Toward Practical MAC Design for Underwater Acoustic Networks

Yibo Zhu*, Zaihan Jiang†, Zheng Peng*, Michael Zuba*, Jun-Hong Cui* and Huifang Chen‡
*Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA
†Acoustic Division, U.S. Naval Research Lab, Washington DC, USA
‡Department of Information Science and Electronic Engineering, Zhejiang University, China
Email: {yibo.zhu, zhengpeng, michael.zuba, jcui}@engr.uconn.edu, zaihanjiang@yahoo.com, chenhf@zju.edu.cn

Abstract—**Recently, various Medium Access Control (MAC) protocols have been proposed for underwater acoustic networks. These protocols have significantly improved the performance of MAC layer in theory. However, two critical characteristics, low transmission rates and long preambles, found in the commercial modem-based real systems, drastically degrade the performance of existing MAC protocols in the real world. A new practical MAC design is demanded. Toward a proper approach, this paper analyzes the impact of the two newly found modem characteristics on the random access-based MAC and handshake-based MAC, which are two major types of MAC protocols for underwater acoustic networks. We further develop the nodal throughput and collision probability models for representative solutions of these two MAC protocol types. Based on the analysis, we believe time sharing-based MAC is very promising. Along this line, we propose a time sharing-based MAC and analyze its nodal throughput. Both analytical and simulation results show that the time sharing-based solution can achieve significantly better performance.**

I. INTRODUCTION

Underwater Acoustic Networks (UANs) are a class of ad hoc networks deployed in underwater environment in which the communications among nodes use acoustic waveforms. Compared with terrestrial wireless networks in which RF signals are typically used, UAN features low available bandwidth, long propagation delays and high dynamic channels. These unique characteristics pose significant challenges to network protocol design, especially Medium Access Control (MAC) protocols for underwater acoustic networks [1]–[6].

During the past a few years, many studies have focused on the area of underwater acoustic networks and proposed various solutions, especially for MAC protocol designs. Initially, random access-based solutions were widely studied. Vieira et al. found Slotted-ALOHA degenerated to pure ALOHA in the long propagation delay characterized underwater network environment [7]. The throughput of Slotted-ALOHA was then improved by introducing guard time. Additionally, two variants of ALOHA were designed to achieve a better performance [8].

Since the random access-based approach cannot effectively avoid collisions, some studies have explored the handshake-based MAC with contention. In order to improve the energy efficiency of FAMA [9], which is designed for long propagation delay characterized networks, Slotted-FAMA [10] is proposed. In Slotted-FAMA, time is slotted and any packet can only be transmitted at the beginning of a slot. Still using control packets to reserve time slots for data packets, UW-FLASHR [11] reduces the cost of channel reservation by training collision-free packets sending pattern.

Although these protocols can achieve good performance in theory, there is still a big gap for them to being adopted in the real world. Through a series of field tests, we have identified two practical issues caused by real acoustic modem characteristics, low available transmission rates and long preambles, which significantly degrade the performance of existing MAC protocols. Although [12] presents the simulation results with consideration of these factors, it neither theoretically analyzes the impact nor provides dedicated solutions. The challenges posed by these two characteristics call for a new MAC design so that good performance can be achieved in the real system.

To explore the practical MAC design, in this paper, we analyze the impact of these two modem characteristics on random access-based MAC and handshake-based MAC. Through the analysis, we believe that time sharing-based MAC is promising to address the aforementioned challenges. In literature, several time sharing-based protocols [13]–[15] have been proposed. However, these protocols pre-schedule time slots when the topology is given and thus are not on-demand. In other words, they cannot adapt to dynamic environments or dynamic user requirements. To study the capability of time sharing-based MAC, we propose **C**luster-based **O**n-**D**emand **T**ime **S**haring MAC solution (COD-TS) and develop the corresponding throughput model. We further compare the performance of Slotted ALOHA, Slotted FAMA and COD-TS in simulations. The results agree with our analysis that COD-TS outperforms the other two MAC protocols.

The rest of this paper is organized as follows. Section II presents newly identified modem characteristics and the corresponding impact. In Section III collision probability and nodal throughput models are developed for random access-based MAC and handshake-based MAC respectively. Section IV describes COD-TS and its nodal throughput model. In Section V simulations are conducted to validate the developed models and compare the performance of various solutions. Finally, Section VI draws our conclusions and briefs future work.

II. MODEM CHARACTERISTICS AND IMPACTS

Making use of Aqua-TUNE [16], a recent testbed for underwater acoustic networks, numerous field tests have been conducted to evaluate the performance of various MAC protocols. Through these field tests, several real modem characteristics have been identified. Among them, some characteristics significantly restrict the application of existing MAC protocols. For example, the maximum transmission range cannot be predetermined and closely depends on the deployment environment. However, it is usually required by many existing MAC protocols, such as Slotted-FAMA [10]. Due to space limitation, we only focus on two major characteristics, long preambles and low transmission rates, which pose new challenges to practical MAC design for underwater acoustic networks.

A. Characteristics

1) Long Preamble: For purpose of synchronization, signal detection and channel estimation, a preamble sequence is included at the beginning of a physical layer frame. In wired and radio networks, preamble is very short and transmission rate is high, so the delay T_{pre} caused by preamble is very short and usually ignored. But strictly the packet transmission time T_{tx} should include T_{pre} and be defined as Eq. 1.

$$T_{tx} = \frac{L}{B} + T_{pre} \tag{1}$$

where L is the packet size and B is effective modem transmission rate. In the real acoustic system, we have found that the preamble is very long and cannot be ignored anymore. However, the preamble is usually ignored by the research community. In Aqua-TUNE, the physical device is ATM series Teledyne Benthos Modem [17], a widely used commercial underwater acoustic modem. Through the tests, we have noticed that the preamble lasts for about 1.5 s, which drastically increases the packet transmission time and accordingly makes the packet collision probability much higher. To the best of our knowledge, all acoustic modems have long preamble issues and this may be determined by the low acoustic transmission rate. In section II-B we will show its impact on packet collision probability in detail.

2) Low Transmission Rate: Many existing works claim that modem transmission rate is about 10 $kbps$ or even higher [10], [11], [15], [18]–[20]. However, the useable transmission rate is much lower than the theoretical value, especially in shallow water environment, i.e., with horizontal channel. For example, although the transmission rate of Benthos Modem is claimed to be up to 15 $kbps$, the highest useable transmission rate is much lower than that in order to ensure a low packet error rate with horizontal channel. In many of our field experiments, with horizontal distance of less than a few kilometers, a practical acoustic rate is 800 bps. In fact, based on our experiment results as illustrated in Fig. 1, the effective transmission rate is even lower.

