

# Opportunistic transmission scheduling using medium access control protocol

Arockia Rogini.A<sup>1</sup>, Arumai Ruban.J<sup>2</sup>

Student, Computer Science and Engineering, St. Joseph's College of Engineering and Technology, Thanjavur, India<sup>1</sup>

Asst.Prof., Computer Science and Engineering, St. Joseph's College of Engineering and Technology, Thanjavur, India<sup>2</sup>

**Abstract:** Mobile underwater networks with acoustic communications are confronted with several unique challenges such as long propagation delays, high transmission power consumption, and node mobility. In particular, slow signal propagation permits multiple packets to concurrently travel in the underwater channel, which must be exploited to improve the overall throughput. To this end, we propose the delay-aware opportunistic transmission scheduling (DOTS) protocol that uses passively obtained local information (i.e., neighbouring 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 even with node mobility.

**Key Words:** Underwater, AUV, medium access control, opportunistic transmission, CSMA

## I. INTRODUCTION

Mobile underwater sensor networks have recently been proposed as a way to explore and observe the ocean with wide area coverage at reasonable cost when compared to traditional tethered approaches (e.g., seabed sensors) [1], [2], [3], [4]. Towards this goal, a swarm of mobile sensors, e.g., autonomous underwater vehicles (AUVs) such as REMUS and IVER2 or floats such as UCSD Drogues [5], can be deployed to the venue of interest for short-term ad hoc real-time aquatic missions such as oil and chemical spill monitoring, submarine detection, and surveillance [6], [7]. Mobile nodes monitor local underwater activities and report collected sensor data using acoustic multi-hop routing to other mobile nodes for collaboration or simply to a distant data collection center.

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 delay (1.5 km/s, five orders of magnitude slower than radio signals), and relatively high transmission power (e.g., more than 100-fold more power consumption than reception [8], [9]). Moreover, the unreliable nature of underwater wireless channels due to complex multipath fading and surface scattering further aggravates data communications [10]. 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., 0.5 sec versus 0.04 sec to transmit a 256 byte data packet with the data rate of 50 kbps over a 750 m 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 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) [11] allows nodes to send multiple reservation requests for transmission time slots (i.e., request to transmit, RTS). In Underwater-FLASHR (UW-FLASHR) [12], 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 [12] or solving computationally hard optimal scheduling problems as in STMAC [13] and STUMP [14]. Distributed approximation algorithms for optimal scheduling were proposed in the literature [13], [14]. However, 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 other's packets by their intended receivers. To satisfy this requirement, each node must evaluate the collision conditions for neighbouring 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 [15]. The key intuition is that each node can predict whether its upcoming packet transmission will collide with another's if it has the neighbouring 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.

## II. BACKGROUNDS AND RELATED WORK

We review the mobile underwater networks, types of mobile sensors, their constraints (e.g., communication characteristics and energy consumption) and then thoroughly examine underwater MAC protocols.

### A. Underwater Networks and Resource Constraints.

The design of oceanic networks for monitoring and scientific exploratory purposes can be largely classified into two categories: (1) static sensors tethered at the seabed or buoys on the ocean surface with external power sources (e.g., NEPTUNE [21]), and (2) mobile sensors such as AUVs and underwater floats (e.g., Sea Web [22], ARGO [23], UCSD Drogues [5]). Static sensors are typically used for long term, pre-planned missions such as seismic activity monitoring, whereas battery-powered mobile sensors are used for short-term missions such as oil and chemical spill monitoring. The key benefit of mobile sensing is that mobility permits more flexible underwater exploration with wide area coverage at reasonable cost. AUVs can follow planned trajectories such as a sequence of track lines, waypoints, and depth excursions [24], while floats have restricted mobility as they move along with water current (e.g., ARGO [23], UCSD Drogues [5]). Given that the cost effective coverage is one of the primary concerns of mobile sensors, such networks must employ low-cost, energy-efficient mobile nodes, and thus, resource constraints must be carefully examined.

