Handling Triple Hidden Terminal Problems for Multichannel MAC in Long-Delay Underwater Sensor Networks

Zhong Zhou, *Student Member, IEEE*, Zheng Peng, *Student Member, IEEE*, Jun-Hong Cui, *Member, IEEE*, and Zaihan Jiang, *Senior Member, IEEE*

Abstract—In this paper, we investigate the multichannel MAC problem in underwater acoustic sensor networks. To reduce hardware cost, only one acoustic transceiver is often preferred on every node. In a single-transceiver multichannel long-delay underwater network, new hidden terminal problems, namely, multichannel hidden terminal and long-delay hidden terminal (together with the traditional multihop hidden terminal problem, we refer to them as "triple hidden terminal problems"), are identified and studied in this paper. Based on our findings, we propose a new MAC protocol, called CUMAC, for long-delay multichannel underwater sensor networks. CUMAC utilizes the cooperation of neighboring nodes for collision detection, and a simple tone device is designed for distributed collision notification, providing better system efficiency while keeping overall cost low. Analytical and simulation results show that CUMAC can greatly improve the system throughput and energy efficiency by effectively solving the complicated triple hidden terminal problems.

Index Terms—Underwater acoustic sensor network, multichannel, medium access control, energy efficiency.

1 Introduction

Underwater acoustic sensor networks have received rapidly growing interests in last several years [1], [2], [3], [4], [5], [6], [7]. On the one hand, underwater sensor networks have a wide range of aquatic applications, such as pollution monitoring, mine-countermeasure, offshore exploration, and submarine detections. On the other hand, the adverse underwater environments pose grand challenges for efficient communication and networking.

Electromagnetic signal does not work well due to its quick attenuation in conductive media like water. Acoustical waveform is the only one that could propagate long distance in underwater environment in practices. However, there are some fundamental differences between underwater environment and the terrestrial counterpart, which will result in significant difference of the network topology and protocol design from RF networks. First, the propagation speed of acoustic signals in water is about $1.5 \times 10^3 \mathrm{m/s}$, which is five orders of magnitude lower than the radio propagation speed (3×10^8 m/s). This causes significant propagation delay in underwater communication. Second, the available bandwidth is extremely limited, and it is distance/frequency dependent. For example, there are only a few hundred kHz for short range, few 10 kHz for medium distance, and few hundred Hz for long range. However, since the center frequency is usually at low band, the bandwidth is relatively "wide" compared with it. Hence, acoustic signal could not be treated as narrow band signal anymore. Third, multipath spread is severe. It could be as long as few hundred symbols, which results in a significant frequency-selective signal distortion. Fourth, Doppler spread/shift, due to the dynamic nature of the water medium and motion of the nodes, is relatively high. All these make underwater acoustic channel one of the most difficult media for information exchange. Consequently, building underwater sensor networks encounters grand challenges at almost every level of the protocol stacks, among which efficient medium access control (MAC) is one of the most fundamental issues.

In this paper, we investigate the MAC protocols for underwater acoustic networks. Due to the long propagation delay of acoustic channels, directly adapting the MAC protocols designed for terrestrial radio networks to underwater environments would be very inefficient. This new challenge for MAC in underwater networks has recently inspired a significant amount of research efforts, see references [8], [9], [10], [11], [12], [13], [14], [15]. All these studies focus on single-channel network scenarios.

Recent research on multichannel MAC protocols, which use multiple channels in parallel for data transmission, shows us that such a parallelism can greatly improve the network throughput, decrease channel access delay, and lower energy consumption [16], [17], [18]. The fast development of underwater acoustic communications [19] has also made it possible to utilize multiple acoustic channels in parallel. "AquaNetwork" modem from DSPCOMM [20], for example, is such a product that could be purchased off-the-shelf. Pioneering work in [21], [22], and [23] have also

- *Z. Zhou, Z. Peng, and J.-H. Cui are with the Computer Science & Engineering Department, University of Connecticut, 371 Fairfield Way, Unit 2155, Storrs, CT 06269. E-mail: {zhz05002, zhengpeng, jcui}@engr.uconn.edu.*
- *Z. Jiang is with the US Naval Research Laboratory, Acoustic Division, Code 7120, Washington, DC 20375. E-mail: zaihan.jiang@nrl.navy.mil.*

Manuscript received 6 Mar. 2010; revised 11 Nov. 2010; accepted 31 Dec. 2010; published online 2 Feb. 2011.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-03-0112.
Digital Object Identifier no. 10.1109/TMC.2011.28.

shown us the benefits of multichannel in underwater acoustic networks.

Before we can fully enjoy the benefits of the multichannel technology, however, some critical issues should be noticed and solved. First, MAC protocols for single-channel networks cannot be directly used in multichannel networks because of their low efficiency [24], [16], [25]. Second, the high expense of underwater transceivers (usually several hundred to thousand dollars each [26]) make the multitransceiver-based MAC protocols [21], [18], [24] very costly. Lastly, the triple hidden terminal problems (which include two new problems: multichannel and long-delay hidden terminal problems and the traditional multihop hidden terminal problem, to be discussed in Section 2) inherent in the single-transceiver multichannel long-delay underwater environment should be well addressed. Without careful design, these problems can greatly degrade the overall system performance.[1] In summary, the new multichannel long-delay network scenario calls for innovative solutions in the MAC layer.

In this paper, we propose a protocol, called Cooperative Underwater Multichannel MAC (CUMAC), for long-delay underwater sensor networks. By using a single transceiver, we can alleviate the budget burden of the underwater system. Our work is the first to identify the triple hidden terminal problems, i.e., multihop, multichannel, and long-delay hidden terminal problem that are inherent in the single-transceiver multichannel long-delay network scenario. We solve these complicated problems with the introduction of a cooperative collision detection scheme. An inexpensive tone device (as in [15], and [27]), is adopted to further improve the system efficiency. Our CUMAC solution features low cost, high network throughput, low energy consumption, and on-demand channel assignment. Compared with traditional RTS/CTS-based MAC protocols and Aloha-based random MAC protocols, our protocol can greatly reduce the collisions and improve the system performance. Compared with the static TDMA/CDMA-based MAC protocols, our protocol assigns channel on demand more flexibly to dynamic network environments.

The rest of this paper is organized as follows: in Section 2, we discuss the triple hidden problems. Then in Section 3, we present the system model. After that, we describe and analyze CUMAC in detail in Sections 4, 5, and 6. Simulation results are presented in Section 7. Finally, we review some related work in Section 8, followed by our conclusions and future work in Section 9.

2 Triple Hidden Terminal Problems

RTS/CTS-based multichannel MAC protocols have been extensively researched for terrestrial radio networks [24], [18], [25], [28]. In these studies, the communication link is divided into one control channel and multiple data channels and channel assignment is integrated into the RTS/CTS handshaking process on the control channel. It has been shown that RTS/CTS like coordination-based multichannel MAC protocols greatly outperform the Aloha-like multichannel MAC protocols [21]. For single-transceiver multichannel long-delay underwater networks, however, simple RTS/CTS negotiation approaches are not efficient. This is because, except for the traditional multihop hidden terminal problem for the single channel network, they will suffer from two new hidden terminal problems that are inherent in the new network scenario: multichannel and long-delay hidden terminal problems.

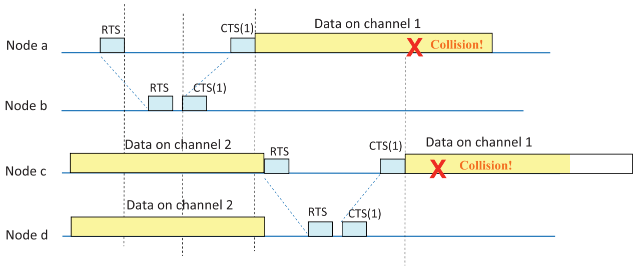


Fig. 1. Multichannel hidden terminal problem.

2.1 Multichannel Hidden Terminal Problem

Multichannel hidden terminal problem was first discovered in [16] for nodes with single transceivers. If each node has only one transceiver, it can work either on the control channel or on a data channel, but not on both. This essentially causes the multichannel hidden terminal problem. As shown in Fig. 1, when nodes a and b do handshaking on the control channel, nodes c and d are communicating with each other on data channel 2. Thus, nodes c and d do not know the channel that is selected by nodes a and b (i.e., data channel 1). Later, when node c wants to send a data packet to node d, it initiates a handshaking process on the control channel. Since node d does not know that channel 1 has been used by others at this time, it may select the same channel and thus create a collision.

Multichannel hidden terminal problem can be solved by having one dedicated transceiver listening on the control channel continuously and thus, at least two transceivers are needed on every node. Because of the high expense of underwater acoustic transceivers, it is quite costly to equip multiple transceivers on each node. To reduce system cost, our CUMAC solution targets at the network scenario where there is only one acoustic transceiver on every node.

2.2 Long-Delay Hidden Terminal Problem

The long propagation delays of the underwater acoustic channel introduce another kind of hidden terminal problem. As shown in Fig. 2, at the beginning, all nodes are listening to the control channel. Node a starts its handshaking process with node b on the control channel and then selects channel 1 for communication. Later, nodes c and d also negotiate on the control channel for their data transmission. Let us assume the CTS message from node b arrives at node d after it selects its own data channel and sends its CTS message back to node c. In this case, node d does not know that the channel has already been used by node b. It may select channel 1 to communicate with node c and thus create a collision. We call this delay-related

[1] In long-delay underwater networks, the hidden terminal problems in the multichannel environment is significant. The results from [21] shows us that the network throughput degrades more than 30 percent in a normal network setting due to the hidden terminal problems.

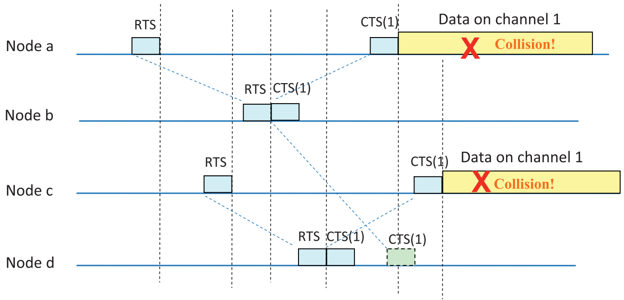


Fig. 2. Long-delay hidden terminal problem.

hidden terminal problem as "long-delay hidden terminal problem." This problem is usually negligible in terrestrial radio networks due to the high propagation speed of radio signal. For long-delay underwater acoustic networks, however, this problem has to be well addressed.