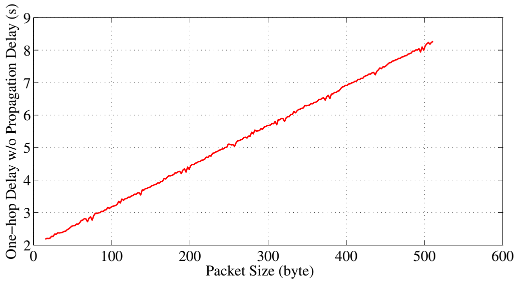


Fig. 1. Measured one-hop delay with Benthos modem

In the experiment, two modems transmitting at 800 bps are placed less than one meter apart such that the propagation delay can be neglected. In this way, the measured delay only contains the transmission delay, preamble delay, and some other hardware delays. By fitting the data to a curve, we get an approximate relationship between packet size L and one-hop delay D (excluding propagation delay) as shown in Eq. 2.

$$D = 0.0015L + 1.9615 \tag{2}$$

From Eq. 2 we can tell that B, indicated by the reciprocal of the slope, is about 667 bps, which is extremely low compared with 10 $kbps$. Then substituting $B = 667$ bps and $T_{pre} = 1.5$ s into Eq. 1 gives the packet transmission time with Benthos modem as below

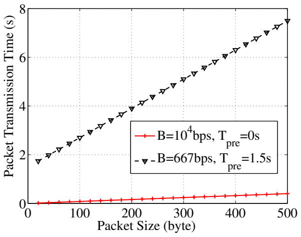


Fig. 2. Transmission time with varying packet size

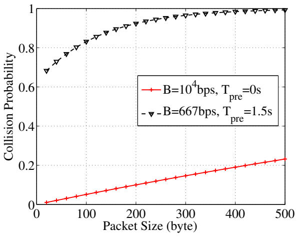


Fig. 3. Collision probability with varying packet size

$$T_{tx} = \frac{L}{667} + 1.5 \tag{3}$$

B. Impacts

Intuitively, compared with $B = 10^4$ bps and $T_{pre} = 0$ s, namely, **conventional settings**, $B = 667$ bps and $T_{pre} = 1.5$ s, namely, **practical settings**, renders a much longer packet transmission time, which further results in a much higher packet collision probability. In order to demonstrate the impact, we take ALOHA as example and evaluate the transmission time as well as the packet collision probability with the conventional settings and practical settings respectively. In the evaluation, the number of neighbors n is set to 6, the nodal traffic rate λ is 0.06 pkt/s. Packet transmission time is calculated according to Eq. 1 and collision probability is measured based on Eq. 4 [21].

$$P_c = 1 - e^{-2G} \tag{4}$$

where $G = n\lambda T_{tx}$. Although this model is originally proposed for ALOHA in radio networks, many works like [22] have reused it for ALOHA in underwater acoustic networks. The evaluation results are shown in Fig. 2.

From Fig. 2, we can see that the packet transmission time with practical settings is much longer than the one with conventional settings. And as illustrated in Fig. 3, the collision probability roars when long preambles and low transmission rates arise, which indicates the random access-based MAC suffers heavy collisions with the real system. In particular, according to Fig. 3 even a packet of size 20 $bytes$ may collide with probability 0.68, which is significantly higher than that measured with conventional settings. As the collision probability of short packet is not low anymore, handshake-based MAC becomes inefficient. Basically, in radio networks short control packets are used to reserve channel for long data packets. This mechanism can help to alleviate collisions because the short control packets have a very low collision probability, and this is the principle and assumption of handshake-based approaches. As shown in Fig. 3, however, this assumption does not hold in real underwater acoustic system, which implies that handshake-based MAC cannot efficiently avoid collisions or achieve a high throughput. Therefore, the low transmission rate and long preamble issues severely impact the performance of both random access-based MAC and handshake-based MAC. In order to better investigate the impact, in next section we will model the collision probability as well as the nodal throughput of a random access-based MAC protocol and a handshake-based MAC protocol respectively.

III. Analysis and Implications

In this section, we model the collision probability and nodal throughput of Slotted ALOHA and an RTS/CTS-based MAC, which belong to random access-based MAC and handshake-based MAC, respectively. Then based on the models, we analyze the root reasons which severely impair the performance of these two protocols. For clarity, we first introduce the following network model adopted by the rest parts of this paper.

- As a standard assumption, the traffic rate generated by each sender follows a Poisson process with the same λ parameter and all data packets are of size L_D.
- Since this work emphasizes on the analysis of collision probability, for simplicity packet error not caused by collisions is not considered in the model.
- All models are developed in the same network scenario, a fully connected one-hop network with N nodes including n $(n \leq N)$ senders.
- Same as [11], [13], [15], all nodes have the same transmission rate B and maximum transmission range R. Then, the maximum propagation delay T_{mp} is R/v_p, where v_p is the propagation speed.

A. Slotted-ALOHA

In Slotted-ALOHA, time is slotted. At the beginning of each slot, a node transmit a packet if there is any queued. The duration of a slot equals the transmission time of the maximum frame plus T_{mp}. In addition, to overcome other possible delays, such as processing delays, a short constant time interval T_{guard}, namely, guard time, is added to the slot. In this way, Slotted-ALOHA can ensure that there is no overlapped receptions at the receiver unless packets are transmitted in the same slot, and it can achieve a higher nodal throughput than ALOHA. And that is why we choose Slotted-ALOHA instead of ALOHA as the representative of the random access-based approach.

According to the description, a node can send at most one packet within a slot. Also note that packets arrive following a Poisson process with parameter λ, so the MAC layer at each node can be modeled as an M/G/1 queue system. In this system, the packet arrival rate is λ, and the packet processing rate is $1/T_s$, where T_s is the duration of one slot. Note that all packets are of size L_D, with which we can obtain packet transmission time T_D via Eq. 1. Then, according to the definition of slot duration, T_s is given as

$$T_s = T_D + T_{mp} + T_{guard} \tag{5}$$

Note that T_s is the expected service time per packet, and thus system utilization factor ρ can be obtained as

$$\rho = \lambda T_s \tag{6}$$

Furthermore, according to [23], the probability P_{ne} that a node's queue is not empty is

$$P_{ne} = min\{\rho, 1\} \tag{7}$$

In addition to P_{ne}, packet collision is also related to network topology due to spatial-temporal difference. However, according to our experiment experience, in real system, T_D is longer than T_{mp}, so we can ignore the impact of network topology and a packet can be correctly received if only one packet is transmitted in a slot without collision. Based on this observation, accordingly, the corresponding probability P_{succ} is

$$P_{succ} = \left(\begin{array}{c} \text{n} \\ 1 \end{array} \right) P_{ne}(1 - P_{ne})^{n-1} \tag{8}$$

If more than one packet are sent during the same slot, there would be a collision. Thus, excluding P_{succ} and the probability that no packet is sent in one slot from 1 gives the collision probability P_c as below

$$P_c = 1 - P_{succ} - (1 - P_{ne})^n \tag{9}$$

With P_{succ} we can further get the nodal throughput. On average, these n senders can successfully delivery $P_{succ}L_D$ data in one slot of duration T_s, so the nodal throughput Λ is

$$\Lambda = \frac{P_{succ}L_D}{nT_s} \tag{10}$$

B. RTS/CTS-based MAC

The RTS/CTS-based MAC, as illustrated in Fig. 4, is actually a simplified version of Slotted-FAMA [10]. Same as Slotted-FAMA, time is slotted and all packets can only be sent at the beginning of a slot such that the impact of long propagation delay can be alleviated. The slot length is set to the sum of the transmission time of the CTS packet, the maximum propagation delay, and the guard time.