*Mobile Sensor Types.* The most common AUV configuration is a torpedo-like vehicle (e.g., REMUS, IVER2) with a streamlined body with propeller and control surfaces at the stern [24]. The speed of such AUVs in the range of 0.5 to 5 m/s, and most vehicles operate at around 1.5 m/s. Another configuration is a glider (e.g., Sea gliders [1]) that uses small changes in its buoyancy in conjunction with wings to make up-and-down, saw tooth-like movements. Although gliders have such restricted mobility patterns, due to the energy efficiency, they can provide data collection on temporal and spatial scales that

would be costly if traditional shipboard methods are used. Unlike AUVs, underwater floats like UCSD Drogues and ARGO [23] mainly use a buoyancy controller for depth adjustment and passively move along with the water current.

*Resource Constraints of Mobile Sensors.* We review the resource constraints of mobile underwater sensors, namely acoustic communications and energy consumption. Communications in the underwater acoustic channel are with two innate characteristics: low bandwidth and large propagation delay.

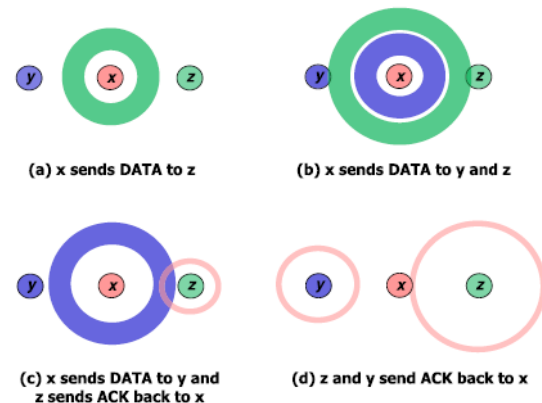


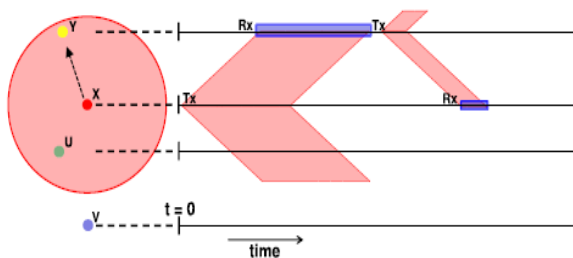
Fig. 1. Temporal reuse.

### B. Review of Underwater MAC Protocols

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 [26]. Since CSMA does not require strict scheduling, it can support node mobility, which is also a major challenge in mobile underwater networks. 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 mobile underwater networks, which is a well-known problem. Moreover, carrier sensing may fail to detect an ongoing transmission due to the propagation delay, which impairs the performance of CSMA protocols.

*Temporal Reuse.* One potential solution for improving CSMA in mobile underwater networks is to utilize temporal reuse that exploits the long propagation latencies of acoustic waves. Fig. 1 demonstrates the notion of temporal reuse. Node x sends a DATA packet to node z in Fig. 1a and again at a later time another DATA packet to node y in Fig. 1b. Node z sends an acknowledgment (ACK) back to node x as node y is about to receive the transmission from node x in Fig. 1c. Finally, node y sends an ACK back to node x in Fig. 1d. This case enables the data and ACKs to be transmitted and received without any collision. To harness this temporal reuse, Yackoski et al. [12] 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. [13] 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 colouring problem. A heuristic, called the Traffic-based one-step trial approach (TOTA), is then proposed to solve the colouring problem. Kredo et al. [14] 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 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. Thus, TDMA-like protocols are not suitable for resource constrained mobile sensor networks.



(a) Exclusive medium access: *exposed terminal*

Fig. 2. Spatial reuse

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) [11], but it does not detail scheduling strategies for out-of-order packet delivery. To reduce the control overhead (e.g., reservation, acknowledgement), R-MAC [28] 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 [29]. Given the fact that two sequential carriers travelling 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. Ng et al. [30] proposed MACA-U which is a redesigned MACA [31]; i.e., the five-state transition rule by considering long propagation delay, the packet forwarding strategy based on priority, and a binary exponential back-off algorithm. Yet MACA-U only considers the case of handling two neighbouring source nodes concurrently transmitting RTS packets, and thus, it does not fully exploit the temporal reuse.

*Spatial Reuse.* Spatial reuse in mobile underwater networks also improves the channel utilization by allowing concurrent transmissions. In Fig. 2a, 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.

