36] and [37] is implemented in the physical layer of QualNet. The path loss over a distance d for a signal of frequency f due to large scale fading is given as $A(d, f) = d^k a(f)^d$ where k is the spreading factor and $a(f)$ is the absorption coefficient. The geometry of propagation is described using the spreading factor ($1 \leq k \leq 2$); for a practical scenario, k is given as 1.5. The absorption coefficient $a(f)$ is described by the Thorp's formula [37]. As in [36], [38], we use Rayleigh fading to model small scale fading. Unless otherwise mentioned, the data rate is set to 50*kbps* as in [39]. We vary data size from 512*bytes* to 1*kbytes* to observe behavior of each protocol in terms of varying data size. Note again that at a data rate of 50*kbps* a 1*kbyte* frame requires 0.16384*sec* to transmit and the one-way trip delay on a 750*km* link is approximately 0.5*sec* ($\gg$ tx duration = 0.16384*sec*) considering acoustic propagation delay and transmission duration. We measure throughput and energy consumption per node as the functions of the offered load on the sensor network. The load is varied between generating a single frame every 30*sec* down to a single frame every 0.25*sec*. In our simulation, each run lasts 1*hour*. Unless otherwise specified, we report the average value of 50 runs with the 95% confidence interval.

2) Topology: As shown in Fig. 7, we deployed the nodes in a line topology and a star topology in a 3D region of $5km \times 5km \times 5km$. In the line topology depicted in Fig. 7(a), four nodes are deployed in a line and with a fixed distance between one-hop neighbors. The distance between the nodes is fixed to 750*m* for the experiments, and thus the two nodes, B and C, are exposed to each other. We adopt this line topology to show how spatial reuse affects system throughput. The star topology depicted in Fig. 7(b) shows a more aggressive traffic toward the center node (c) since the four surrounding nodes attempt to simultaneously send their data to the center node. In this scenario, we create a high contention situation between the four outer nodes for the center node. The distance between the center node and the four surrounding nodes is fixed to 750*m* over our experiments. Here, increasing number of senders to the center node will attest to the benefits of temporal reuse in the presence of high contention.
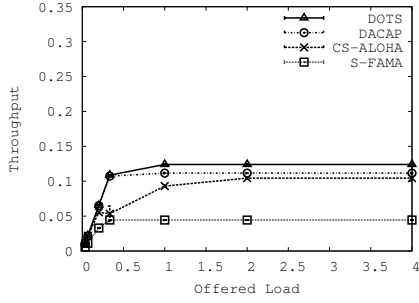
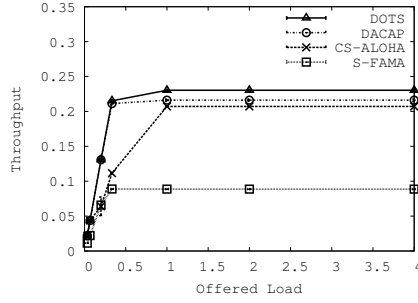Fig. 8. Line topology: throughput as a function of offered load with fixed data size (512*bytes*)

Fig. 9. Line topology: throughput as a function of offered load with fixed data size (1*kbytes*)
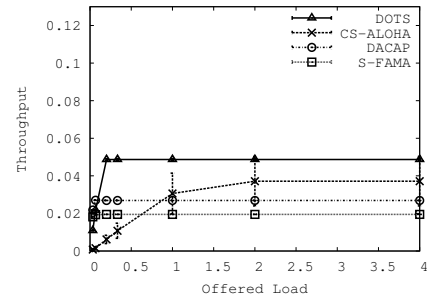

Fig. 10. Star topology: throughput as a function of offered load with fixed data size (512*bytes*)
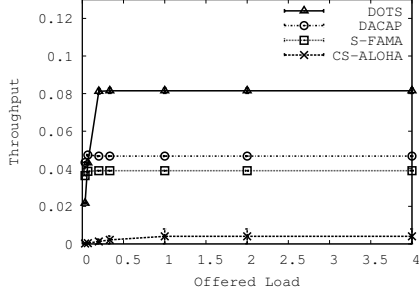

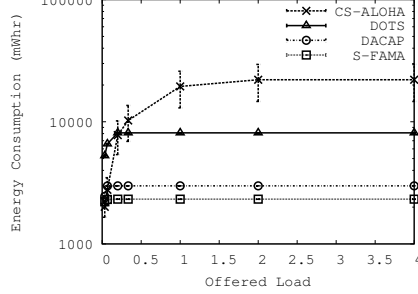Fig. 11. Star topology: throughput as a function of offered load with fixed data size (1*kbytes*)