In short, new solutions are demanded in order to effectively solve the triple hidden terminal problems in single-transceiver multichannel long-delay underwater networks. Our CUMAC protocol is proposed to tackle these new challenges.

3 System Model

In this paper, we use the following multichannel underwater acoustic network model:

- There are multiple channels in the network, with each channel having equal bandwidth. One control channel is dedicated for control message exchange. Other channels (i.e., data channels) are for data transmission. Every node knows the common control channel and listens to it when there are no data to send or receive.
- Each node has only one acoustic transceiver, which can dynamically switch to different channels including control channel and data channels. Further, each node is equipped with an inexpensive out-of-band tone device which can send and receive tone signal (similar to the tone hardware in [15] and [27]).
- For the ease of presentation and analysis, we use a circular transmission range R for each node. Two nodes do not interfere with each other if their distance is larger than R. The propagation speed of the acoustic signal is v. And thus, the maximal propagation time T for the acoustic signal from a node to reach its transmission range equals $\frac{R}{v}$. If a node receives signal from more than one nodes simultaneously on the same channel, a collision happens and packets get lost.
- Every node remains static or moves slowly. Every node is assumed to know its own location information. This is a reasonable assumption as localization has been widely explored for underwater acoustic networks [29], [30], [31], [32]. Furthermore, location information is also needed by networking functions such as Geo-routing [33] and applications such as environmental monitoring [34], [3], [6]. Therefore, this assumption will not increase the system cost much. Strict synchronization is not needed though some localization protocols may require synchronization service.

4 Protocol Description

In this section, we first give an overview of CUMAC, and then we present the two key techniques it employs: cooperative collision detection and tone pulse sequence. After that, we discuss how CUMAC effectively solves the triple hidden terminal problems.

4.1 Protocol Overview

In CUMAC, when a node has packets to send, it will initiate a channel negotiation process which consists of a RTS/Beacon/CTS control message exchange on the control channel. During this process, the receiving node cooperates with its neighboring nodes for channel selection and collision detection. After a data channel has been successfully selected, both the sender and the receiver switch to the selected data channel for data transmission. Therefore, CUMAC can be divided into two phases: RTS/Beacon/CTS Channel Negotiation and Data Transmission. We brief each phase as follows:

- RTS/Beacon/CTS Channel Negotiation. Before sending data, a node first sends a RTS message out, which includes some useful information such as its available data channel set, on the control channel. After receiving the RTS message, the receiver will select a data channel based on the received RTS message and its own perceived channel conditions. Then it broadcasts a *beacon message* on the control channel and seek the cooperation from its neighboring nodes for collision detection. We term this technique as *cooperative collision detection* (to be descried in Section 4.2). The receiver will then start a timer and wait for the responses from its neighbors. If no collisions are detected before the timer times out, the receiver sends a CTS message back to the sender to inform it of the data channel that has been selected. Otherwise, the receiver selects another data channel and broadcasts a beacon message again, performing another round of cooperative collision detection. Note that in a dense network, the multiple responses from neighboring nodes will easily congest the control channel. To overcome this problem, CUMAC employs a technique called *tone pulse sequence* with the assistance of an out-of-band tone device on each node. This technique will discussed in Section 4.3.
- Data Transmission. This phase of the protocol is quite straightforward. After a successful RTS/Beacon/CTS channel negotiation process, both the sender and the receiver switch to the selected data channel for data transmission. Afterward, both of them will switch back to the control channel.

4.2 Cooperative Collision Detection

4.2.1 Basic Idea

In CUMAC, a cooperative collision detection scheme is designed to alleviate the hidden terminal problems

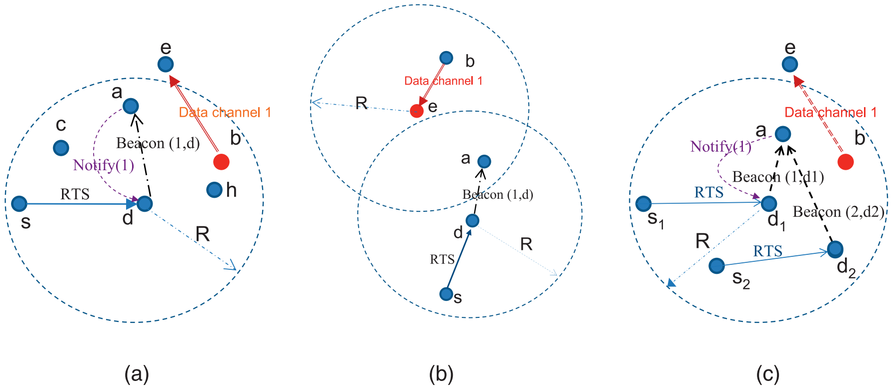


Fig. 3. Cooperative collision detection: (a) The basic idea. (b) Invalid multihop scenario. (c) Problem with tone signal.

discussed in Section 2. In this scheme, neighboring nodes cooperate with each other to select a suitable data channel and detect potential collisions on the selected data channel. The key idea of cooperation is the following: since a node will listen to the control channel if it does not have data to send or receive, it can obtain the channel usage information of its neighboring nodes by overhearing the control channel, and such information can be effectively leveraged by other neighboring nodes to detect any potential collision.

Taking Fig. 3a as an example. While node b is sending data to node e on data channel 1, node s tries to communicate with node d and sends a RTS message to it. After node d receives the RTS from node s, it selects a data channel (e.g., channel 1) which is deemed to be free by itself at this time.[2] Node d then broadcasts a beacon message on the control channel to inform its neighboring nodes of the channel it will use for data transmission. However, since both nodes b and e are on data channel 1, they can neither hear the beacon message from node d on the control channel nor tell node d about the potential collision on data channel 1. Fortunately, as shown in the figure, some of the neighboring nodes of both nodes b and d might know the status of data channel 1 by overhearing the messages on the control channel. For example, in Fig. 3a, node a is a neighbor of both nodes b and d. And a knows that data channel 1 has been used by the communication between nodes e and b through overhearing their handshaking process. Thus, when node a hears the beacon message from node d, it can notify node d of the potential collision on data channel 1. In a dense network, such a cooperative collision detection scheme is expected to be quite effective in suppressing the multichannel hidden terminal problem. As the example in Fig. 3a, nodes c, a, and h can all judge the possible collision on data channel 1, and any of them can inform node d to avoid the collision.

To put this cooperative collision detection idea into practice, however, two critical problems, *heterogeneous collision region* and *redundant collision notification*, need to be addressed.

4.2.2 Heterogeneous Collision Region

In a multihop network scenario, the collision region of every node is different. Then how can a node tell there will be collisions on its neighboring nodes? For example, as shown in Fig. 3b, node a knows that nodes e and b are using data channel 1 and it later receives a beacon message from node d saying data channel 1 is selected for data transmission. Fig. 3b shows us that these two communication pairs (e and b, s and d) are far enough to use data channel 1 simultaneously without collision, which is quite different from the case in Fig. 3a. However, how can the cooperative node (node a in the examples) identify different situations? We term this problem as *heterogeneous collision region problem*.

In CUMAC, location information of each node is needed and could be included in the control messages such as RTS/Beacon/CTS. Thus, a node can obtain the locations of all its known data senders and receivers. Then, it can calculate the distances between these nodes and judge whether collision among them will happen or not. For example, as in Fig. 3a, node a knows the locations of nodes s, d, b, and e by overhearing the control channel. Node a thus can calculate the distances between these nodes and find that nodes b and d are close enough to interfere with each other.

4.2.3 Redundant Collision Notification

In a dense network, it is possible that multiple neighboring nodes perceive the same collision event, which stimulates all of them to notify the ongoing communication pair. Multiple collision notification messages thus might be generated. For example, as shown in Fig. 3a, besides node a, nodes c and h might also find the potential collision on data channel 1 and will notify node d. Apparently, it is unwise to transmit all these notification messages on the control channel directly since these messages will overwhelm the channel. This will significantly increase the collision probability of the control channel and degrade the system performance. Therefore, such redundant collision notification should be suppressed.

To address this problem, in CUMAC, an out-of-band tone device is required on each node and it is used for the collision notification purpose. The simplest method is that if a node detects collision, it sends out a tone signal for

[2] Node d might miss the handshaking process between nodes b and e because of the single transceiver architecture and thus it does not know that channel 1 has already been used.

notification. However, such a naive method does not work in the multichannel network scenario. As multiple data channels are used concurrently, multiple tone signals might exist simultaneously, then how can a node with only one tone device differentiate multiple tone signals for different channels and different communication pairs? For example, as shown in Fig. 3c, nodes s_1 and d_1 are trying to communicate with each other on data channel 1 and nodes s_2 and d_2 are planning to use data channel 2 at the same time. Both d_1 and d_2 send out a beacon message to ask for help from their neighbors. At this time, node a finds collisions on data channel 1 and then it will send its tone signal out to inform node d_1. However, at the same time, d_2 is also waiting for the response from its neighboring nodes and it can also receive the tone signal for node d_1. Because tone signals cannot be decoded, how can node d_2 know that the received tone signal is not for it? In CUMAC, we introduce the technique of *tone pulse sequence* to solve this problem.

4.3 Tone Pulse Sequence

4.3.1 Basic Idea

Tone pulse is defined as a short tone signal and **tone pulse sequence** is a sequence of n periodic tone pulses,[3] where n is a predefined system parameter. In CUMAC, a node that has detected collision on a data channel will send out a tone pulse sequence at a specific time. With careful design, tone pulse sequences destined for different nodes will arrive at a node at different time. If a node knows the schedule of the tone pulse sequence destined for itself, this node can pick its tone pulse sequence out and make a correct decision.

In CUMAC, a node, say i, which has received a RTS and selected a data channel for its data receiving will first broadcast a beacon message, on the control channel. This beacon message includes this node's location information and the periodicity τ_i of its tone pulse sequence. Node i then listens for the tone pulse every τ_i starting from the time it sends its beacon message out. And we define these time points where node i listens for the tone pulse as the **detection points** of node i. For example, if node i sends its beacon message out at time t_i, then $t_i + k\tau_i, 1 \leq k \leq n$ are its detection points.

After a neighboring node receives a beacon message, it will first estimate whether there will be collisions based on the beacon message and its perceived data channel conditions. If a neighboring node detects a collision, based on the location information, it will calculate the time to send its tone pulse sequence in order to make them arrive at node i on the expected detection points. For example, if the distance between node i and a neighboring node j is d_{ij} and at time t_j node j receives the beacon message, then, at time $k\tau_i - 2\frac{d_{ij}}{v} + t_j, (k = 2\lceil\frac{d_{ij}}{v\tau_i}\rceil)$, node j will send out its first tone pulse, where v is the propagation speed of the underwater acoustic signal. After that, node j will send out its tone pulse every τ_i until n tone pulses (i.e., a tone pulse sequence) are sent out.