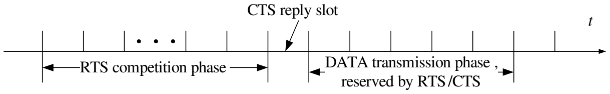


Fig. 4. RTS/CTS-based MAC

As entering the RTS competition phase, each node having queued data packets randomly selects one of m slots in RTS competition phase. When the selected slot arrives, the node sends an RTS which carries the slot number for data packet transmission. If multiple nodes send RTS in the same slot, a conflict among RTS requests occurs and all nodes re-enter the RTS competition phase. This procedure will repeat until one node win the channel competition by solely sending RTS in a slot successfully. Then CTS is replied in the next slot. Subsequently, the sender transmits the data packet during the slots reserved by RTS and CTS, while other nodes keep silent during this period. Recall that packet loss is only caused by collision, so data packet must be delivered correctly. Accordingly, acknowledgement scheme is not adopted for simplicity. After the transmission of data packet, all nodes enter the RTS competition phase again.

According to the description, RTS/CTS-based MAC can be modeled as an M/G/1 queue at each node and the corresponding state transition diagram is shown in Fig. 5.

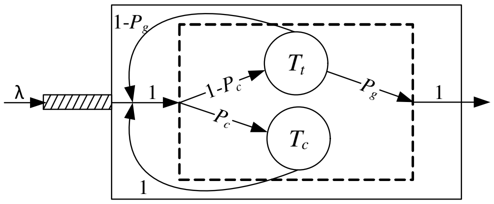


Fig. 5. State transition diagram of M/G/1 queue model for RTS/CTS-based MAC

As illustrated in Fig. 5, the arrival rate of data packet is λ. When senders compete the channel in one RTS competition phase, the conflict of RTS requests occurs with probability P_c while one sender win the competition with probability $1-P_c$. If conflict happens, time T_c is wasted by this failed channel competition phase. Otherwise, one of the n senders must grasp the channel and continue the remaining interaction with the receiver, and meanwhile other $n-1$ senders keep silent. We assume T_t is the expected time to grasp the channel in one RTS competition phase and finish the CTS and data packets transmission. Note it is fair for these n senders to grasp the channel. Therefore, given that the channel is grasped, each node grasps the channel and finally deliver a data packet with probability $P_g = \frac{1}{n}$, and then the packet can leave this queue system with probability 1. Similarly, with probability $1-P_g$ a sender does not grasp the channel and wastes time T_t.

Before going further, we first simplify the state transition diagram by multiplying the probability on the directly connected edges, and get the one in Fig. 6. Based on this state transition diagram, we can calculate the key parameters of the M/G/1 queue system, and then derive the collision probability as well as nodal throughput.

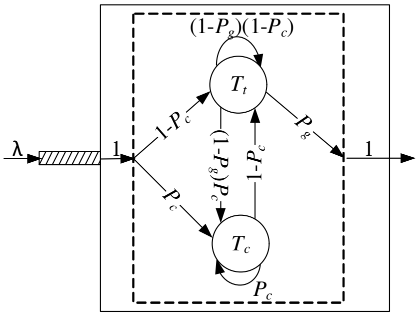


Fig. 6. Equivalent state transition diagram of M/G/1 queue model for RTS/CTS-based MAC

If P_{ne} is the probability that a node's data packet queue is not empty, then with probability P_{ne} the corresponding node randomly selects one of m slots to send an RTS packet. Let A_k denote the event that no node selects the first $k-1$ slots, and its probability is

$$P(A_k) = \prod_{i=1}^{n} P_{ne} \frac{m-(k-1)}{m} = \left[P_{ne} \frac{m-(k-1)}{m}\right]^n \quad (11)$$

Given that event A_k occurs, the conditional probability of E_k, the event that only one node chooses the k^{th} slot, is

$$P(E_k|A_k) = \left(\begin{array}{c} \mathrm{n} \\ 1 \end{array}\right) \frac{P_{ne}}{m-(k-1)} \left(1 - \frac{P_{ne}}{m-(k-1)}\right)^{n-1} \quad (12)$$

Note that a node win the competition in the k^{th} slot if and only if both events A_k and E_k happen, so the probability $P_{g,k}$ that a node wins the channel competition in the k^{th} slot is

$$P_{g,k} = P(E_k A_k) = P(A_k) P(E_k|A_k) \quad (13)$$

Consequently, the probability P_w that one node wins the channel competition during any of the m slots is

$$P_w = \sum_{k=1}^{m} P_{g,k} \quad (14)$$

Then, given there is a winner in one competition, the conditional probability that a node wins the competition during the k^{th} slot is $\frac{P_{g,k}}{P_w}$. Thus, given there is a winner in one competition, n_s, the expected number of slots taken to win the channel, is

$$n_s = \sum_{k=1}^{m} k \frac{P_{g,k}}{P_w} \quad (15)$$

As the channel is grasped, CTS and data packets will be transmitted successively. CTS packet takes one slot to be transmitted, and the time for transmitting data packet is the least number of slots no less than the data packet transmission time plus the maximum propagation delay. Accordingly, T_t is

$$T_t = n_s T_s + T_s + \left\lceil \frac{T_D + T_{mp} + T_{guard}}{T_s} \right\rceil T_s \quad (16)$$

As for T_c, it can be evaluated by $P_{f,k}$, the probability that the competition fails in the k^{th} one among the m slots, i.e., both events A_k and C_k occur, where C_k is the event that an RTS collision occurs in the k^{th} slot. Given that A_k happens, the conditional probability of C_k is

$$P(C_k|A_k) = \\ 1 - \sum_{i=0}^{1} \left(\begin{array}{c} \mathrm{n} \\ \mathrm{i} \end{array}\right) \left(\frac{P_{ne}}{m-(k-1)}\right)^i \left(1 - \frac{P_{ne}}{m-(k-1)}\right)^{n-i} \quad (17)$$

