This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

Receiver-Initiated Spectrum Management for Underwater Cognitive Acoustic Network

Yu Luo, *Student Member, IEEE,* Lina Pu, *Student Member, IEEE,* Haining Mo, *Student Member, IEEE,* Yibo Zhu, *Student Member, IEEE,* Zheng Peng, *Member, IEEE,* and Jun-Hong Cui, *Member, IEEE,*

Abstract— Cognitive acoustic (CA) is emerging as a promising technique for environment-friendly and spectrum-efficient underwater communications. Due to the unique features of underwater acoustic networks (UANs), traditional spectrum management systems designed for cognitive radio (CR) need an overhaul to work efficiently in underwater environments. In this paper, we propose a receiver-initiated spectrum management (RISM) system for underwater cognitive acoustic networks (UCANs). RISM seeks to improve the performance of UCANs through a collaboration of physical layer and medium access control (MAC) layer. It aims to provide efficient spectrum utilization and data transmissions with a small collision probability for CA nodes, while avoiding harmful interference with both "natural acoustic systems", such as marine mammals, and "artificial acoustic systems", like sonars and other UCANs. In addition, to solve the unique challenge of deciding when receivers start to retrieve data from their neighbors, we propose to use a traffic predictor on each receiver to forecast the traffic loads on surrounding nodes. This allows each receiver to dynamically adjust its polling frequency according to the variation of a network traffic. Simulation results show that the performance of RISM with smart polling scheme outperforms the conventional sender-initiated approach in terms of throughput, hop-by-hop delay and energy efficiency.

Index Terms—Underwater cognitive acoustic networks (UCANs), spectrum management, receiver-initiated, traffic prediction

1 Introduction

Ocean environment, where multiple networks coexist, usually features high competition for acoustic spectrum among different users, such as sonars, acoustic nodes and marine animals. Meanwhile, available communication frequencies in oceans are quite limited, due to the severe frequency dependent attenuation and narrowband response of acoustic transducers [1]. Therefore, the spectrum is a scarce resource for underwater acoustic systems.

In recent years, underwater cognitive acoustic network (UCAN) is advocated as a promising technique to achieve the *environment-friendly* and *spectrum-efficient* transmission over acoustic channel [1]–[3]. Similar to cognitive radio (CR), nodes in UCAN are allowed to intelligently detect whether any portion of the acoustic spectrum is vacant, and correspondingly change their transmission frequencies, power or other operating parameters to temporarily use the idle frequency for communications without interfering other acoustic systems.

There have been many spectrum management systems proposed for cognitive radio (CR) networks [4]–[11], which commonly consist of three components, i.e., the spectrum sensing, the spectrum sharing and the spectrum decision. Spectrum management for CRs has been a well-explored research topic. However, existing systems may work inefficiently in a UCAN due to the unique features of acoustic communications, such as the long propagation delay, the narrow band response of transducer and the existence of marine animals.

In [12], the authors introduced a new delay induced hidden terminal problem in underwater multichannel networks, which is referred to as the long-delay hidden terminal. This unique problem is caused by the long propagation delay of acoustic signal in water and will result in data collisions if not well addressed. The studies in [1] demonstrated that the narrow bandwidth of acoustic transducers and the severe frequency-dependent attenuation of acoustic signals could limit the available bandwidth for UCAN to dozens of kilohertz or even less. The authors also analyzed the impact of marine animals as primary users on UCANs. Particularly, unlike a CR network, where signals are mainly created by man-made devices and have certain predictable patterns, UCANs may need to share acoustic channels with marine animals. There are limited understandings on the signal pattern of marine mammals, which poses grand challenges on the spectrum sensing and spectrum decision mechanisms.

In order to tackle the aforementioned challenges in UCANs, we propose a receiver-initiated spectrum management (RISM) system, which allows acoustic nodes to efficiently and friendly share the precious spectrum resource with both "natural acoustic systems" and "artificial acoustic systems". In RISM, the intended receiver first schedules a sensing pattern, i.e., which frequencies senders should work on, for its neighboring senders. Thereafter, by collecting local sensing results from its neighbors, a receiver will have a global picture of the spectrum usage. Finally, for high throughput and low delay purposes, the receiver assigns vacant frequencies and optimal transmission power for its surrounding senders based on the spectrum sensing results and the quality of acoustic links.

The main contributions of our work are summarized as follows:

- We propose a receiver-initiated spectrum management system, RISM, tailored to static UCANs. RISM can work in *non-synchronized* underwater networks without a dedicated control center (e.g., base station, access point or fusion center), which makes RISM broad applications in real sea.
- RISM is a "semi-centralized" system, where receivers run as "semi-center" collecting local sensing results

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

and performing channel allocation for surrounding senders for efficient and reliable data communications. Control packets, which are used for the negotiation among cognitive acoustic (CA) nodes to avoid collisions, could be easily shared by the collaborative spectrum sensing and resource (channel and power) allocation without generating extra traffic loads.