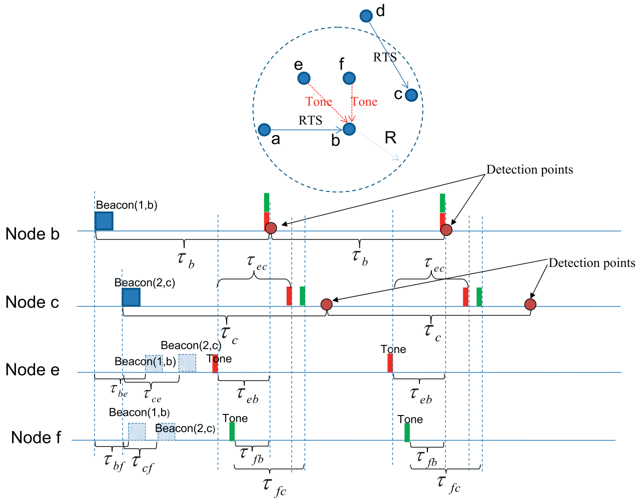


Fig. 4. Mechanism of tone pulse sequence.

Tone pulses from different sources destined for the same node will arrive at the destination on its detection points. If multiple tone pulses arrive at the same time, they add up and the destination can detect the existence of the tone pulses. Since the duration of a tone pulse is small and every node randomly chooses the periodicity for its tone pulse sequence, the probability that the tone pulse sequence destined for one node overlaps with the tone pulse sequences destined for another node is quite small.

For example, as shown in Fig. 4, nodes b and c select channels 1 and 2, respectively, for data receiving. And they send out beacon messages to inform neighboring nodes of their selected channels. Both nodes e and f find that channel 1 is not available and send out tone pulse sequences back to node b. Since node b's location is known by e and f through its beacon message, e and f can calculate the time to send their tone pulse sequences out. And as shown in this figure, the tone pulses from different neighbors (node e and f) will arrive at node b at its detection points. From Fig. 4, it is clear that the tone pulse sequences from node e and f arrive at node c at times different from the detection points of node c. Thus, node c will not mistake the tone pulse for node b as its own.

4.3.2 Why Tone Pulse Sequence?

In CUMAC, multiple tone pulses (a tone pulse sequence) are needed to reduce its error probability. As shown in Fig. 5, nodes a and c send out their beacon messages and node b finds potential collision for node a. While for node c, no potential collisions are detected on its selected data channel. Although the tone pulse that is sent by b at time $t_a + \tau_a - t_{ab}$ arrives at node a at time $t_a + \tau_a$, it arrives at node c at time $t_c + \tau_c$, which happens to be a detection point of node c. It is clear that if node c makes its judgment simply based on this single tone pulse, it will make a wrong decision.

Therefore, in CUMAC, every node which has detected possible collisions for node i will send out n tone pulses (a tone pulse sequence) periodically according to its received periodicity τ_i to let them arrive at node i at its detection points. With high probability, different nodes will

[3] Tone devices which can send short tone signal for underwater sensor networks have been explored in [27] and [15]. In this paper, we assume that tone signal is omnidirectional. No strict synchronization is needed among tone devices.

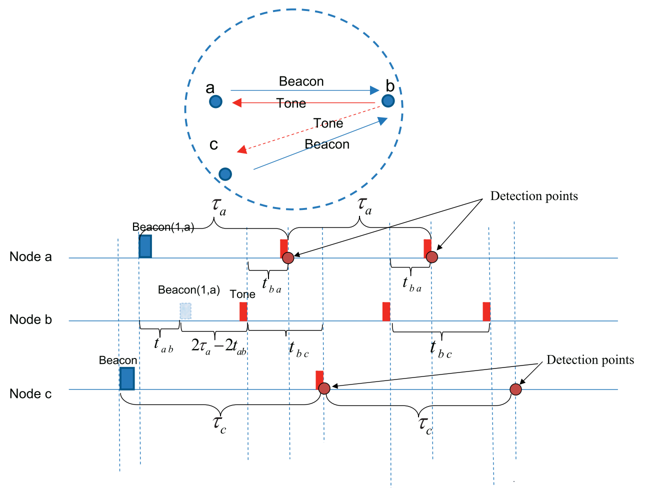


Fig. 5. Strength of tone pulse sequence.

have different τ_i. Thus, it is highly unlikely that all n tone pulses destined for one node fall on the detection points of other nodes. For example, as shown in Fig. 5, although the first tone pulse destined for node a falls on one detection point of node c, the second tone pulse does not, which prevents the error from happening on node c. In Section 6, we will analyze the false alarm and false approval probability of our protocol.

4.4 Discussions

With the import of the RTS/Beacon/CTS handshaking process and the cooperative collision detection scheme, CUMAC can greatly alleviate, if not eliminate, the triple hidden terminal problems. As the traditional multihop hidden terminal problem (which has been well studied in the literature) can be easily addressed by a typical RTS/CTS handshaking process, in the following, we mainly discuss how CUMAC handles the two new hidden terminal problems: multichannel and long-delay hidden terminal problems.[4]

First, the multichannel hidden terminal problem is caused by the incomplete channel usage information available to each node, which is the result of the single transceiver architecture. In CUMAC, the cooperative collision detection scheme asks for help from neighboring nodes. Collisions on the data channels can only happen when all neighboring nodes do not perceive the channel condition correctly, which is highly unlikely in a relatively dense network. Thus, the multichannel hidden terminal problem is greatly suppressed in CUMAC.

Second, the long-delay hidden terminal problem originates from the fact that the long propagation delays of underwater acoustic channels prevent nodes from getting updated data channel conditions timely. In CUMAC, the RTS/Beacon/CTS handshaking process effectively addresses this problem. Before a data channel is used by a node for data communication, this node first reserves this channel and sends a beacon message out to inform the neighboring nodes. Only after neighboring nodes get this information and agree, this channel can be used. Thus, the long-delay hidden terminal problem is also alleviated.

5 Protocol Details and Implementation

In CUMAC, a data channel might be in the following three states:

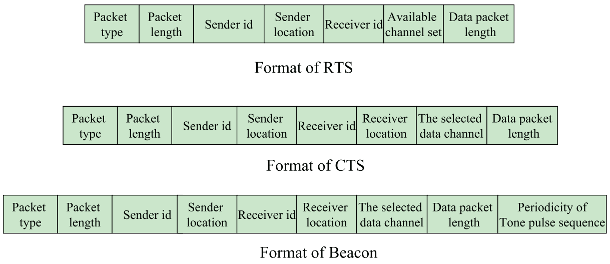
- Free: this channel is free and can be used or reserved.
- Reserved: this channel is reserved by some nodes for data packet transmissions.
- Busy: this channel is busy in transmitting data packets.

Every node will maintain a channel usage table to record the usage information of data channels. Every item in the channel usage table records some useful information of a data channel such as the channel status, the length of its data packet, the source node, the destination node, and their locations. As shown in Fig. 3b, it is possible that a node (for example, node a in Fig. 3b) might perceive multiple noninterfering communication pairs on a data channel in its neighboring area. These pairs should be recorded as different items in the channel usage table. Thus, multiple items might exist in a node's channel usage table for a channel.

At the beginning, no data channels have been used and all channels are in free status. Thus, the channel usage table of every node is empty. When a node has packets to send, it sends a RTS message out, which includes its available channel set, its id, location, and data length. The available channel set of a node consists of the free data channels which do not appear in its channel usage table. When the intended receiver, let's say node i, receives the RTS, it selects one channel randomly from the **intersection** of its own available channel set and that of the source node which is contained in the RTS message. Then, Node i sends a beacon message out, which includes the selected channel, the data packet length, the ids and locations of the sender and the receiver. The periodicity of its tone pulse sequence τ_i, which has been discussed in Section 4.3, is also included in the beacon message. After that, node i inserts a corresponding item in its channel usage table and sets the channel in the reserved status.

The receiver (node i) then waits for the tone pulses and sets a timer which will expire after $2T + n\tau_i$, where T is the maximal propagation delay and n is the number of tone pulses in a tone pulse sequence. If less than n tone pulses are received at its detection points before it times out, this node believes that there will be no collisions on the selected data channel and will send a CTS message out. The CTS messages includes the selected data channel, the data packet length, the locations as well as the ids of the sender and the receiver. The packet format of all control messages is shown in Fig. 6. Node i also changes the corresponding item in its channel usage table to busy. Then, this node will switch to the data channel. After receiving the CTS message, the data sender also switches to the data channel and transmits its data. After the data

[4] The introduction of the cooperative collision scheme lengthens the handshaking process for the data packets, which may increase the packet delay. Since underwater networks are mostly energy and bandwidth constrained, packet delay is not so critical, especially for delay insensitive applications such as scientific data collection.



Packet type	Packet length	Sender id	Sender location	Receiver id	Available channel set	Data packet length

Format of RTS

Packet type	Packet length	Sender id	Sender location	Receiver id	Receiver location	The selected data channel	Data packet length

Format of CTS

Packet type	Packet length	Sender id	Sender location	Receiver id	Receiver location	The selected data channel	Data packet length	Periodicity of Tone pulse sequence

Format of Beacon

Fig. 6. Packet format of control messages.

transmission, both of the sender and the receiver will switch back to the control channel.

If n or more tone pulses are received on the detection points of the receiver before it times out, the receiver will delete the corresponding item in its channel usage table. Then it selects another channel and repeats the above process. After k channel selection trials, if this node still cannot get a right channel which is approved by all its neighbors for data receiving, it gives up and will not send beacon messages out any more. On the other hand, when the sender times out on receiving CTS message, it will randomly backoff for some time and resend a RTS to the receiver. The optimal backoff strategy is beyond the scope of this paper and in our simulation, we adopt the exponential backoff strategy.

A node that receives a beacon message correctly detects whether collisions will happen on the selected data channel based on its own channel usage table. If no collision is detected between this communication event and other perceived communication events, a new item will be inserted into this node's channel usage table and set its status to be reserved. If a collision is detected between this communication event and other on-going data packets (i.e., busy items in the node's channel usage table), this node will send out a tone pulse sequence to inform the corresponding receiver to select another channel for data receiving. If this communication event is predicted to collide with other in-negotiation transmission events (reserved items in the node's channel usage table), this node will compare the new communication event with the existing colliding ones. The one that sends out its beacon message first will be deemed to be effective. This node will then send tone pulse sequences out to notify the noneffective communication pairs and delete the corresponding items in its channel usage table.