Furthermore, $P_{f,k}$ can be obtained as

$$P_{f,k} = P(C_k A_k) = P(A_k) P(C_k|A_k) \quad (18)$$

Since the collision may occur in any of the m slots, the collision probability P_c in the RTS competition phase is

$$P_c = \sum_{k=1}^{m} P_{f,k} \quad (19)$$

Subsequently, T_c can be obtained based on P_c and $P_{f,k}$. Concretely, given that a request collision occurs during one of the m slots, the conditional probability that a collision occurs during the k^{th} slot is $\frac{P_{f,k}}{P_c}$, and thus T_c is

$$T_c = T_s \sum_{k=1}^{m} k \frac{P_{f,k}}{P_c} \quad (20)$$

Until now we have all the parameters, T_t, T_c, P_c and P_g, in the state transition diagram. Then, based on Fig. 6, it is straightforward to get the entrance vector $\mathbf{p}$, the transition matrix $\mathbf{P}$, and the process rate matrix $\mathbf{M}$ [23] as below.

$$\begin{aligned} \mathbf{p} &= \left[\begin{array}{cc} 1-P_c & P_c \end{array}\right] \\ \mathbf{P} &= \left[\begin{array}{cc} (1-P_g)(1-P_c) & (1-P_g)P_c \\ 1-P_c & P_c \end{array}\right] \\ \mathbf{M} &= \left[\begin{array}{cc} {T_t}^{-1} & 0 \\ 0 & {T_c}^{-1} \end{array}\right] \end{aligned} \quad (21)$$

Accordingly, the process time matrix $\mathbf{V}$ is

$$\begin{aligned} \mathbf{V} &= \left[\mathbf{M}(\mathbf{I}-\mathbf{P})\right]^{-1} \\ &= \left[\begin{array}{cc} {T_t}^{-1}(P_g + P_c - P_g P_c) & {T_t}^{-1}(P_g - 1)P_c \\ {T_c}^{-1}(P_c - 1) & {T_c}^{-1}(1-P_c) \end{array}\right]^{-1} \end{aligned} \quad (22)$$

where $\mathbf{I}$ is 2×2 identity matrix. Furthermore, the expected time T_p to deliver one data packet can be obtained as

$$T_p = \mathbf{pV}\boldsymbol{\epsilon}' \tag{23}$$

where $\boldsymbol{\epsilon}=[\ 1 \quad 1\ ]$. Then, same as the analysis in section III-A, P_{ne} is calculated as

$$P_{ne} = min\{\rho, 1\} = min\{\lambda T_p, 1\} \tag{24}$$

By substituting all parameters into Eq. 24, we can get an equation for variable P_{ne}. Since the right side of the equation is too complicated, we resort to numerical method (iteration) to solve it. After getting P_{ne}, we can calculate T_p via Eq. 23. Finally, note that each sender can correctly deliver $L_D P_{ne}$ bytes data during the T_p period, so the nodal throughput is

$$\Lambda = \frac{L_D P_{ne}}{T_p} \tag{25}$$

C. Collision Probability Evaluation and Implications

In order to quantitatively show the impact of aforementioned two modem characteristics, we evaluate the collision probabilities of Slotted ALOHA and RTS/CTS-based MAC via the developed models above. Then we discuss their implications for underwater MAC design. Note that throughput is directly determined by collision probability, so the validation of throughput models in section V can also justify the collision probability models.

In the evaluations, there are 4 pairs of sender and receiver, total 8 nodes. The data packet and control packet are of size 400 $bytes$ and 20 $bytes$, respectively. Then setting packet generation rate λ from 0.01 pkt/s to 0.2 pkt/s in step 0.01 pkt/s gives the evaluation results shown in Fig. 7.

From Fig. 7 we can see that the collision probability of Slotted ALOHA with the practical settings is much higher than that with the conventional settings. Particularly, when λ exceeds 0.1 pkt/s, the collision probability is almost 1, which is a disaster for a network. That is, the performance of Slotted ALOHA is severely poor with the practical settings. As for RTS/CTS-based MAC, the collision probabilities with both settings finally reach the same upper bound. According to Eq. 11, 18, 19 and 17, essentially, this is because the collision probability P_c is only determined by m when λ is high enough such that P_{ne}=1. From Fig. 7, it is also observed that the collision probability when adopting the practical settings immediately reaches the upper bound as λ slightly increases, while the other one gradually approaches to the upper bound. This indicates that RTS/CTS-based MAC with the practical settings cannot alleviate the collisions as well as that with the conventional settings.

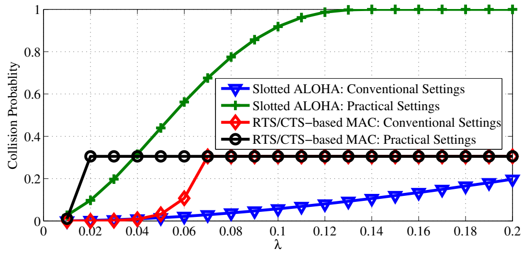


Fig. 7. The impact of real modem characteristics on the collision probability

From the discussions above we can see that both random access and handshake-based MAC cannot achieve a good performance when the long preambles and low transmission rates issues arise. The essential reason is that these two modem characteristics result in significantly long packet transmission time even for a very short packet. Thus the key point of practical MAC design becomes how to effectively avoid collisions for packets with long transmission time. Delightedly, this is right the feature of time sharing-based approach. Basically, in time sharing-based MAC nodes send packets following collision-free schedule such that collisions can be resolved regardless of packet transmission time. In order to explore the capability of time sharing-based MAC, we will design a basic protocol and analyze its performance in the remaining parts of this paper.

IV. COD-TS: A Time Sharing-based MAC

As discussed in section III-C, time sharing-based MAC has the innate feature to avoid collisions among packets regardless of packet transmission time, and thus it is promising to overcome the challenges posed by long preambles and low transmission rates. In order to explore the capability of time sharing-based MAC, **C**luster-based **O**n-**D**emand **T**ime **S**haring MAC (COD-TS) is introduced and analyzed. Different from existing time sharing-based MAC protocols [13]–[15] which calculate schedule off-line based on costly topology information and cannot be on-demand, COD-TS adopts cluster technique to enable local schedule and on-demand feature.

Since in this paper we focus on exploring the capability of time sharing-based MAC, here we first introduce COD-TS for one-hop network, which the analysis of capability is based on. After that, we extend the scheme to multi-hop scenarios. Due to space limitations, we can only provide a brief description.