Fig. 12. Star topology: energy Consumption in the star topology with fixed data size (512*bytes*)
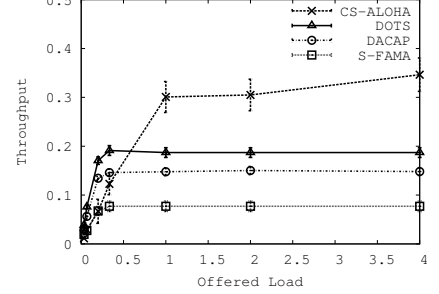

Fig. 13. Random topology: throughput as a function of offered load with MCM mobility

We also randomly deploy 10 nodes in a 3D region of $430m \times 430m \times 430m$ with a transmission range of $750m$ to test node mobility to support the SEA Swarm architecture. This region enables all deployed nodes to be fully connected and exposed to high levels of channel contention as in [15], [32]. We adopt an extended 3D version of the Meandering Current Mobility (MCM) Model [7] to model the motility of each sensor node. Unlike most existing sensor node mobility patterns from literature which assumes that each node moves independently of all others, wherein its path vector is determined from an independent realization of a stochastic process, the MCM model considers fluid dynamics whereby the same velocity field advects all nodes. Here, the MCM model considers the effect of meandering sub-surface currents (or jet streams) and vortices on the deployed nodes to pattern its path vector. In our simulations, we restrict the nodes move with a maximum speed of $0.3m/s$ with the MCM model to test the resiliency of the guard time in DOTS.

B. Simulation results

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

$$\text{Throughput} = \frac{\text{\# of rx } data \text{ frames} \times \Delta_{data}}{\text{Simulation Duration}} \quad (11)$$

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

$$\text{Offered Load} = \frac{\text{\# of generated } data \text{ frames} \times \Delta_{data}}{\text{Simulation Duration}} \quad (12)$$

The performance of DOTS was compared to that of three CSMA protocols, namely Slotted FAMA (S-FAMA) [34], DACAP [40], and CS-ALOHA with ACK [41]. S-FAMA is a synchronized underwater MAC protocol based on RTS/CTS handshaking. The main idea of S-FAMA is to time slot exclusive access to the channel medium so that the time duration of each slot is long enough to ensure that any frame transmitted at the start of the slot will reach the destination before the slot duration ends. DACAP is a non-synchronized protocol that allows each node to use different handshaking lengths for different distances between the sender and the receiver. To reduce collision, DACAP follows these two collision avoidance conditions: 1) when a receiver overhears an RTS threatening its pending data reception, the receiver sends a very short warning frame to its intended sender to defer its data transmission until the predefined waiting period 2) after sending an RTS, if a sender overhears a CTS threatening the neighbor's pending data reception, it defers its data transmission. CS-ALOHA with ACK is ALOHA adapted for the underwater environment, where each node transmits whenever the channel is idle without performing the RTS/CTS handshaking process.

Fig. 8 and 9 show the throughput of the four protocols with different data sizes in the line topology (exposed terminal). As shown in Fig. 8 and 9, DOTS outperforms S-FAMA by a factor of two and DACAP and CS-ALOHA by around 15% for a $750m$ transmission range with both $512bytes$ and $1024bytes$ data frame sizes. It is noteworthy that DACAP outperforms S-FAMA by two times because DACAP allows for concurrent transmissions of the two sender-receiver pairs in Fig. 7(a); when a sender-receiver pair (A-B) is undergoing data transmission in the line topology, the other pair (C-D) can also perform parallel data transmission because the two collision avoidance conditions of DACAP cannot suppress the transmissions of the two sender nodes (B and C). Consequently, this

allows DACAP to perform concurrent transmissions possibly with collisions; however, it is the result of avoiding these minor collisions which explains the utilization gain of DOTS over that of DACAP. By varying the data size, Fig. 8 and 9 show that data size is proportional to the increase in throughput of all handshaking based protocols. The throughput of CS-ALOHA shows similar throughput performance against DOTS. Although it takes advantage of spatial reuse, it lacks the capability to avoid collisions, thereby offsetting the gains from spatial reuse, which will be addressed in the star topology. It is also interesting to note that the all four protocols show a saturation point. The throughput increases as the offered load increases until a threshold limit. After reaching the threshold point, the all four protocols suppress their transmissions and thus their performance becomes saturated.