If an item in the channel usage table corresponding to the receiver i has been in the reserved status for more than $2T + n\tau_i$ and its CTS message has not been received, this item will be deleted. When the CTS message is received, this item will be changed to the busy status. It should be noted that the RTS/CTS message includes the length of the subsequent data packets, which indicates how long this channel will be occupied for data transmissions. Every node that has received the CTS message will set the corresponding item in its channel usage table to the busy status for the period for the data transmission and then delete it. After a receiver sends its CTS to the sender and switches to the selected data channel, it will start a timer which will expire after $2T$. If it does not receive any data from the sender before it times out, the CTS message might get lost and the receiver switches back to the control channel. It then deletes the corresponding item in its channel usage table.

The working process of CUMAC on the sender, receiver, and the neighboring nodes are shown in Fig. 7.

One example of the process of CUMAC is shown in Fig. 8. Here, node a wants to send a packet to node b. And at this time, nodes c and d are also listening to the control channel. Node a first sends out a RTS message to b and after b gets RTS message, it will select a channel (e.g., data channel 1) and send a beacon message out. Then, node b inserts a new item about data channel 1 in its channel usage table and set its status reserved. After nodes c and d receive the beacon message, both of them judge there will be collisions on the selected data channel and will send out their tone pulses at the specified time. The tone pulses from nodes c and d will arrive at node b at its detection points. Thus, node b knows that the selected data channel 1 cannot be used. Node b then deletes the corresponding item about data channel 1 from its channel usage table and selects another data channel (e.g., data channel 2). After receiving the beacon message for data channel 2 from nodes b, c, and d do not detect collision at this time. Then, both nodes c and d will insert a new item about data channel 2 into their channel usage tables. No tone pulses will be sent out now. After some time, node b sends out its CTS message to inform node a about the selected data channel 2. And finally, both nodes a and b switch to the data channel 2 for data transmission.

6 Analysis

With the import of the cooperative collision detection and the RTS/Beacon/CTS handshaking process, CUMAC can greatly alleviate, if not eliminate, the double hidden terminal problems.

First, the multichannel hidden terminal problem is caused by the incomplete channel usage information available to every node, which is the result of its single transceiver architecture. Our cooperative collision detection in CUMAC asks for help from neighboring nodes. Collisions on the data channel can only happen if all neighboring nodes do not perceive the channel condition correctly, which is highly unlikely considering the relatively high node density of sensor networks. Thus, the multichannel hidden terminal problem is greatly suppressed in CUMAC.

Second, as for the delay related hidden terminal problem, it originates from the fact that the long propagation delay of the underwater acoustic channel prevents nodes from getting updated data channel conditions. However, the RTS/Beacon/CTS handshaking process of CUMAC addresses this problem well. Before a data channel is used by a node for data communications, this node first reserves this channel and sends a beacon message out to inform the neighboring nodes. Only after neighboring nodes get this information and agree, this channel can be used. Thus, the delay-related hidden terminal problem is also alleviated.

The benefits of our protocol do not come without a cost. Two kinds of errors, false alarm and false approval, come along with our cooperative collision detection process.

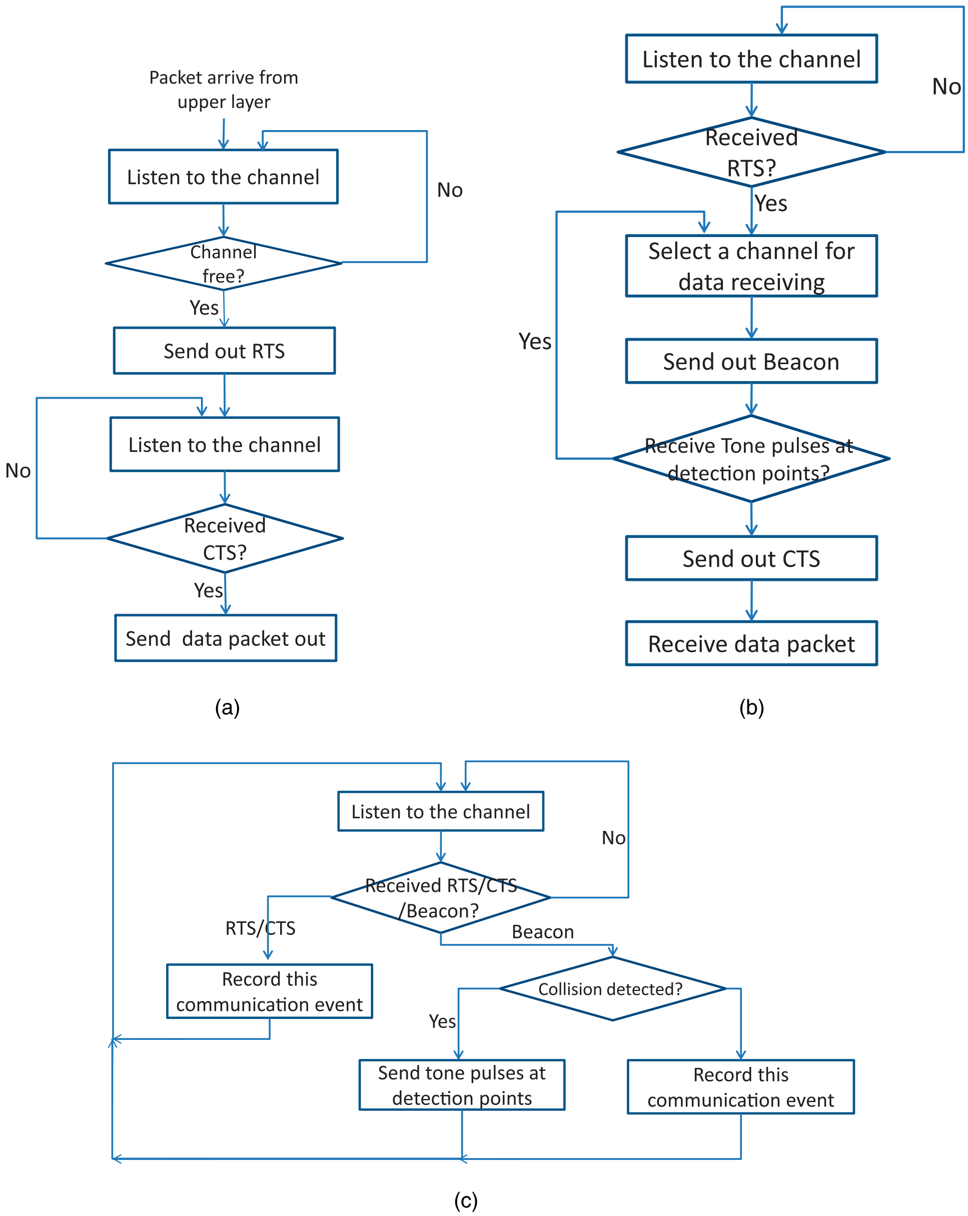


Fig. 7. Flowchart of CUMAC. (a) Sender side. (b) Receiver side. (c) Other neighboring nodes.

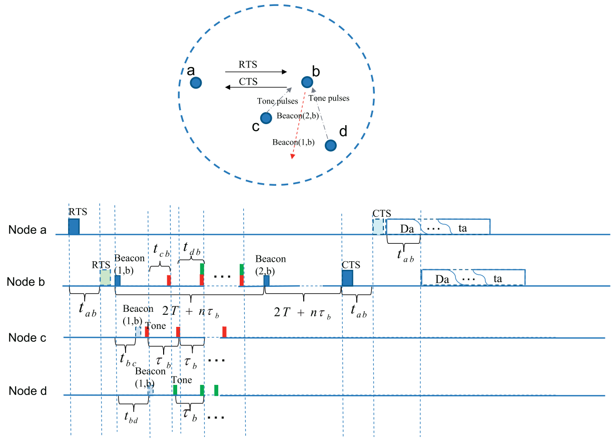


Fig. 8. An example of CUMAC process.

However, as we will show later, these kinds of errors are bounded and negligible compared to the benefits of CUMAC. Next, we will analyze and discuss them.

To simplify our analysis, we make the following assumptions. Nodes follow a uniform distribution in the network and the node density is ρ. The average neighbors of a node is thus $\frac{4\pi R^3}{3}\rho$. The traffic of every node is assumed to be an identical independent (iid) Poisson process with the parameter λ. Set the duration of the tone pulse to be t_{tp}. The periodicity of the tone pulse sequence for node i is τ_i. And τ_i is in a range of $[\tau_{\min}, \tau_{\max}]$ and takes discrete values among $\tau_{\min} + 2k \times T_{tp}, (0 \leq k \leq s)$, where $s = \frac{\tau_{\max} - \tau_{\min}}{2t_{tp}}$, with equal probability as in our simulations.

6.1 False Alarm Probability

Here, **false alarm** means that although there will be no collision on the selected data channel, CUMAC mistakenly believes that there will be collisions and asks for data channel reselection. False alarm increases the traffic on the control channel, which will degrade the system's performance.

In CUMAC, for a node (e.g., node i) that has sent out its beacon message and is waiting for the response from its neighbors, if it receives n tone pulses which are destined for other nodes at its n detection points, node i will make a false decision and false alarm happens. To get the upper bound for the false alarm, we assume that all neighbors of node i are on the control channel (this is the worst case because some nodes might be on the data channel and will not be able to send tone pulse sequences). Since nodes are uniformly distributed in the network, the tone pulses destined for other nodes arrive at node i following a uniform distribution.