A. Description of COD-TS in One-hop Network

In COD-TS, nodes need to be synchronized such that they can measure the propagation delays to their neighbors. This can be done with existing synchronization algorithms like that in [24] and achieve microsecond accuracy. Then, based on the propagation delays, nodes can map assigned slots to their own timelines using the method in [25] with consideration of spatial-temporal differences. Compared with the propagation delays and transmission delays on the order of seconds, the nodes' clock drift is insignificant during a relatively long period. Plus, using guard time technique like that in [11], COD-TS can easily tolerate time difference among nodes on the order of milliseconds or even higher. This allows a relatively long interval between re-synchronizations, so the synchronization cost is not high.

In COD-TS, nodes are organized into clusters by running distributed clustering algorithm [26]. As shown in Fig. 8, cluster members M_1 and M_2 are one-hop neighbors of cluster head H_A, which is responsible for scheduling data sending times for all cluster members. A communication round of a cluster is defined as a sequence of data or control packet exchanges between two successive schedule updates of the cluster head. Each cluster repeats the communication round again and again. In Fig. 8, for example, the period between time t_1 and t_2 is one communication round of H_A's cluster. In each round, cluster head schedules sending slots for its members according to their requests. That is, the duration of the communication round is not fixed but on-demand unless total scheduled packet transmission time exceeds the predefined maximum round time D_{max}, which is to guarantee the fairness among clusters. Note that the slots here may not have equal durations.

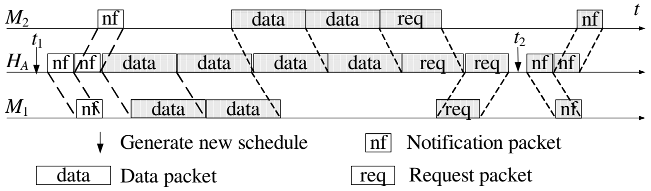


Fig. 8. A communication round for single hop case

Each communication round starts with the new schedule generation, followed by sending notification packets to cluster members. These two kinds of events are arranged by the previous schedule update. And the current schedule update arranges the time slots for data and request packets in this round, the next schedule generation, and the notification packets to members in the next communication round. In Fig. 8, for example, when H_A updates schedule at time t_1, it assigns slots for all shaded packets followed by next schedule update time t_2 using Algorithm 1. Particularly, the initial schedule also arranges the notification packets to members in the current round. With the notification packet, a cluster member can know when to send the request packet and data packets.

When the scheduled time for a cluster member i to send request packet arrives, it sends the cluster head a request packet $P_{req}\left\{\mathbf{R}_i, \mathbf{P}_i\right\}$, where $\mathbf{R}_i$ is the set of requested times for the data packets to be sent and $\mathbf{P}_i$ is the set of propagation delays between node i and its neighbors. The propagation delays can be evaluated based on the timestamps in the overheard packets from neighbors. Particularly, if a cluster member does not have any data packet to send, it simply skips its request slot.

After collecting the requests from members, cluster head generates new schedule with Algorithm 1 at the arranged schedule update time.

In Algorithm 1, except for the input parameters have been introduced above, $T_{tx,D}$ is the transmission time of the data packet of size L_D. In line 1, $getDSDur()$ returns the maximum duration for data sending and the method is described in detail in section IV-B. The outputs $\mathbf{T}_R$, $\mathbf{T}_N$ are the sets of slots for members to send requests and receive notifications respectively. T_u is the time when cluster head will generate the next new schedule. $\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_n$ are the sets of time slots indicating when the corresponding member should send data packets. Specifically, when finding an available slot for a packet from the node i to j, node i's cluster head maps and marks node i's sending/recving slots, node i's neighbors' receiving slots as well as node j's sending/recving/intereference slots on the timeline from node i's perspective using the method in [25]. Then the usable slot can be found by locating the first slot which does not overlap with any marked ones.

After obtaining the new schedule, cluster head generates a notification packet $P_{nf}\{\mathbf{S}_i, T_{R,i}\}(1{\le}i{\le}N)$ for each member and sends it out at time $T_{N,i}$, where $T_{R,i}{\in}\mathbf{T}_R$ and $T_{N,i}{\in}\mathbf{T}_N$ are the time for member i to send request packet to cluster head, and the time for cluster head to send notification packet to member i, respectively. Then, member i can send data packets and request during the arranged slots. To improve the channel utilization, each cluster member can piggy-back ACKs in the request packet. Then, the cluster head includes the ACKs in the notification packets to the corresponding senders.

Algorithm 1 Generate new schedule

```
Input: $n, D_{max}, T_{mp}, T_{tx,D}, \mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_n$
Output: $\mathbf{T}_R, \mathbf{T}_N, T_u, \mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_n$
 1: $T_{req,total} \leftarrow 0$, $D_{mdata} \leftarrow getDSDur()$
 2: Initialize allocable request sets $\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_n$
 3: while any of $\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_n$ is not empty do
 4:     for $i = 1 \to n$ do
 5:         $T_r \leftarrow$ the first request slot in $\mathbf{R}_i$
 6:         if $T_r + T_{req,total} < D_{mdata}$ then
 7:             add $T_r$ to $T_{req,total}$, insert $T_r$ into $\mathbf{A}_i$
 8:         else
 9:             exit while loop
10:         end if
11:     end for
12: end while
13: for $i = 1 \to n$ do
14:     Find the first available slots for the slots in $\mathbf{A}_i$, and
        store the results in $\mathbf{S}_i$
15: end for
16: Find the first available slots for requests from members,
    the next schedule update time, notifications to members
    in order, which are stored in $\mathbf{T}_R$, $\mathbf{T}_N$, $T_u$, respectively
```

B. Nodal Throughput Model

Because of the spatial-temporal difference, in COD-TS a node may start to send when the intended receiver is sending or receiving. Thus, it is complicated to derive an accurate nodal throughput model for COD-TS. For simplicity, in the model we assume a node cannot send until T_{mp} time after the current sender accomplishes the transmissions. This assumption will give the lower bound of the nodal throughput.