### 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 [47]. 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 [18], [48], [49]. 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 mobile underwater networks using the time synchronization for high latency protocol [16]. Two challenges face synchronization of distributed clocks. First, they must be synchronized to a single common event in absolute time or offset (different boot time).

Second, one must determine the skew of a given clock relative to some absolute frequency because clocks are imperfect and run at slightly different rates. For the clock offset between two nodes, they can factor out propagation delay via a two-way message (time-stamp) exchange and fix their time difference with one assumption of no clock skewing during the message exchange. For the clock skew, 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. 3), which is extensible software defined underwater acoustic platform [17]. UANT uses GNU Radio and the USRP for the physical layer and uses TinyOS for upper network layers. The application was created to connect PCs forming a network, using Universal software radio peripheral (USRP). The USRP created by Ettus Research [50], is a radio front-end that is commonly used with GNU Radio. Although the option of

using a sound card provides a low cost solution, the USRP offers a wider frequency range as well as more dedicated hardware. The USRP has a total of four ADCs and four DACs allowing for up to 16 MHz of bandwidth each way, which is proficient for a custom preamplifier board that also incorporates a switch in order to allow for one transducer per node, as well as amplify the received signal entering the USRP.

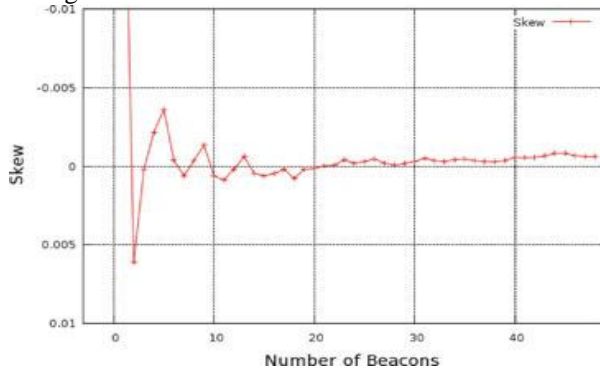


Fig. 3. No. of beacons used in TSHL vs skew estimate.

#### 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 [53]. Recall that this will happen more frequently and be more expensive in mobile underwater networks than in terrestrial radio networks due to the high latency and transmission costs.

##### C. 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.

##### D. Transmission Scheduling

Based on the delay map, a node decides whether or not it can transmit a packet without possible interference with a neighbor node's packet reception. While node *y* is waiting for CTS transmission, node *u* also receives this RTS and has data to send. Considering that a collision only occurs in receiver side, 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 (node *u*'s prospective RTS and DATA transmissions should not interfere with node *x*'s CTS and ACK receptions). *Prospective non-interference*. Its future receptions (CTS and ACK) must not be interfered with by neighbors' prospective transmissions (node *u*'s prospective CTS and ACK receptions should not be interfered with by node *x*'s prospective DATA transmission).

#### Algorithm 1 Transmission Scheduling Algorithm

```

1: procedure neighboring non-interference(Message m)
2: for all e ∈ delay map entries do
3:   arrivalm→e.src ← propdelay + txdelay
4:   if e.frame_type == RTS then
5:     arrivalCTS ← e.timestamp + delaye.src→e.dest +
      propctrl_delay
6:     if arrivalm→e.src ∈ [arrivalCTS ± tGUARD] then
7:       return collision detected
8:     end if
9:     arrivalACK ← e.timestamp + delaye.src→e.dest +
      (2 × propctrl_delay) + propdata_delay
10:    if arrivalm→e.src ∈ [arrivalACK ± tGUARD] then
11:      return collision detected
12:    end if
13:   else if e.frame_type == CTS then
14:     arrivalDATA ← e.timestamp + delaye.src→e.dest +
      propctrl_delay
15:     if arrivalm→e.src ∈ [arrivalDATA ± tGUARD] then
16:       return collision detected
17:     end if
18:   else if e.frame_type == DATA then
19:     arrivalACK ← e.timestamp + delaye.src→e.dest +
      propdata_delay
20:     if arrivalm→e.src ∈ [arrivalACK ± tGUARD] 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

```

##### E. Schedule Recovery

A node may miss its neighbors' RTS/CTS/DATA packets due to the half-duplex nature of the acoustic modem 1. To simplify the pseudo code, the lossy nature of an acoustic channel. Under such circumstances, a node may begin its transmission sequence with an incomplete delay map, which may cause packet collision.