If another node, say node j, selects the same periodicity as that of node i and a tone pulse destined for node j arrives at node i at the detection point of node i, it is self-evident that the following tone pulses for node j will also arrive at node i at its detection points. If more than n tone pulses destined for node j falls in the detection periods of node i, false alarm happens on node i. The probability that another node j selects the same periodicity of its tone pulse sequence as that of node i will be $1/s$ since every node selects the periodicity for its tone pulse sequence independently. And the probability that a tone pulse destined for node j arrives at node i on the detection point of node i will be $\frac{2T_{tp}}{\tau_i}$. Thus, we can get the probability $P_{B_{ij}}$ that node i makes a false alarm based on the tone pulse sequence which is destined to another node j as

$$P_{B_{ij}} = \frac{1}{s}\left(1 - \left(1 - \frac{2T_{tp}}{\tau_i}\right)^{\frac{4}{3}\pi R^3 \rho}\right). \tag{1}$$

Since we assume that the input traffic for every node follows an identical independent Poisson process, thus, the probability $P(k)$ that there are k other ongoing channel negotiations which might interfere with node i can be written as

$$P(k) = \frac{\left(\frac{4}{3}\pi R^3 \rho \lambda 4T\right)^k}{k!} e^{-\left(\frac{4}{3}\pi R^3 \rho \lambda 4T\right)}, \tag{2}$$

where T is the maximal propagation delay. Since in our protocol, every node sends out its beacon message and chooses the periodicity of its tone pulse sequence independent of others, we can get the upper bound of the probability of this kind of false alarm $P(B)$ as

$$P_B = \sum_{k=1}^{\infty}(1 - (1 - P_{B_{ij}})^k)P(k). \tag{3}$$

The above false alarm is brought about by the tone pulse sequence destined for another node. However, in practice, it is possible that multiple tone pulses destined for different nodes happen to arrive at node i on its n detection points, which also leads to false alarm. By assuming that all neighbors are on the control channel, we can approximately derive the upper bound of the probability for this kind of false alarm P_A.

If there are other k on-going channel negotiations in the neighborhood of node i, at most $\frac{4}{3}\pi R^3 \rho k$ tone pulse sequences will be generated. Set $P(A/K = k)$ to be the probability that node i makes this kind of false alarm under the condition that there are other k channel negotiations in progress

$$P(A/K = k) = 1 - \sum_{l<n} C_{l}^{\frac{4}{3}\pi R^3 k\rho} \left(\frac{2T_{tp}}{\tau_i}\right)^l \left(1 - \frac{2T_{tp}}{\tau_i}\right)^{\frac{4}{3}\pi R^3 k\rho - l}. \tag{4}$$

Since the input traffic of every node is assumed to be independent Poisson process, we can get the

$$\begin{aligned} P_A &= \sum_{k=1}^{\infty} P(K = k)P(A/K = k) \\ &= \sum_{k=0}^{\infty} \frac{\left(\frac{4}{3}\pi R^3 \rho \lambda 4T\right)^k}{k!} e^{-\frac{4}{3}\pi R^3 \rho \lambda 4T} \\ &\quad \times \left(1 - \sum_{l<n} C_{l}^{\frac{4}{3}\pi R^3 k\rho} \left(\frac{2T_{tp}}{\tau_i}\right)^l \left(1 - \frac{2T_{tp}}{\tau_i}\right)^{\frac{4}{3}\pi R^3 k\rho - l}\right). \end{aligned} \tag{5}$$

The overall false alarm probability P_{f1} thus can be obtained as

$$P_{f1} \leq P_A + P_B. \tag{6}$$

Our simulation results in Section 7 show that the false alarm probability under the normal network condition is very small, usually less than 0.1.

6.2 False Approval Probability

False approval means that although there exist potential collisions on the selected data channel, CUMAC fails to detect it, which finally results in the collision on the data channel. Since data packets usually are much longer than control packets, collisions on the data channel will lead to relatively high bandwidth and energy wastage.

Because we use a cooperative collision detection scheme, a false approval only happens if no neighbors oppose to a false decision. Set p_c to be the probability that a node stays on the control channel and p_{col} to be the average collision probability on the control channel. A neighbor can correctly receive a control message if it is on the control channel and there is no collision on the control channel, the probability of which is $p_c(1 - p_{col})$. Only if a neighbor can receive the beacon/CTS message correctly from both colliding communication pairs, it can judge collisions. The false approval probability P_{f2} thus can be written as

$$P_{f2} = [1 - (p_c(1 - P_{col}))^2]^{\frac{4\pi R^3}{3}\rho}. \tag{7}$$

Next, we need to estimate the probability of p_c for one data packet transmission. Averagely, at least $\frac{1}{(1-p_{col})^2}$ times of RTS/Beacon/CTS exchange need to be performed, and every RTS/Beacon/CTS exchange will take at least $4T + t_{rts} + t_{cts} + t_b$, where t_{rts}, t_{cts}, and t_b are the transmission time for a RTS, CTS, and beacon message, respectively. Thus, we can get

$$p_c \geq \frac{\frac{1}{(1-p_{col})^2}(4T + t_{rts} + t_{cts} + t_b) + \frac{1}{\lambda}}{\frac{1}{(1-p_{col})^2}(4T + t_{rts} + t_{cts}) + t_b + \frac{1}{\lambda} + t_D}, \tag{8}$$

where t_D is the transmission time for a data packet. Here, $\frac{1}{\lambda}$ is included because a node will stay on the control channel when it has no traffic and the input traffic is a Poisson process with parameter λ. For the collision probability on the control channel p_{col}, as in [18], by assuming $t_{rts} = t_{cts} = t_b = t_c$, we can approximate aggregated traffic of RTS/Beacon/CTS on the control channel as a Poisson process with parameter $\lambda + 2(1 - p_{col})\lambda$. Thus, we can get

$$p_{col} = 1 - e^{-\frac{4\pi R^3}{3}\rho(1+2(1-pcol))\lambda t_c}, \tag{9}$$

which can be used to calculate p_{col} numerically.

7 Performance Evaluation

In this section, we evaluate the performance of our CUMAC via simulations.

7.1 Simulation Settings

Based on NS-2, UConn UWSN Lab has developed a simulation package, called Aqua-Sim, for underwater sensor networks [35]. We implemented CUMAC in Aqua-Sim. Both random and fixed topology networks are investigated. The channel model in Aqua-Sim is the same as in [36]. The average channel loss is given by

$$A(d, f) = d^a \beta(f)^d, \tag{10}$$

where d is the distance of the acoustic channel; a is the spreading factor ($1 \leq a \leq 2$); and $\beta(f)$ is the absorption coefficient which is determined by the frequency of the acoustic channel f.

Unless specified otherwise, the simulation parameters are as follows: there are eight channels in the network and the bandwidth of each channel is set to $1\ \mathrm{kbps}$. The propagation speed of the acoustic signal is $1500\ \mathrm{m/s}$. The transmission range of every node is set to be $500\ \mathrm{m}$. The length of a tone pulse is set to 2 ms. The periodicity of the tone pulse sequence for each node is in a range of $[12\ \mathrm{ms}, 60\ \mathrm{ms}]$ and takes discrete values among $12 + 4k, (0 \leq k \leq 12)$ ms with equal probability. The number of tone pulses in a tone pulse sequence is three. The average data packet length is 300 bytes. The average transmitting and receiving power of the acoustic transceiver is set to be 0.6 and 0.2 Watt. And the idle listening power is 0.02 Watt. Every simulation runs for 500 s and every data point is the average of 100 simulations.

For comparison purpose, we implemented two other multichannel MAC protocols, random multichannel MAC protocol (*random scheme* for short) and RTS/CTS-based multichannel MAC protocol (*RTS/CTS scheme* for short).

- *Random multichannel MAC protocol.* In this protocol, when a node has data to send, it first randomly chooses a data channel and then sends out a small control packet on the control channel to inform the receiver of its selection. After that, this node switches to the selected data channel and sends out its data packet immediately. If the receiver gets the control packet correctly, it switches to the data channel for data receiving. This random protocol is analyzed for underwater acoustic networks in [21].
- *RTS/CTS-based multichannel MAC protocol.* In this protocol, if a node has data to send, it starts with a RTS on the control channel to inform the receiver. After the RTS been received correctly, the receiver will send a CTS message which includes the selected data channel back to the sender. Both the sender and the receiver then switch to the selected data channel for data transmission. This protocol is similar to the protocol in [24], [18] which is designed for terrestrial sensor networks and analyzed in [21] for underwater acoustic networks.

For all three protocols, we measure two metrics: *average network throughput* and *average energy consumption per byte*. The first metric is defined as the average number of successfully transmitted data bytes per second. The second metric is obtained by dividing the overall energy consumption in the network[5] by the successful transmitted data bytes, which is measured in millijoule per byte.

7.2 Simulation Results

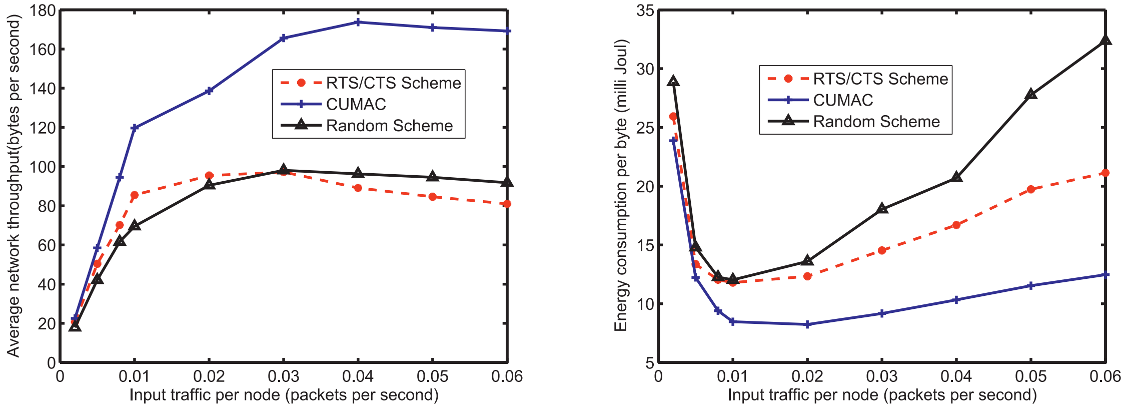
7.2.1 Results in Random Networks

We simulate a random network where 20 underwater nodes are uniformly distributed in a $300\ \mathrm{m} \times 300\ \mathrm{m} \times 300\ \mathrm{m}$ area. A sending node randomly chooses one of its neighbors as the receiver.

Impact of input traffic. In this set of simulations, we change the input traffic of every node from 0.002 to 0.06 packet/s. As shown in Fig. 9, for all three protocols, the throughput increases rapidly with the input traffic at first. And then, it decreases. For example, the throughput of CUMAC achieves its maximal when the input traffic increases to 0.04 packet/s. After that, because of the increasing collisions in the network, the throughput decreases slowly with the input traffic. Fig. 9a shows us that our CUMAC can achieve much higher throughput than the other two. For example, when the input traffic is 0.03 packet/s, the throughput of CUMAC is about 160 bytes/s. While that of the other two is less than 100 bytes/s.