For clarity, $D_{p,req}$, $D_{p,nf}$, and $D_{p,ds}$ are used to denote the total durations of slots for request, notification, and data sending, respectively. According to section IV-A, $D_{p,req}$ and $D_{p,nf}$ are

$$\begin{aligned} D_{p,req} &= (N-1)\left(T_{tx,req} + T_{mp} + T_{guard}\right) \\ D_{p,nf} &= (N-1)\left(T_{tx,nf} + T_{mp} + T_{guard}\right) \end{aligned} \tag{26}$$

where $T_{tx,req}$ and $T_{tx,nf}$ are the transmission times of the request packet, and notification packet, respectively. For simplicity, we name them as total control packets, and the corresponding duration $D_{p,ctrl}$ for delivering them is

$$D_{p,ctrl} = D_{p,req} + D_{p,nf} \tag{27}$$

If $D_{p,ds}$ is also known, the duration of one round D_R is

$$D_R = D_{p,ctrl} + D_{p,ds} \tag{28}$$

Because the cluster head only assigns data sending slots to the cluster members who are senders, there will be n nodes participating in the data sending phase. Also note that each node tries to request sending times for all queued data packets such that they can send out all packets in this round, so there must be a steady state of D_R as long as the generated packets do not exceed the channel capacity. In the steady state, nodes in each round can send out all packets generated in the previous round, i.e., D_R should meet

$$D_{p,ds} = n\lambda D_R T_{tx,D} + nT_{mp} \quad if\ n\lambda T_{tx,D} < 1 \tag{29}$$

Then, combining Eq. 28 and 29 gives the closed-form of D_R

$$D_R = \frac{D_{p,ctrl} + nT_{mp}}{1 - n\lambda T_{tx,D}} \tag{30}$$

However, when λ is high, the channel may be saturated. In such case, the duration of the cluster round is subject to the upper bound D_{max}. Considering this constraint gives

$$D_R = min\left\{\frac{D_{p,ctrl} + nT_{mp}}{1 - n\lambda T_{tx,D}}, D_{max}\right\} \tag{31}$$

Note that D_{max} must be the smaller one when $n\lambda T_{tx,D} \rightarrow 1$ because the result of Eq. 30 approximates to infinity in this case. Then, with Eq. 28 $D_{p,ds}$ is obtained as

$$D_{p,ds} = min\left\{\frac{D_{p,ctrl} + nT_{mp}}{1 - n\lambda T_{tx,D}}, D_{max}\right\} - D_{p,ctrl} \tag{32}$$

Considering the assumption introduced at the beginning of this section, the assigned sending time slots for each sender contain T_{mp}. Excluding this part, the rest time is D_{mdata}. Thus, N_D, the total number of data packets sent in $D_{p,ds}$, is

$$N_D = \frac{D_{p,ds} - nT_{mp}}{T_{tx,D}} \tag{33}$$

Accordingly, the amount of data received during $D_{p,ds}$ is $N_D L_D$, which is also the amount of data delivered during D_R. Therefore, the nodal throughput of COD-TS is

$$\Lambda = \frac{N_D L_D}{N D_R} = \left(1 - \frac{nT_{mp} + D_{p,ctrl}}{D_R}\right)\frac{L_D}{N\left(L_D/B + T_{pre}\right)} \tag{34}$$

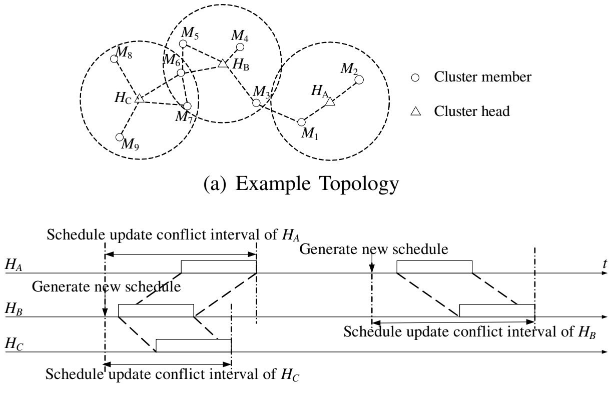
According to Eq. 34, Λ is determined by D_R when n, N and L_D are fixed. And the larger D_R is, the more Λ is closer to its upper bound $\frac{L_D}{N(L_D/B+T_{pre})}$, which is close to the maximum nodal throughput B. Thus, COD-TS is expected to achieve a high throughput.

C. Extending COD-TS to Multi-hop Networks

In order to analyze the capability of time sharing-based MAC, we have introduced how COD-TS works within one cluster. In this part, we will extend COD-TS to multi-hop case.

Essentially, if the schedule can avoid the four types of collisions described in [13], [20], it can guarantee that a member's packet successfully arrives at the intended receiver. Note that all collisions in ad hoc networks are either caused by one-hop neighbors or hidden terminals. Therefore, in order to eliminate collisions, a cluster head needs to know the latest schedule of the member's neighbors within two hops when generating schedule. Benefiting from the cluster structure, a cluster head can obtain such information by exchanging up-to-date schedule with its **n**eighboring **c**luster **h**eads (NCHs). Specifically, if a node is within two hop distance from another node, then their cluster heads are NCHs to each other. In Fig. 9(a), for example, H_B is H_A's NCH, but H_C is not. To allow NCHs to exchange their schedules, a cluster head can increase its transmission power. This is feasible since most existing modems support transmission power adjustment [17], [27], [28]. And then the schedule exchange packets can be sent in the CDMA approach like that described in [29].

Fig. 9(b) shows the schedule exchange among NCHs. For clarity, we call the packet for exchanging schedule among NCHs as schedule exchange packet. In order to avoid schedule update collision, each cluster head cannot generate or notify its new schedule during the **s**chedule **u**pdate **c**onflict **i**ntervals (SUCIs). Specifically, as demonstrated in Fig. 9(b), a cluster head's SUCI is the interval between the timepoint when its NCH generates new schedule and the timepoint when the corresponding schedule exchange packet arrives at this node. This rule can ensure that there is no schedule update collision while the communication round of each cluster is independent to others. After getting a schedule exchange packet, a cluster head maps and marks all slots on its own timeline, which is used for finding new sending slots as described in IV-A.


M8
M6
M5
M4
H_B
M3
H_A
M2
H_C
M7
M9
M1
○ Cluster member
△ Cluster head
Schedule update conflict interval of H_A
Generate new schedule
H_A
t
Generate new schedule
H_B
Schedule update conflict interval of H_B
H_C
Schedule update conflict interval of H_C

(a) Example Topology

(b) Schedule exchange and SUCI

Fig. 9. Schedule exchange among neighboring cluster heads

When generating the new schedule, a cluster head has the update-to-date schedule information of all members' two-hop neighbors, so it can guarantee the new schedule is collision-free. The scheduling algorithm is similar to Algorithm 1, and the only difference is that it avoids the overlap with the SUCIs to find the suitable slots for T_u and schedule exchange packet. Then, cluster members follow the schedule to send packets to neighboring nodes either within or out of the same cluster.

We also introduce id-based schedule initialization to prevent both packet collisions and schedule collisions among the initial schedule exchange packets. Concretely, a cluster head cannot generate its initial schedule or send out the first schedule exchange packet before it receives the schedule exchange packets from all NCHs which possess smaller ids.

In the multi-hop case, a cluster head may receive ACKs to nodes in neighboring clusters. In this case, it simply piggybacks the ACKs in the schedule exchange packet and the corresponding NCH will help to deliver.