- To solve the when-to-poll problem in the receiver-initiated system, we propose a smart polling scheme. With the assistance of the traffic predictor, each receiver could estimate the traffic loads of its neighboring senders, thereby adapting its polling frequency to the variation of network traffic. This scheme further improves the network performance in terms of throughput and delivery delay.

The rest of paper is organized as follows. In Section 2, we introduce the related work of spectrum management in cognitive radios. In Section 3, we describe the proposed spectrum sharing scheme and discuss how it schedules the control messages. Section 4 presents the collaborative spectrum sensing scheme and briefly introduces a solution for marine mammal detection. After that, we discuss the joint channel and power allocation algorithm in Section 5. In order to assist a receiver to poll senders adaptively, we propose a traffic prediction based smart polling strategy in Section 6. We evaluate the performance of RISM in Section 7 and conclude the paper in Section 8.

2 Related Work

A spectrum management system generally consists of three key components, including the spectrum sensing, spectrum sharing and spectrum decision. The spectrum sensing aims to detect the presence of PUs in a reliable manner and to maximize the spectrum access opportunity of cognitive nodes. The spectrum sharing scheme is to handle the self-coexistence of cognitive nodes and eliminate the data collisions. The goal of a spectrum decision algorithm is to allocate the frequencies and transmission power properly to improve the resource utilization.

In the past decade, various spectrum management systems have been proposed for CR networks. The authors in [9] propose a protocol called cognitive medium access control (C-MAC) for distributed CR networks. Each channel in C-MAC is divided into recurring superframes, each of which consists of a slotted beaconing period (BP) and a data transfer period (DTP). A node transmits a beacon in the designated beacon slot for multi-channel rendezvous and collision avoidance, and sends the data in DTP.

In order to increase the spectrum opportunities while minimizing the average time taken to sense the spectrum, an advanced spectrum sensing scheme is proposed in [8]. In this method, each node has two different sensing patterns: the reactive sensing and the proactive sensing. Nodes in the proactive sensing mode detect different channels periodically, and the idle channels will be added to the idle channel list for switching. Once the current channel is occupied by PUs and there is out of an available channel in the idle channel list, the reactive sensing mode will be triggered for an opportunistic spectrum access. The authors in [10] explore the method to optimize the spectrum sensing time based on the traffic rate of the cognitive network.

The cooperative spectrum sensing is highly advocated to further improve the sensing accuracy and efficiency. In [7], a dynamic frequency hopping communication (DFHC) scheme is proposed for centralized spectrum management. In this scheme, each CR node is supposed to work on two separate sub-channels simultaneously. During the data transmission on one frequency band, the node could sense the other intended channel in parallel. All cognitive nodes in DFHC need to upload the sensing results to a leader, while the leader will calculate the hopping patterns for the cognitive nodes. The authors in [11] propose a multi-channel MAC for CR (MMAC-CR). MMAC-CR consists of two phases: an ad hoc traffic indication message (ATIM) phase and a data phase. The ATIM phase is designed for synchronization, fast scan spectrum sensing and channel competition for the following data phase, i.e., channel rendezvous. Cognitive nodes exchange scan results to perform cooperative sensing. In the data phase, nodes that are assigned the same channel compete for the channel access using traditional handshake approach similar to IEEE 802.11.

In most of the aforementioned approaches, senders initiate the negotiation for spectrum competition. The spectrum sensing, spectrum sharing, and spectrum decision are designed and optimized separately, and use individual control packets on each component. The overhead issue for the whole system, however, is usually overlooked, especially when taking into account the long propagation delay, the long preamble and the bandwidth constraints in acoustic communications [13]. This promotes us to design a new receiver-initiated spectrum management system, RISM, for UCANs.

Compared with sender-initiated systems, we could obtain several benefits by starting a handshake process at the receiver. Specifically, each receiver in RISM could run as a "semi-center" collecting local sensing information from neighboring nodes for a reliable primary user (PU) detection, and performing joint channel and power allocation for surrounding senders. Therefore, the spectrum sensing mechanism, the spectrum sharing scheme, and the spectrum decision algorithm could be considered as a whole in the new system. In addition, the control packets are shared amongst different components, thereby significantly reducing the overhead traffic and improving the channel utilization.

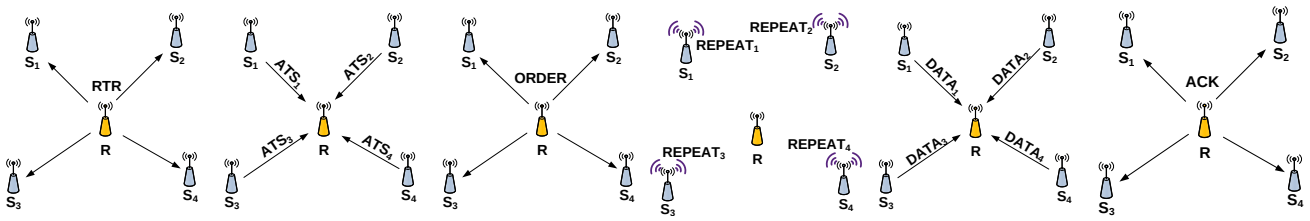
3 Receiver-Initiated Spectrum Sharing

We first introduce the receiver-initiated spectrum sharing (RISS) scheme, the backbone of RISM. In RISS, a handshake process is initiated at the receiver side to negotiate the vacant spectrum sharing.