From Fig. 9b, we can see that for all three protocols, the average energy consumption per byte decreases at first with the increase of the input traffic. This is because with the increase of the input traffic, less energy will be wasted on the idle listening (the idle listening power consumption in our simulation is 0.02 Watt). However, with the increase of the input traffic, the collision probability in the network will also increase, which results in the degradation of energy efficiency. Fig. 9b also shows us that CUMAC can achieve much higher energy efficiency than the other two. For the random scheme, a node sends its packets in a totally random manner. There is no method to suppress the collision in the network. Thus, its energy efficiency is the worst. For the RTS/CTS scheme, the RTS/CTS

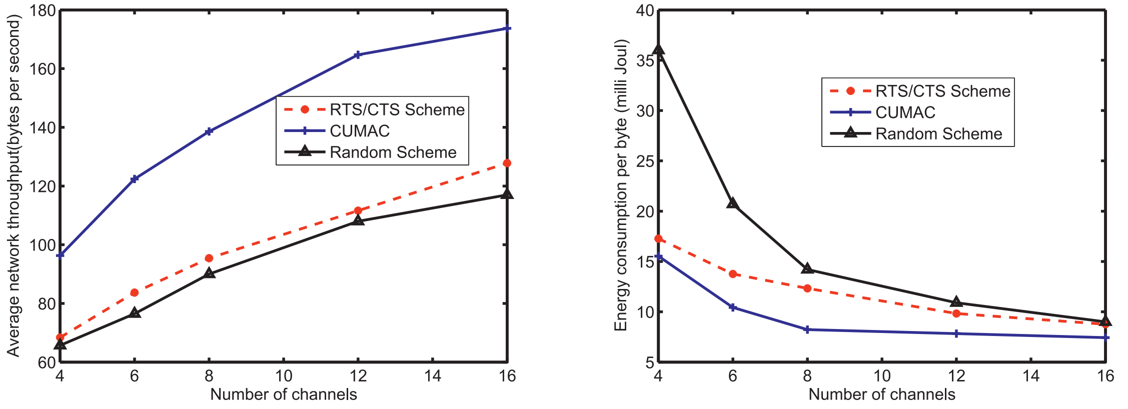
[5] The overall energy consumption includes all kinds of energy consumption in the network such as the energy consumption for data sending/receiving, beacon message sending/receiving and tone pulse sending/receiving.



(a)

(b)

Fig. 9. Performance comparison with different input traffic. (a) Average network throughput. (b) Average energy consumption per byte.



(a)

(b)

Fig. 10. Performance comparison with varying number of channels. (a) Average network throughput. (b) Average energy consumption per byte.

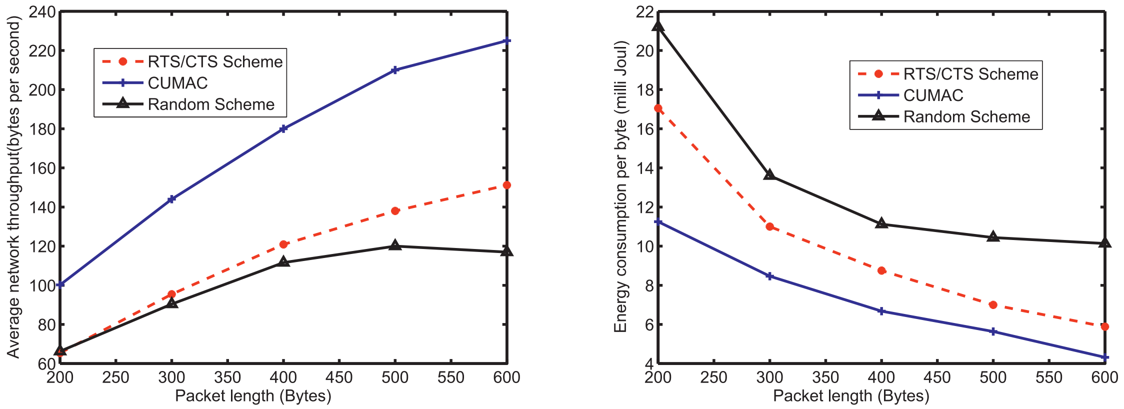
exchange before the data transmission reduces the collisions on the data channels to some extent. However, the RTS/CTS scheme suffers from the multichannel and the long-delay hidden terminal problems which are inherent in the single-transceiver multichannel long-delay underwater network. As for CUMAC, its cooperative collision detection scheme along with the tone pulse sequence mechanism suppresses the triple hidden terminal problems efficiently, which results in higher throughput and better energy efficiency.

Impact of number of channels. In this set of simulations, we set the input traffic of every node to 0.02 packet/s and change the number of channels from 4 to 16. Without surprise, as shown in Fig. 10, both the network throughput and the energy efficiency improve with the number of channels. This is because the more the data channels, the less the collision probability on the data channels since the net input traffic to every data channel will be reduced. However, the improvements on the system performance will slow down with the increase of the number of data channels. This is because the system's performance is jointly determined by the control channel and the data channels. With the increase of the number of data channels, the system will be more constrained by the control channel. Thus, the impacts of increasing the number of data channels become less and less. To further improve the system performance, the bandwidth of the control channel should change dynamically with the network conditions, which is an interesting research topic and will be studied in our future work.

Impact of data packet length. In this set of simulations, we fix the input traffic of every node to 0.02 packet/s and change the data packet length from 200 bytes to 600 bytes.[6] As shown in Fig. 11, in CUMAC and the RTS/CTS scheme, the network throughput increases monotonically with the packet length. This is reasonable since the longer a data packet is, the more data will be transmitted on the data channels for a successful handshaking process on the control channel. Consequently a higher throughput can be achieved.

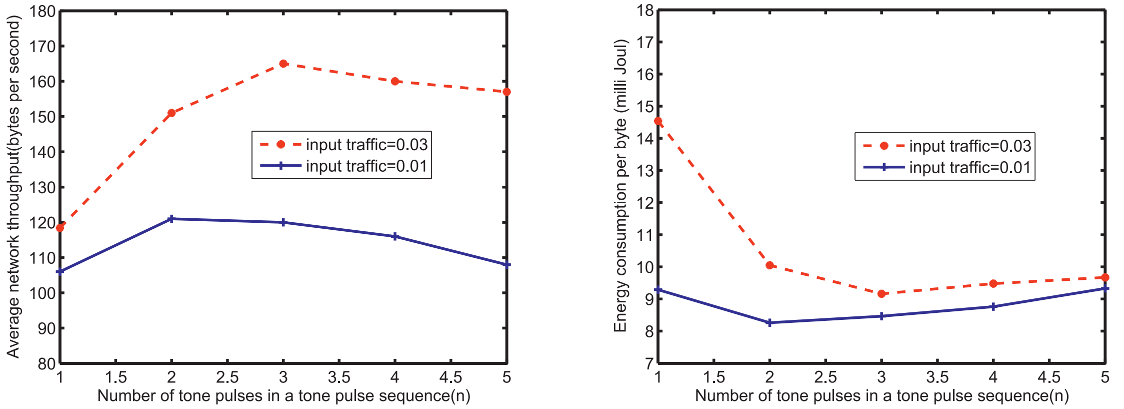
For the random scheme, when the packet length is small (less than 500 bytes), the network throughput increases

[6] Here, the packet length does not necessarily mean the length of one data segment, it might be the total length of a series of small data segments. Here, we define the packet length as the total data bytes followed a control messages exchange.



(a) (b)

Fig. 11. Performance comparison with varying data packet length. (a) Average network throughput. (b) Average energy consumption per byte.



(a) (b)

Fig. 12. Performance comparison with varying tone pulse sequence length. (a) Average network throughput. (b) Average energy consumption per byte.

slowly with the packet length, but decreases afterward. This is because for the random scheme, the length of data packets has double effects. On one hand, longer data packets may contribute to higher collision probability on the data channels, which might lead to the decrease of the throughput. On the other hand, potentially, with longer data packets, one successful data packet transmission contributes more throughput than the case with shorter data packets. When the average length of data packets is short, the second factor dominates the first, therefore the network throughput will increase. But with the increase of data packet length, the first factor plays a major role and thus the network throughput will decrease.

From Fig. 12b, we can also observe that for all three protocols, the average energy consumption decreases with the increase of the average data packet length, which means that the longer the data packet, the higher the energy efficiency. Thus, in practice, we may increase the length of data packets for high energy efficiency.

Impact of tone pulse sequence length. In this set of simulations, we change the number of tone pulses in a tone pulse sequence (n) from 1 to 5. Fig. 12 illustrates the strength of our tone pulse sequence mechanism. For example, when the input traffic is 0.03 packets/s, the throughput for the system with single tone pulse is only 118 bytes/s. By simply increasing the number of tone pulses in a tone pulse sequence to three, we can improve the throughput to more than 160 bytes/s. And at the same time, the system's energy consumptions also decrease. This is because our tone pulse sequence mechanism can effectively reduce the false alarm in the system and thus reduce the unnecessary renegotiation on the control channel.

From Fig. 12, we can see that when the length of the tone pulse sequence reaches some large values, the system performance decreases slowly. This is because, on the one hand, when the length of the tone pulse sequence is large, further increasing it cannot contribute much to the decrease of the false alarm in the system. On the other hand, the duration for a RTS/Beacon/CTS handshaking process in our CUMAC increases slowly with the increase of the tone pulse sequence length, which contributes to the reduction of the system throughput. It is clear in Fig. 12 that the benefits of our tone pulse sequence mechanism is more significant when the input traffic is heavy. This is reasonable since with the increase of the input traffic, more

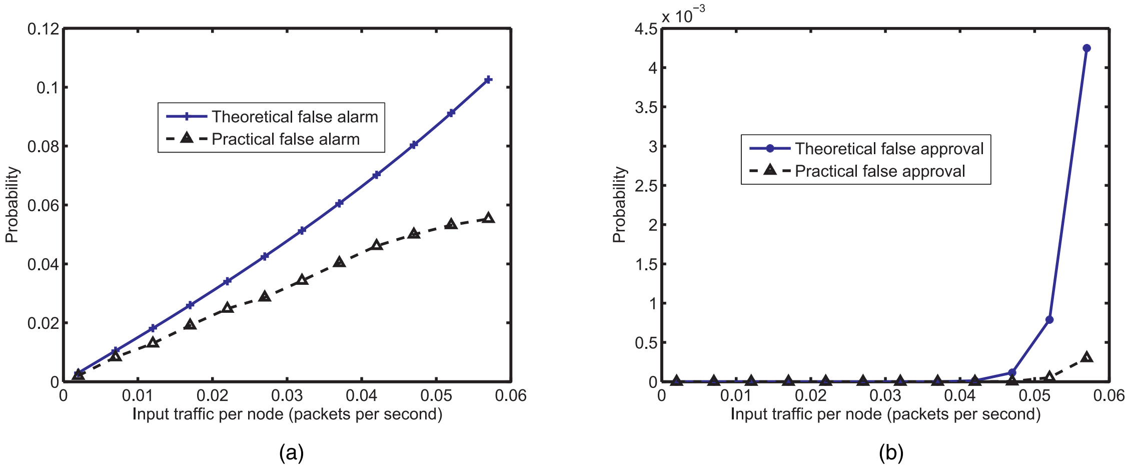


Fig. 13. False alarm and false approval: analysis versus simulation. (a) False alarm probability. (b) False approval probability.