D. Impact of Packet Errors on COD-TS

From the description above, we can see that COD-TS is vulnerable if any schedule exchange packet is lost. Although COD-TS can prevent packet collisions, a packet loss can still occur due to poor channel condition. When such a situation arises, the schedule would be messed up and the next schedule exchange packet may not be delivered correctly. This would be a diaster to COD-TS. In order to handle this case, besides of the regular schedule exchange, each cluster head is required to periodically sends the update-to-date schedule to its NCHs. The period is predefined and much longer than the communication round, so it does not introduce too much overhead and each cluster can avoid schedule collision with the reserved slots for these packets.

With this mechanism, even if a cluster head fails to decode a schedule exchange packet and accordingly causes collisions, the collision-free schedule can be guaranteed again after receiving a periodical schedule exchange packet from the same NCH.

V. Performance Evaluation

In this section, we conduct simulations to validate the nodal throughput models. We also compare the performance of various MAC protocols in multi-hop network scenarios to justify the advantages of time sharing-based MAC. The simulation platform is Aqua-Sim [30], an NS-2 based underwater acoustic network simulator.

In the network scenario for model validation, there are 8 nodes consisting of 4 pairs of sender and receiver within a one-hop network. In the multi-hop network scenario, 80 nodes are uniformly distributed in network of size $6000m \times 6000m \times 20m$, and each node randomly picks up one neighbor as its receiver. In both scenarios, all nodes are configured with the same transmission range, 1100 meters. Each sender generates data packets of size $400B$ following a Poisson process with the same λ parameter. In addition, nodes in all evaluated schemes maintain a long queue which never overflows so that it can match the infinity queue assumption of the models. Due to space limitations, only the results of nodal throughput, the most important metric of the low bandwidth featured networks, are presented to quantitatively show the performance.

A. Model Validation

From Fig. 10, we can see that all these three nodal throughput models well coincide with the corresponding simulation results, which justifies the correctness of our models. Since the collision probability directly decides the nodal throughput in Slotted ALOHA and RTS/CTS-based MAC, the validation of the nodal throughput models can also justify the collision probability models.

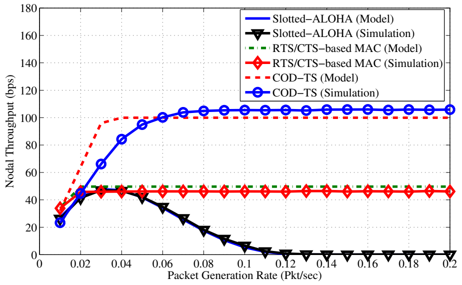


Fig. 10. Comparison of nodal throughput models and simulation results

As for COD-TS, the nodal throughput is smaller than our developed lower bound model when λ is low. This is reasonable because it takes a long time to reach the steady state of the communication round with a small λ. Before reaching the steady state, the communication round is relatively short and therefore the control packets incur significant overhead. This actually degrades the average measured throughput. With a large λ, more packets can be ready at MAC layer in a short period, and thus the low throughput achieved by the short rounds does not impact the average value too much. Therefore, we can see the measured throughput is exactly lower bounded by the developed model when $\lambda > 0.05$.

B. Evaluation of Nodal Throughput in Multi-hop Networks

1) Nodal Throughput with Varying λ: We vary λ from $0.01\ pkt/s$ to $0.2\ pkt/s$ of step $0.01\ pkt/s$ to compare the performance of Slotted-ALOHA, Slotted-FAMA and COD-TS. The comparisons are shown in Fig. 11. Note that hidden terminal problem is involved in multi-hop network scenario, so the nodal throughputs of all three solutions are lower than that measured in one-hop network scenario.

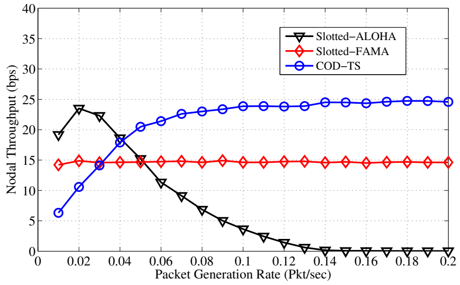


Fig. 11. Nodal throughput with varying λ

As shown in Fig. 11, Slotted-ALOHA achieves the best performance when λ is low. This is because Slotted ALOHA has the lowest overhead. When λ is low, there are fewer collisions and thus most data packets can be delivered. However, Slotted-ALOHA cannot effectively prevent collisions, so the collisions become heavier and accordingly the throughput approaches 0 as λ increases.

As for Slotted-FAMA, its throughput almost remains the same. This is determined by its handshaking scheme. Because of the low transmission rates and long preamble issues, the arrived data packets exceed the processing capability of Slotted-FAMA even though λ is low. At this time, its packet queue is full and the data packets are sent out with a fixed rate.

From Fig. 11, we can see that the throughput of COD-TS is relatively low with a small λ. This is because COD-TS utilizes a series of control packets to enable its scheduling scheme. When λ is low, the communication round is short and thus these control packets, especially the schedule exchange packets, cause significant overhead, which severely reduce the available time for data packets. As λ becomes higher, more data packets can be sent in each round, so the impact of the overhead becomes insignificant. As a result, COD-TS achieves higher throughput and outperforms the other two solutions when λ is larger than $0.05pkt/s$. However, limited by D_{max} which is introduced to guarantee the fairness among clusters, the throughput of COD-TS finally approaches to a constant.

2) Nodal Throughput with Varying Packet Size L_D: In this set of simulation, $\lambda = 0.08\ pkt/s$ and L_D is increased from $300\ B$ to $800\ B$ of step $100\ B$. Simulation results in Fig. 12 demonstrate that COD-TS outperforms the other two.

From Fig. 12, we can see the nodal throughput of both COD-TS and Slotted-FAMA rises as L_D increases. This is because T_{pre} is fixed. Specifically, the larger L_D is, the more

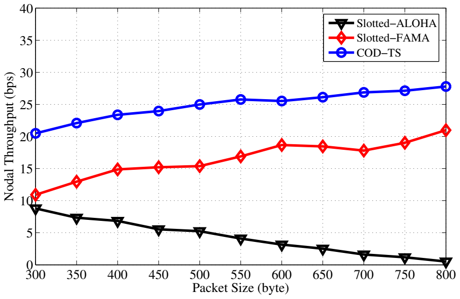


Fig. 12. Nodal throughput with varying packet size

data is transmitted in a packet. As a result, more data bytes are delivered in the unit time, and thus the nodal throughput of COD-TS and Slotted-FAMA goes up.

It is observed that the curve of Slotted-FAMA is not smooth. This is caused by its slot mechanism. Since the data packet transmission time is converted to the number of slots, a data packet may require an extra slot for transmission even if L_D is increased by only one byte. Considering one byte more contribution and the extra one slot overhead, the throughput must be lower. Then before requiring another slot, the nodal throughput increases again as L_D is larger. Thus, the shape of the curve of Slotted-FAMA is stair.