##### F. Guard Time

DOTS use a guard time to support node mobility caused by the ocean currents. Each node calculates this guard time as  $2 \times \delta_{\text{average movement distance}} = \text{speed of sound} \times P$  when it checks the transmission scheduling algorithm.

## V. SIMULATION AND EVALUATION

### A. Simulation Setup

1) *Simulation Parameters*: For acoustic communications, the channel model described in [55] and [56] is implemented in the physical layer of QualNet.

2) *Topology*: As shown in Fig 6, we deployed the nodes in a line and a star topologies in a 3D region of 5 km\_ 5 km\_ 5 km. In the line topology depicted in Fig. 6a, four nodes are deployed in a line and with a fixed distance between one-hop neighbors. The distances between the nodes are varied from 750 m to 1:5 km for the experiments, and thus the two nodes, B and C, are exposed to each other. We adopt this line topology to show how spatial reuse affects system throughput. As the distance between each pair increases, simulation results will also indicate how temporal reuse can affect system throughput.

### B. Simulation Results

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

$$\text{Throughput} = \frac{\# \text{ of rx data frames} \times \Delta \text{data}}{\text{Simulation Duration}},$$

where  $D_{\text{data}}$  denotes the duration of transmitting a data frame.

$$\text{Offered Load} = \frac{\# \text{ of generated data frames} \times \Delta \text{data}}{\text{Simulation Duration}}.$$

2) *Energy Consumption*: 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 percent 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) *Impact of MCM Mobility* :The effect of MCM mobility is examined in Figs. 12 and 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 0:3 m/s with each node having a 750 m transmission range.

The overall trend of this MCM scenario is quite consistent with the previous results in Fig. 11. Note that DOTS consumes more energy than S-FAMA and DACAP because it delivers by far more frames than these two protocols. In the same way, the energy consumption of CS-ALOHA is several times higher than other protocol because of increased number of received packets and collisions.

4) *Guard Time*: Evaluating the performance of DOTS by varying the guard time intervals is important as we can show the sensitivity of guard time with respect to the speed of nodes. If the guard time is too short, the chances of packet collisions will be too high. If it is too long, packet collisions will rarely happen, but we have lower chances of exploiting temporal/spatial reuse. In Fig. 15, we show the throughput performance based on different guard time intervals ranging from 1 to 8 ms. All intervals show positive correlation with offered load. It shows that the guard time interval of 2 ms shows the best throughput

performance. The guard time intervals of 1 and 8 ms show slightly lower throughput performance due to collisions and lower utilization, respectively.

5) *Packet Delivery Latency*: We compare the overall latency for packet delivery that includes RTS/CTS exchanges, data delivery, and ACK reception. We measure the latency by analyzing the packet transmission log data. For latency measurement, we search for the first RTS packet of a successfully delivered data packet; thus, considering channel contention and packet loss. For fair comparison of different protocols, we did not include queueing delay; DOTS is more favorable in terms of queueing delay than the other protocols. To understand the impact of topologies, we use two scenarios, i.e., line and star topologies

In this high contention scenario, each sender transmits 512 byte data frames to the receiver. Fig. 17 shows the latencies of the four protocols. Fig. 14. Energy consumption in the MCM scenario with fixed data size (512 bytes). Fig. 15. Guard time sensitivity to a MCM mobility speed (3 m/s). As shown in Fig. 9, DOTS's latency is given as 8.86 s on average, whereas other protocols have much higher latencies. DACAP's latency is 16.23 s on average and it outperforms S-FAMA whose average latency is 20.93 s. This gain is due to DACAP's capability of spatial reuse. As in the line topology, CS-ALOHA's performance is far superior to other protocols as it lacks channel reservation, resulting the latency of 3.86 s. However, as shown earlier, this gain comes at the cost of low data rate and protocol fairness due to high collision and lack of fairness control).

6) *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 Syed et al. [18]. 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 [62], defined as below

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

where  $x_i$  denotes the throughput of node  $i$  and  $n$  denotes the number of nodes in the network. Fig. 18, 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 4.3, 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.