RTS/Beacon/CTS handshaking processes will be initiated on the control channel. This will contribute to the increase of the false alarm probability. Consequently, our tone pulse sequence mechanism which aims to reduce the false alarm probability plays a more important role in the system.

False alarm and false approval probability. In this set of simulations, we change the input traffic of every node from 0.002 to 0.06 packets/s. Fig. 13 shows that with the increase of the input traffic, both the false alarm and the false approval probability increase. Compared to the false alarm, the value of false approval is much smaller (almost 0). This means that in a network with relatively high node density, the RTS/Beacon/CTS handshaking process of CUMAC can almost eliminate the collision on the data channel, which results in high energy efficiency. As in Fig. 13, the false alarm probability is low in most cases, usually, less than 10 percent. To further improve the performance of our protocol, we can dynamically adjust the bandwidth of the control channel to minimize the false alarm probability, which is another interesting research topic and is our future work.

Fig. 13b shows the false approval probability, which indicates the probability that our CUMAC fails to detect a collision. The false approval probability actually is the collision probability for our CUMAC. From Fig. 13b, we can see that the collision probability of CUMAC is very low, much less than 1 percent, which further demonstrates that our protocol can effectively suppress the collisions in the long-delay underwater network.

Fig. 13 demonstrates us that our theoretical analysis results in Section 6 can be safely served as the upper bounds for both errors. Our upper bounds are tight when the input traffic is low and they become looser with the increase of the input traffic. This is because in our analysis, we assume that all nodes are on the control channel, which can be approximately satisfied when the input traffic is low. However, with the increase of the input traffic, more and more nodes are involved in the data transmission on the data channels rather than control channel, which contributes to the looseness of our upper bounds in the heavy traffic-loaded network.

7.2.2 *Results in Star-Topology Networks*

In this set of simulations, we configure a multihop star network topology as shown in Fig. 14, where node 1 is the sink node. Nodes 8, 9, and 10 are the source nodes to generate traffic toward node 1. There are four channels in the network. A VBF-like routing protocol [33] is used here to route the traffic as show in Fig. 14. Nodes in the same interference area interfere with each other. This topology is quite similar to the topology around the sink node in a practical underwater network. For this topology, we use the end-to-end throughput to measure its performance, which is defined as the average number of successfully transmitted data bytes per second from the source nodes (i.e., 8, 9, and 10) to the sink node (node 1).

Fig. 15 demonstrates the advantages of CUMAC in such a practical multihop network environment. The performance of the random scheme in this network is poor (the end-to-end throughput is almost 0) and we ignore its results here. Compared to the RTS/CTS scheme, CUMAC can improve the end-to-end throughput by almost 40 percent with much higher energy efficiency. In this simulation, we also observed that although the simulated network is not dense (only 10 nodes), it gets saturated when the input traffic of every source node is only about 0.05 packet/s. One of the main reasons for such a quick saturation lies in the limited capacity of the

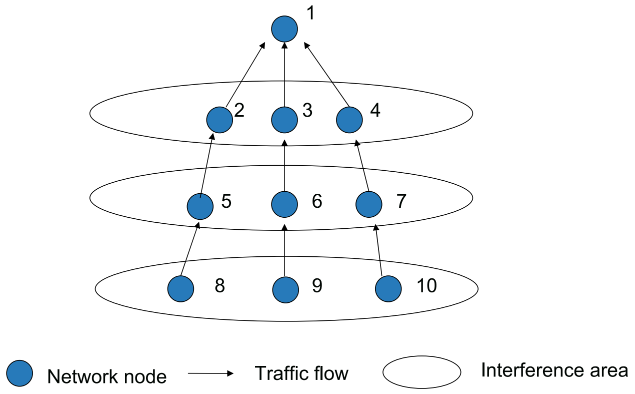
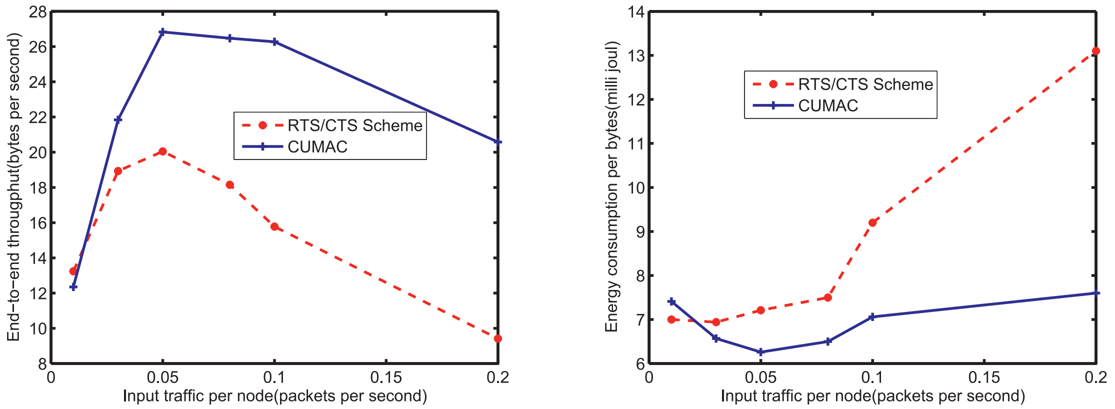


Fig. 14. A star topology multihop network.



(a) (b)

Fig. 15. Performance comparison with different input traffic in the star topology multihop network. (a) Average network throughput. (b) Average energy consumption per byte.

sink node. Here, although the sink node works as the traffic aggregator, it only has one acoustic transceiver. When the sink node is communicating with one node, the other nodes have to wait, which will greatly degrade the system performance. One way of solving this problem is to equip the sink node with multiple acoustic transceivers if the extra cost is allowable. We would like to explore this topic in our future work.

7.3 Summary

Our simulation results clearly show us that our CUMAC can achieve high performance in single-transceiver multichannel long-delay underwater sensor networks. Its RTS/Beacon/CTS handshaking process along with the cooperative collision detection can effectively suppress the hidden terminal problems. Although false alarm and false approval exist, they are bounded and usually negligible compared with the benefits of CUMAC. And the tone pulse sequence technique is an effective and efficient means to improve the performance.

8 Related Work

In this section, we first briefly review some related work on multichannel MAC protocols for terrestrial radio wireless networks. Then, we discuss some research on multichannel MAC protocols for underwater acoustic networks and show their differences from our work.

Multichannel MAC protocols have long been investigated for terrestrial radio wireless networks. In [37], the authors propose a receiver initiated multichannel MAC protocol based on frequency hopping for multihop wireless networks. Strict synchronization among nodes is needed for this protocol. The authors of [16] propose a multichannel MAC protocol with a single transceiver. They identify the multichannel hidden terminal problem and solve it with a synchronized MAC protocol which splits the time into channel negotiation phase and data transmission phase. In [18], the authors study the performance of multichannel MAC protocols with Aloha-like reservation on a dedicated control channel for wireless networks. In [17], the authors analyze and compare four different multichannel MAC protocols for single-hop wireless network. Their analysis and simulation results show that different protocols have different properties and thus are preferred in different network scenarios. All these studies are conducted for radio communication where propagation delay is negligible.

Multichannel MAC protocols for underwater acoustic networks have also aroused significant research interest recently. For example, in [22], a hierarchical multichannel MAC protocol is proposed for clustered underwater networks where TDMA is used for the intracluster communication and CDMA is used for the intercluster communication. Strict synchronization among all nodes is needed in this scheme. In [23], the authors utilize CDMA as the multiple access technique. A RTS/CTS handshaking scheme is employed for every channel before actual data transmission. In this scheme, CDMA spreading codes are distributed first by some predefined algorithm and every node is assumed to get a unique spreading code among its one-hop neighbors. In [21], the authors analyze two generalized multichannel MAC protocols for underwater sensor networks. The long propagation delay of underwater acoustic channel is taken into account in their analysis model. Their results demonstrate the great benefits of multichannel MAC protocols. However, in [21], at least two acoustic transceivers are assumed to be available on every node. Different from all the existing work, this paper considers a cost-effective network architecture where one and only one transceiver is needed on each node. We identify the triple hidden terminal problems which are inherent in the single-transceiver multichannel long-delay underwater scenario and propose a new protocol to effectively deal with them.

9 Conclusions and Future Work

In this paper, we identified and investigated the triple hidden terminal problems in single-transceiver multichannel long-delay underwater sensor networks. Based on our findings, we proposed CUMAC, a new multichannel MAC protocol. By employing a novel cooperative collision detection scheme and a tone pulse sequence technique, CUMAC handles the triple hidden terminal problems well and greatly reduces the collision in the network. Simulation results show that CUMAC can significantly improve the network throughput with high energy efficiency in both one-hop and multihop networks.

As for the future work, we would like to pursue our studies in three directions: 1) Investigate the optimal bandwidth allocation strategy for the control channel and data channels. 2) Implement CUMAC in underwater sensor network testbeds and evaluate its feasibility and practicality in the real world. 3) Investigate other collision free multichannel MAC protocols such as TDMA/CDMA-based MAC protocols for static underwater acoustic networks.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under CAREER Grant No. 0644190, Grant No. 0709005, Grant No. 0721834, and Grant No. 0821597, and the US Office of Navy Research under YIP Grant No. N000140810864. Part of this work was presented at IEEE INFOCOM 2010, San Diego, California.

REFERENCES

[1] I.F. Akyildiz, D. Pompili, and T. Melodia, "State-of-the-Art in Protocol Research for Underwater Acoustic Sensor Networks," *Proc. First ACM Int'l Workshop Underwater Networks (WUWNet '06)*, pp. 7-16, Sept. 2006.