As for Slotted-ALOHA, when L_D is larger, the slot duration, which is also the time for transmitting one data packet, becomes longer correspondingly. As a result, more packets are queued, which further increases the probability that multiple nodes send packets in the same slot. Therefore, the nodal throughput decreases as L_D increases.

VI. Conclusions and Future Work

In this paper, we first discuss two newly identified acoustic modem characteristics, long preambles and low transmission rates, which severely degrade the performance of existing MAC protocols. In order to show the impact, we develop the collision probability and nodal throughput models for a random access-based MAC and a handshake-based MAC respectively. Based on the models and evaluation results, we believe that a time sharing-based approach can achieve better performance. Toward this end, we propose COD-TS and develop its nodal throughput model. Both analysis and simulation results justify that our time sharing-based MAC is a promising solution toward practical MAC design.

Future Work We will implement COD-TS in real modems (e.g., Aqua-fModem [28]) . We plan to conduct a series of field tests (in both lakes and seas) to evaluate and fine-tune the design, shooting for a functioning MAC protocol in the real world.

References

[1] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater Acoustic Networks," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 72–83, 2000.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–281, 2005.

[3] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications," *IEEE Network*, vol. 20, no. 3, pp. 12–18, 2006.

[4] J. Partan, J. Kurose, and B. N. Levine, "A Survey of Practical Issues in Underwater Networks," in *Proceedings of ACM WUWNet*, 2006, pp. 11–24.

[5] L. Liu, S. Zhou, and J.-H. Cui, "Prospects and Problems of Wireless Communication for Underwater Sensor Networks," *Wiley Wireless Communications and Mobile Computing*, vol. 8, no. 8, pp. 977–994, 2008.

[6] Z. Jiang, "Underwater acoustic networks–issues and solutions," *Internaltion Journal of Intelligent Control and Systems*, vol. 13, no. 3, pp. 152–161, 2008.

[7] L. F. M. Vieira, J. Kong, U. Lee, and M. Gerla, "Analysis of Aloha Protocols for Underwater Acoustic Sensor Networks," in *Poster abstract in ACM WUWNet*, 2006, pp. 1–2.

[8] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "Aloha-based MAC protocols with collision avoidance for underwater acoustic networks," in *Proceedings of INFOCOM*, 2007, pp. 2271–2275.

[9] C. L. Fullmer and J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," in *Proceedings of ACM SIGCOMM*, Cambridge, MA, USA, 1995, pp. 262 – 273.

[10] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC Protocol for Underwater Acoustic Networks," in *Proceedings of IEEE/MTS OCEANS*, 2006, pp. 1–7.

[11] J. Yackoski and C.-C. Shen, "UW-FLASHR: Achieving High Channel Utilization in a Time-based Acoustic MAC Protocol," in *Proceedings of ACM WUWNet*, 2008, pp. 59–66.

[12] C. Petrioli, R. Petroccia, and J. Potter, "Performance evaluation of underwater mac protocols: From simulation to at-sea testing," in *IEEE OCEANS*, 2011, pp. 1–10.

[13] C.-C. Hsu, K.-F. Lai, C.-F. Chou, and K. C.-J. Lin, "ST-MAC: Spatial-Temporal MAC Schedulging for Underwater Sensor Networks," in *Proceedings of IEEE INFOCOM*, 2009, pp. 1827–1835.

[14] K. B. K. II, P. Djukic, and P. Mohapatra, "Stump: Exploiting position diversity in the staggered tdma underwater mac protocol," in *IEEE INFOCOM*, 2009, pp. 2961–2965.

[15] J. Ma and W. Lou, "Interference-aware Spatio-Temporal Link Scheduling for Long Delay Underwater Sensor Networks," in *Proceedings of IEEE SECON*, 2011, pp. 1–9.

[16] Z. Peng, S. Le, M. Zuba, H. Mo, Y. Zhu, L. Pu, J. Liu, and J.-H. Cui, "Aqua-TUNE: A Field Testbed for Underwater Networks," in *Proceedings of IEEE OCEANS*, Jun. 2011, pp. 1–9.

[17] Teledyne-Benthos, "Acoustic Telemetry Modems: User's Manual," 2006.

[18] M. K. Park and V. Rodoplu, "UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 3, pp. 710–720, 2007.

[19] J. Partan, J. Kurose, B. N. Levine, and J. Preisig, "Spatial Resue in Underwater Acoustic Networks," UMASS, Dept. of Computer Science Technical Report: UM-CS-2010-045, Tech. Rep., 2010.

[20] Y. Guan, C.-C. Sheng, and J. yackoski, "MAC Scheduling for High Throughput Underwater Acoustic Networks," in *Proceedings of IEEE WCNC*, 2011, pp. 1–6.

[21] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I–Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, 1975.

[22] Y. Xiao, Y. Zhang, J. H. Gibson, G. G. Xie, and H. Chen, "Performance analysis of ALOHA and p-persistent ALOHA for multi-hop underwater acoustic sensor networks," *Cluster Computing*, vol. 14, no. 1, pp. 65–80, 2011.

[23] L. Lipsky, *QUEUEING THEORY: A Linear Algebraic Approach*, 2nd ed. New York: Springer-Verlag, 2008.

[24] J. Liu, Z. Wang, M. Zuba, Z. Peng, J.-H. Cui, and S. Zhou, "JSL: Joint Time Synchronization and Localization Design with Stratification Compensation in Mobile Underwater Sensor Networks," in *Proceedings of IEEE SECON*, 2012, pp. 317–325.

[25] Y. Zhu, Z. Zhou, Z. Peng, M. Zuba, and J.-H. Cui, "An efficient georouting aware mac protocol for underwater acoustic networks," *ICST Transactions on Mobile Communications and Applications*, vol. 11, no. 7–9, 9 2011.

[26] S. Bandyopadhyay and E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," in *Proceedings of IEEE INFOCOM*, 2003, pp. 1713–1723.

[27] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms," in *IEEE Oceans*, 2005, pp. 1086 – 1092.

[28] H. Yan, L. Wan, S. Zhou, Z. Shi, J.-H. Cui, J. Huang, and H. Zhou, "DSP based Receiver Implementation for OFDM Acoustic Modems," *Elsevier Journal on Physical Communication*, vol. 5, no. 1, pp. 22–32, 2012.

[29] K. Kredo and P. Mohapatra, "A Hybrid Medium Access Control Protocol for Underwater Wireless Networks," in *Proceedings of ACM WUWNet*, 2007, pp. 33–40.

[30] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. Cui, Z. Shi, Y. Fei, and S. Zhou, "Aqua-Sim: An NS-2 Based Simulator for Underwater Sensor Networks," in *IEEE/MTS OCEANS*, 2009, pp. 1 – 7.