In the star topology, the four outer nodes compete to send their frames to the one center node. Fig. 10 and 11 show that DOTS outperforms S-FAMA and CS-ALOHA by two times and DACAP by 70% for a $750m$ transmission range with both $512bytes$ and $1024bytes$ data frame sizes. By varying the data size, these two figures show that the three handshaking based protocols exhibit the behavior that throughput is proportional to data frame size. On the other hand, CS-ALOHA shows unstable throughput performance; when the data size exceeds a threshold, CS-ALOHA significantly increases its collision rate and reduces its overall throughput. In contrast, DOTS shows a vastly superior behavior. As the number of senders increases, DOTS can better exploit temporal reuse. In this star topology, DOTS outperforms S-FAMA by 2 times and DACAP by 70%. Inversely, CS-ALOHA provides the worst throughput due to absence of collision avoidance.

2) Energy consumption: Fig. 12, which represents the four throughput lines of the protocols in Fig. 10, shows the average power consumption of the four protocols in the star topology with a $750m$ transmission range and $1024bytes$ data frame size. It shows the average energy consumption of each protocol per node during the entire simulation. When it is compared with the throughput lines of the four protocols in Fig. 10, it implicitly indicates that the number of collisions which occur in each protocol. DOTS consumes more energy than S-FAMA and DACAP because it delivers, by far, more frames than these two protocols. Inversely, throughput for CS-ALOHA about 20% lower than that of DOTS, yet the energy consumption of CS-ALOHA is several times higher illustrating that CS-ALOHA consumes significantly more energy due to collisions.

3) Mobility: The effect of random topologies and node mobility are examined in Fig. 13. Ten nodes are randomly deployed to a region which enables full connectivity between all nodes, whereby each node follows a jet stream path vector based on the MCM model. The main jet stream speed of each node is capped at $0.3m/s$ with each node having a $750m$ transmission range. Five pairs of sender-receiver nodes are actively engaged in data communication, transmitting $512byte$ data packets. Note that with a $0.3m/s$ jet stream, nodes can move approximately $20m$ in 60 $seconds$, henceforth a $20ms$ guard time is amply chosen for use in DOTS to allow for approximately up to a $30m$ variation of node locality.
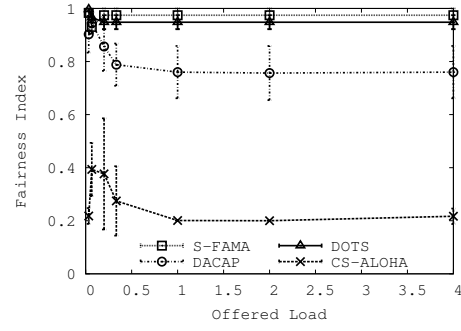

Fig. 14. Jain's Fairness Index for the four protocols

Fig. 13 shows that DOTS outperforms DACAP by 30% and S-FAMA by 3 times. With a random topology and node mobility, DOTS clearly provides reliable throughput and performance gains over DACAP and S-FAMA by utilizing smart and adaptive scheduling techniques to harness temporal and spatial reuse. On the other hand, CS-ALOHA shows the best performance in the random topologies with node mobility for our test parameters, however, this comes at a steep price in terms of energy efficiency and fairness, which will be addressed in the following section.

4) Fairness: MAC protocols with backoff schemes (i.e., binary exponential) based on insufficient information about the network congestion may cause *spatial unfairness*, a form of channel capture, as described in [15]. Since a frame's propagation latency is proportional to the distance from a sender, the channel clears earlier for nodes closer to the sender. Closer nodes consequently have more opportunities to recapture the channel, resulting in unfairness amongst the nodes. To characterize the fairness, we use the Jain Fairness Index [42], defined as below:

$$\text{Fairness Index} = \frac{(\sum x_i)^2}{(n \cdot \sum x_i^2)} \qquad (13)$$