[2] 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, May 2006.

[3] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research Challenges and Applications for Underwater Sensor Networking," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '06)*, pp. 228-235, Apr. 2006.

[4] J. Partan, J. Kurose, and B.N. Levine, "A Survey of Practical Issues in Underwater Networks," *Proc. First ACM Int'l Workshop Underwater Networks (WUWNet '06)*, Sept. 2006.

[5] I.F. Akyildiz, D. Pompili, and T. Melodia, "State of the Art in Protocol Research for Underwater Acoustic Sensor Networks," *ACM SIGMOBILE Mobile Computing and Comm. Rev.*, vol. 11, pp. 11-22, Oct. 2007.

[6] L. Liu, S. Zhou, and J.-H. Cui, "Prospects and Problems of Wireless Communications for Underwater Sensor Networks," *Wireless Comm. and Mobile Computing, Underwater Sensor Networks: Architectures and Protocols*, vol. 8, pp. 977-994, Aug. 2008.

[7] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater Acoustic Communications and Networking: Recent Advances and Future Challenges," *Marine Technology Soc. J.*, vol. 42, no. 1, pp. 103-116, 2008.

[8] M. Molins and M. Stojanovic, "Slotted FAMA: A MAC Protocol for Underwater Acoustic Networks," *Proc. OCEANS Conf.—Asia Pacific*, pp. 1-7, May 2006.

[9] P. Xie and J.-H. Cui, "Exploring Random Access and Handshaking Techniques in Large-Scale Underwater Wireless Acoustic Sensor Networks," *Proc. OCEANS Conf.*, Sept. 2006.

[10] P. Xie and J.-H. Cui, "R-MAC: An Energy-Efficient MAC Protocol for Underwater Sensor Networks," *Proc. Int'l Conf. Wireless Algorithms, Systems, and Applications (WASA '07)*, Aug. 2007.

[11] K.B. Kredo and P. Mohapatra, "A Hybrid Medium Access Control Protocol for Underwater Wireless Networks," *Proc. Second Workshop Underwater Networks (WUWNet '07)*, Sept. 2007.

[12] B. Peleato and M. Stojanovic, "Distance Aware Collision Avoidance Protocol for Ad-Hoc Underwater Acoustic Sensor Networks," *IEEE Comm. Letters*, vol. 11, no. 12, pp. 1025-1027, Dec. 2007.

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

[14] N. Chirdchoo, W.-S. Soh, and K.C. Chua, "Aloha-Based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks," *Proc. IEEE INFOCOM*, pp. 2271-2275, 2007.

[15] A. Syed, W. Ye, and J. Heidemann, "T-Lohi: A New Class of MAC Protocol for Underwater Acoustic Sensor Networks," *Proc. IEEE INFOCOM*, 2008.

[16] J. So and N. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver," *Proc. ACM MobiHoc*, pp. 222-233, May 2004.

[17] J. Mo, H.-S. Wilson, and J. Walrand, "Comparison of Multi-Channel MAC Protocols," *IEEE Trans. Mobile Computing*, vol. 7, no. 1, pp. 50-65, Jan. 2008.

[18] Y.S. Han, J. Deng, and Z.J. Haas, "Analyzing Multi-Channel Medium Access Control Schemes with ALOHA Reservation," *IEEE Trans. Wireless Comm.*, vol. 5, no. 8, pp. 2143-2152, Aug. 2006.

[19] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier Communication over Underwater Acoustic Channels with Nonuniform Doppler Shifts," *IEEE J. Oceanic Eng.*, vol. 33, no. 2, pp. 198-209, Apr. 2008.

[20] "AquaNetwork: Underwater Wireless Modem with Networking Capability," http://bwww.dspcomm.com/products_aquanetwork.html, 2011.

[21] Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, "Analyzing Multi-Channel MAC Protocols for Underwater Acoustic Sensor Networks," Technical Report UbiNet-TR08-02, Univ. of Connecticut, Dept. of Computer Science and Engineering, http://www.cse.uconn.edu/~jcui/publications.html, Aug. 2008.

[22] F. Salva-Garau and M. Stojanovic, "Multi-Cluster Protocol for Ad Hoc Mobile Underwater Acoustic Networks," *Proc. OCEANS Conf.*, pp. 91-98, 2003.

[23] H.-X. Tan and W.K.G. Seah, "Distributed CDMA-Based MAC Protocol for Underwater Sensor Networks," *Proc. IEEE 32nd Conf. Local Computer Networks*, pp. 26-36, 2007.

[24] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," *Proc. Int'l Symp. Parallel Architectures, Algorithms, and Networks (I-SPAN '00)*, May 2000.

[25] J.S. Pathmasuntharam, A. Das, and A.K. Gupta, "Primary Channel Assignment Based MAC(PCAM)A Multi-Channel MAC Protocol for Multi-Hop Wireless Netwroks," *Proc. IEEE Proc. Wireless Comm. and Networking Conf. (WCNC '04)*, pp. 1110-1115, 2004.

[26] B. Benson, G. Chang, D. Manov, B. Graham, and R. Kastner, "Design of a Low-Cost Acoustic Modem for Morred Oceanographic Applications," *Proc. First ACM Int'l Workshop Underwater Networks (WUWNet '06)*, pp. 71-78, Sept. 2006.

[27] A. Syed, J. Heidemann, and W. Ye, "Tones for Real: Managing Multipath in Underwater Acoustic Wakeup," *Proc. Seventh Ann. IEEE CS Conf. Sensor, Mesh and Ad Hoc Comm. and Network (SECON '00)*, Aug. 2010.

[28] A. Nasipuri and J. Mondhe, "Multi-Channel MAC with Dynamic Channel Selection for Ad Hoc Networks," technical report, http://www.ece.uncc.edu/~anasipur, 2004.

[29] Z. Zhou, J. hong Cui, and S. Zhou, "Localization for Large Scale Underwater Sensor Networks," *Proc. IFIP Networking*, pp. 108-119, May 2007.

[30] Z. Zhou, J. hong Cui, and A. Bagtzoglou, "Scalable Localization with Mobility Prediction for Underwater Sensor Networks," *Proc. IEEE INFOCOM*, May 2008.

[31] V. Chandrasekhar, W.K. Seah, Y.S. Choo, and H.V. Ee, "Localization in Underwater Sensor Network - Survey and Challenges," *Proc. First ACM Int'l Workshop Underwater Networks*, pp. 33-40, Sept. 2006.

[32] X. Cheng, H. Shu, Q. Liang, and D.H.-C. Du, "Slient Positioning in Underwater Acoustic Sensor Networks," *IEEE Trans. Vehicular Technology*, vol. 57, no. 3, pp. 1756-1766, May 2008.

[33] P. Xie, L. Lao, and J.-H. Cui, "VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks," *Proc. IFIP Networking*, May 2006.

[34] X. Cheng, H. Shu, and Q. Liang, "A Range-Differnece Based Self-Positioning Scheme for Underwater Acoustic Sensor Networks," *Proc. Int'l Conf. Wireless Algorithms, Systems, and Applications (WASA '07)*, vol. 1, pp. 38-43, Aug. 2007.

[35] "Aqua-Sim: Underwater Acoustic Network Simulator," http://uwsn.engr.uconn.edu/aquasim.tar.gz, 2011.

[36] M. Stojanovic, "On the Relationship between Capacity and Distance in an Underwater Acoustic Communication Channel," *Proc. First ACM Int'l Workshop Underwater Networks*, vol. 1, pp. 41-47, Sept. 2006.

[37] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks," *Proc. IEEE INFOCOM*, pp. 189-198, 2001.

Zhong Zhou received the BEng degree in telecommunication engineering in 2000 and the MEng degree in computer science in 2003, both from the Beijing University of Posts and Telecommunications, China. He received the PhD degree in computer science and engineering from the University of Connecticut (UCONN), Storrs, in August 2010. Now he is working as a software engineer at Amazon, Inc., Seattle, Washington. His research interests include underwater acoustic communication and networking, localization, cross layer design for wireless networks, distributed algorithms, and high performance parallel computation. He is a member of the IEEE.

Zheng Peng received the BS degree in both computer science and control science from Zhejiang University, China, in 2002. He then received the MS degree in computer science from the University of Electrical Science and Technology of China (UESTC) in 2005. He is currently working toward the PhD degree and working as a research assistant at the Underwater Sensor Network (UWSN) Lab, University of Connecticut. His main research interests include underwater acoustic networks, including protocol design, operating systems, underwater sensor nodes, and testbeds. He is a member of the IEEE.

Jun-Hong Cui received the BS degree in computer science from Jilin University, China, in 1995, the MS degree in computer engineering from the Chinese Academy of Sciences in 1998, and the PhD degree in computer science from the University of California, Los Angeles, in 2003. Currently, she is on the faculty of the Computer Science and Engineering Department at the University of Connecticut. Her research interests cover the design, modeling, and performance evaluation of networks and distributed systems. Recently, her research has mainly focused on exploiting the spatial properties in the modeling of network topology, network mobility, and group membership, scalable and efficient communication support in overlay and peer-to-peer networks, and algorithm and protocol design in underwater sensor networks. She is actively involved in the community as an organizer, TPC member, and reviewer for many conferences and journals. She has served as a guest editor for Elsevier's *Ad Hoc Networks* on two special issues (one on underwater networks and the other on wireless communication in challenged environments) and she now serves as an associate editor. She cofounded the first ACM International Workshop on UnderWater Networks (WUWNet '06), and she is now serving as the WUWNet steering committee chair. She received US National Science Foundation CAREER Award in 2007 and a ONR YIP Award in 2008. She is a member of the ACM, ACM SIGCOMM, ACM SIGMOBILE, the IEEE, the IEEE Computer Society, and the IEEE Communications Society. More information about her research can be found at http://www.cse.uconn.edu/~jcui.

Zaihan Jiang received the BE degree from Xi'an Jiaotong University, Xi'an, China, the MS degree from New Jersey Institute of Technology, Newark, in 1999, and the PhD degree from the New Jersey Institute of Technology in 2006, all in electrical engineering. He has spent nine years with the telecommunication industry in the US, before he joined the US Naval Research Laboratory, Washington DC, in 2008. His research interests include capacity analysis, cross layer design, system security, multimedia applications for wireless networks, and underwater acoustic networks/communications. He serves as a program committee member for several IEEE international conferences and has more than 20 publications in referred conferences and journals. He has been a senior member of the IEEE since 2009.