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. This channel capturing also leads to severe data collisions at other nodes which have not captured the channel, inducing poor energy utilization.

## VI. TOWARDS ENABLING MULTIPLE TRANSMISSION SESSIONS

We allow each node to manage multiple independent sessions, and thus, there could be multiple outstanding packets within a session period (pipelined). To illustrate the advantages of MDOTS over DOTS, consider a line topology (A – B – C) where node B can reach both A and C, but they are hidden from one another. In this case, DOTS is only able to transfer one session (i.e., RTSCTS-DATA-ACK sequence) to a single receiver. As depicted in Fig. 19, in the same amount of time, MDOTS can actively initiate two different sessions (one to node A and the other to C); node B first transmits an RTS destined for node A. While the RTS packet is still propagating, node B waits for a random period of time and then transmits another RTS destined for node C. When node A receives its RTS, it waits until time has passed (i.e., total packet transmission time  $\beta$  maximum propagation delay) and then replies with a CTS. Meanwhile, node C has also received its RTS, and replies with a CTS after waiting the appropriate amount of time. Node B receives CTS messages from both nodes A and C.

Note that node B can initiate multiple sessions to each destination node (either A or C) by sending another RTS to the destination before receiving an ACK from nodes A or C for the previous session's DATA transmissions. To support this, the MAC layer is extended to maintain a buffer that queues multiple packets received from the Network layer.

## VII. CONCLUSION

We proposed a MAC protocol called DOTS that alleviates limitations caused by the long propagation latency and the severely limited bandwidth of acoustic communications. DOTS aimed to achieve better channel utilization by harnessing both temporal and spatial reuse. Extensive simulation results showed that (1) DOTS outperforms S-FAMA by 200 percent and DACAP by 15 percent in the line topology (exposed terminal) and S-FAMA by 200 percent and DACAP by 70 percent 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. Moreover, we have introduced a mechanism of enabling multiple transmission sessions (called MDOTS). A preliminary evaluation showed that MDOTS significantly outperforms the original DOTS.

## REFERENCES

- [1] S. Roy, P. Arabshahi, D. Rouseff, and W. Fox, "Wide Area Ocean Networks: Architecture and System Design Considerations," Proc. First ACM Int'l Workshop on Underwater Networks (WUWNet'06), Sept. 2006.
- [2] I.F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," Ad Hoc Networks, vol. 3, pp. 257-279, 2005.

- [3] 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," Proc. IEEE Military Comm. Conf. (MILCOM), 2005.
- [4] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data Collection, Storage, and Retrieval with an Underwater Network," Proc. Third Int'l Conf. Embedded Networked Sensor Systems (Sensys), 2005.
- [5] J. Jaffe and C. Schurgers, "Sensor Networks of Freely Drifting autonomous Underwater Explorers," Proc. First ACM Int'l Workshop on Underwater Networks (WUWNet '06), 2006.
- [6] 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," Proc. IEEE 11th Symp. Computers and Comm. (ISCC '06), 2006.
- [7] Z. Zhou, J. Cui, and A. Bagtzoglou, "Scalable Localization with Mobility Prediction for Underwater Sensor Networks," IEEE Trans. Mobile Computing, vol. 10, no. 3, pp. 335-348, Mar. 2011.

## BIOGRAPHIES



**Arockia Rogini.A** received her B.Tech degree in Information Technology in Loyola Institute of Technology, Chennai, in 2013, Tamilnadu, India. Now she is doing her Master in Engineering in St. Joseph's college of engineering and technology, Thanjavur, Tamilnadu, India. She has attended 2 national conferences. She is interested in mobile computing.



**Arumai Ruban.J** received his B.Sc in Physics and M.C.A in St. Joseph's College, Trichy, Bharathidasan University in 2007, 2010 respectively, Tamilnadu, India. He has done his Master in Engineering in CSE, in Periyar Maniammai University, Thanjavur. He is working as an Assistant Professor in St. Joseph's College of engineering and technology, Thanjavur. He has handled various subjects and interested in Networks. He has attended 4 national conferences, 1 international Journal and 1 international conference.