where x_i denotes the throughput of node i and n denotes the number of nodes in the network. Fig. 14, which is the corresponding fairness plot to Fig. 13, shows that S-FAMA and DOTS exhibit a high fairness index (0.9 and above) and also remain stable and constant with increased offered load. As described in IV-C, when more than one transmission schedule contends in a node, DOTS uses the timestamp knowledge in its delay map database to give preference to one of the transmission schedules. DOTS with random backoff exhibits high fairness for this reason. The reason for the slightly lower fairness of DOTS compared to S-FAMA is due to the use of temporal and spatial reuse. In DOTS, every sender-receiver pair has a fair chance of accessing the medium as in S-FAMA, yet some pairs are given the chance of concurrently accessing the medium, thus slightly affecting the fairness index. DACAP provides a lower fairness index than both S-FAMA and DOTS. This is because DACAP gives priority to the nodes already accessing the channel and consequently causes this bias. CS-ALOHA shows the lowest fairness index and the largest variation. Due to CS-ALOHA's binary exponential

backoff, it allows close sender-receiver pairs to potentially capture the channel, thereby severely degrading the fairness but providing best throughput performance as indicated in Fig. 13. This channel capturing also leads to severe data collisions at other nodes which have not captured the channel, inducing poor energy utilization. Furthermore, as Fig. 13 indicates CS-ALOHA is subject to far greater amounts of instability and throughput variation as a result of this capture effect.

VI. CONCLUSIONS

We have proposed a MAC protocol called DOTS that alleviates limitations caused by the long propagation latency and the severely limited bandwidth of acoustic communications. DOTS achieves better channel utilization by harnessing both temporal and spatial reuse. Extensive simulation results have shown that (1) DOTS outperforms S-FAMA by 2 times and DACAP by 15% times in the line topology (exposed terminal) and S-FAMA by 2 times and DACAP by 70% in the star topology (higher node density and contention), and (2) DOTS provides reliable throughput performance even with node mobility and preserves a high level of fairness for channel access.

There are several directions for future work. First, DOTS can better harness spatial/temporal reuse when we allow out-of-order packet delivery and packet trains at the sender side; yet, this improved efficiency comes at the cost of degrading fairness. Second, we will consider the capture effect as in Interference Aware (IA) MAC [26] where a receiver can correctly decode a packet even in the presence of other concurrent transmissions. Third, when a data frame is correctly received but the corresponding ACK gets lost due to lossy channel or collision, Windowed ACK [29] can help contain the number of spurious retransmissions and increase the throughput. Fourth, the impact of mobility and random topologies on the throughput and fairness will be carefully investigated. Finally, we plan to implement DOTS in a real world testbed to reexamine and verify our simulation results.

REFERENCES

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *in Ad Hoc Networks (Elsevier)*, 2005.
[2] J. Kong, J. H. Cui, D. Wu, and M. Gerla, "Building Underwater Ad-hoc Networks and Sensor Networks for Large Scale Real-time Aquatic Applications," in *MILCOM*, 2005.
[3] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data Collection, Storage, and Retrieval with an Underwater Network," in *Sensys*, 2005.
[4] J. Jaffe and C. Schurgers, "Sensor networks of freely drifting autonomous underwater explorers," in *WUWNet*, 2006.
[5] U. Lee, J. Kong, J. S. Park, E. Magistretti, and M. Gerla, "Time-Critical Underwater Sensor Diffusion with No Proactive Exchanges and Negligible Reactive Floods," in *ISCC*, 2006.
[6] Z. Zhou, J. Cui, and A. Bagtzoglou, "Scalable Localization with Mobility Prediction for Underwater Sensor Networks," in *Infocom*.
[7] A. Caruso, F. Paparella, L. Vieira, M. Erol, and M. Gerla, "The Meandering Current Mobility Model and its Impact on Underwater Mobile Sensor Networks," in *Infocom*, 2008.
[8] "WHOI, *Micro-Modem Overview*, Woods Hole Oceanographic Institution," http://acommn.whoi.edu/micromodem.
[9] R. J. Urick, "Principles of Underwater Sound 3rd Edition," *McGraw-Hill*, 1983.
[10] X. Guo, M. Frater, and M. Ryan, "A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks," in *OCEANS*, 2006.
[11] J. Yackoski and C.-C. Shen, "UW-FLASHR: Achieving High Channel Utilization in a Time-based Acoustic MAC Protocol," in *WUWNet*, 2008.
[12] C. Hsu, K. Lai, C. Chou, and K. C. Lin, "ST-MAC: Spatial-Temporal MAC Scheduling for Underwater Sensor Networks," in *InfoCom*, 2008.
[13] K. Kredo, P. Djukic, and P. Mohapatra, "STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol," in *Infocom*, 2009.
[14] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," in *SIGCOMM*, 1994.
[15] A. A. Syed, W. Ye, and J. Heidemann, "T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks," in *Infocom*, 2008.
[16] A. A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks," in *Infocom*, 2006.
[17] D. Torres, J. Friedman, T. Schmid, and M. B. Srivastava, "Software-defined underwater acoustic networking platform," in *WUWNet*, 2009.
[18] D. Bertsekas and R. Gallager, "Data Networks," NJ: *Prentice Hall*, 1992.
[19] D. Pompili, T. Melodia, and I. F. Akyildiz, "A CDMA-based Medium Access Control Protocol for Underwater Acoustic Sensor Networks," *IEEE Transactions on Wireless Commn*, 2009.
[20] P. Xie and J. H. Cui, "R-MAC An Energy-Efficient MAC Protocol for Underwater Sensor Networks," in *WASA*, 2007.
[21] M. Chen, S. Gonzalez, and V. Leung, "Applications and design issues for mobile agents in wireless sensor networks," *Wireless Communications, IEEE*, 2007.
[22] N. Chirdchoo, W. seng Soh, and K. Chua, "RIPT: A Receiver-Initiated Reservation-Based Protocol for Underwater Acoustic Networks," *Selected Areas in Communications, IEEE Journal on*, 2008.
[23] A. Acharya, A. Misra, and S. Bansal, "Design and analysis of a cooperative medium access scheme for wireless mesh networks," in *BroadNets*, 2004.
[24] K. Mittal and E. Belding, "RTSS/CTSS: mitigation of exposed terminals in static 802.11-based mesh networks," in *WiMesh*, 2006.
[25] M. Cesana, D. Maniezzo, P. Bergamo, and M. Gerla, "Interference aware (IA) MAC: an enhancement to IEEE802.11b DCF," in *VTC*, 2003.
[26] D. Shukla, L. Chandran-Wadia, and S. Iyer, "Mitigating the exposed node problem in IEEE 802.11 ad hoc networks," in *ICCCN*, 2002.
[27] S. Eisenman and A. Campbell, "E-CSMA: Supporting Enhanced CSMA Performance in Experimental Sensor Networks Using Per-Neighbor Transmission Probability Thresholds," in *Infocom*, 2007.
[28] A. Acharya, A. Misra, and S. Bansal, "Maca-p: A mac for concurrent transmissions in multi-hop wireless networks," *PerCom*, 2003.
[29] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing Exposed Terminals in Wireless Networks," in *NSDI*, 2008.
[30] L. G. Roberts, "ALOHA packet system with and without slots and capture," *SIGCOMM Comput. Commun. Rev.*, 1975.
[31] M. Molins and M. Stojanovic, "Slotted FAMA: A MAC Protocol for Underwater Acoustic Networks," in *OCEANS*, 2006.
[32] A. Syed, W. Ye, and J. Heidemann, "Comparison and Evaluation of the T-Lohi MAC for Underwater Acoustic Sensor Networks," *Selected Areas in Communications, IEEE Journal on*, 2008.
[33] R. Eustice, L. Whitcomb, H. Singh, and M. Grund, "Recent advances in synchronous-clock one-way-travel-time acoustic navigation," *OCEANS*, 2006.
[34] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," in *SIGCOMM*, 1995.
[35] ——, "Solutions to Hidden Terminal Problems in Wireless Networks," in *SIGCOMM*, 1997.
[36] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *SIGMOBILE Mob. Comput. Commun. Rev.*, 2007.
[37] P. L. L.M. Brekhovskikh, Yu, *Fundamentals of Ocean Acoustics 3rd Edition*. Springer, 2003.
[38] C. Carbonelli and U. Mitra, "Cooperative multihop communication for underwater acoustic networks," in *WUWNet*, 2006.
[39] Z. Zhou and J.-H. Cui, "Energy efficient multi-path communication for time-critical applications in underwater sensor networks," in *MobiHoc*, 2008.
[40] B. Peleato and M. Stojanovic, "Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks," *Communications Letters, IEEE*, 2007.
[41] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Analysis of Aloha Protocols for Underwater Acoustic Sensor Networks," in *WUWNet*, 2006.
[42] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research, Tech. Rep., 1984.