3.1 Description of RISS

In RISS, a node needs to know the propagation delay to its neighbors, which could be measured at the initialization stage of a network through the classic two-way handshake approach that has been tested in the sea experiment [14]. Furthermore, we assume that there is a common control channel (CCC), which is physically separated from the in-band data channel, as it is widely

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing



(a) Phase 1 (b) Phase 2 (c) Phase 3 (d) Phase 4 (e) Phase 5 (f) Phase 6

Fig. 1. Six phases of the RISS scheme. R is a receiver and s_i is sender i.

accepted in cognitive networks [15]. RISS involves six phases, as shown in Fig. 1.

Phase 1: The receiver, which intends to collect data, sends out a request-to-receive (RTR) packet to its neighbors to start a handshake process. Here, the RTR message has three functions: (a) to help the receiver request data from neighborhoods, (b) to arrange the transmissions of available-to-send (ATS) packets from neighboring senders, and (c) to schedule the spectrum sensing pattern for senders.

Phase 2: The invited senders, which have data for the receiver, will first sense the frequency as arranged and respond with ATS messages to establish connections with the receiver. In order to avoid collisions among ATS packets, senders transmit ATS following the schedule ordered in RTR. Here, the ATS has four functions: (a) to establish a connection with the receiver, (b) to inform the receiver of its number of data packets to send, (c) to notify the receiver about the spectrum usage of other cognitive users for collision avoidance, which will be introduced in Section 5.2, and (d) to upload the local sensing result for a collaborative PU detection.

Phase 3: After ATS packet reception, the receiver aggregates local sensing outcomes from its neighboring senders for the final spectrum decision. Then it broadcasts an ORDER packet, which includes information about the frequency allocation and the transmission power assignment for its neighbors.

Phase 4: If a sender successfully receives the ORDER message, it extracts its own schedule information and broadcasts this information through a REPEAT packet to its neighborhood. This process is to avoid a data collision with other receivers, which will be discussed in Section 5.2.

Phase 5: After the transmission of REPEAT, each sender sends out its DATA packet at the scheduled time according to the ORDER message it received in Phase 3.

Phase 6: Finally, for the purpose of reliable transmission, the receiver replies a common acknowledgment (ACK) to all senders after the data reception.

Here, RTR, ATS, ORDER, REPEAT and ACK are all control packets and thus, share the CCC. From the above description, it is easy to obtain that though there are six phases in RISS, each round of negotiation allows multiple senders to reserve the channel. In addition, the receiver in RISS can effectively work as a fusion center to schedule the sensing pattern and to collect local sensing results for collaborative PU detections, and it could also play a role as the control center to arrange the data transmission of its surrounding senders. This is why we call RISM as a "semi-centralized" system.

3.2 Scheduling of ATS Transmission

As introduced in Phase 1 in Fig. 1, each receiver in RISS needs to schedule the transmission of ATS messages for its neighbors. In order to avoid ATS collisions, the arriving time of ATS packets from different senders to a common receiver should be staggered. Moreover, a minimal ATS reception time is preferred to reduce the overhead on handshakes and to improve the utilization of control channel.

Now, let r, s_i and τ_i denote the receiver, sender i , and the propagation delay between r and s_i, respectively. $\mathcal{S}_L$ is the set of s_i, where L is the size of $\mathcal{S}_L$. Assume at t_0, a receiver transmits an RTR packet, and then receives $\texttt{ATS}_\texttt{i}$ from s_i at the receiver's local time t_i, where $i \in \{1, \ldots, L\}$. The transmission time of RTR and ATS are denoted by $T_{\texttt{RTR}}$ and $T_{\texttt{ATS}}$, respectively. Let w_i represent the time difference between the RTR reception and the $\texttt{ATS}_\texttt{i}$ transmission on s_i.

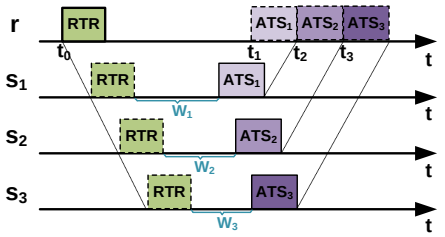


Fig. 2. The schedule of ATS transmissions. A square with solid line and dash line represent a transmission and a reception, respectively.

The receiver needs to calculate w_i for each sender to minimize the total time for all ATS receptions while avoiding collisions among ATS messages. In Fig. 2, we use three senders as an example to show the optimal scheduling for ATS transmissions.

According to the propagation delay between the receiver and the sender i, we have

$$w_i = t_i - (t_0 + T_{\texttt{RTR}}) - 2\tau_i, \quad i \in \{1, \ldots, L\}. \tag{1}$$

Then, we sort senders in $\mathcal{S}_L$ by the propagation delay in an ascending order. Let T_m be the local time that the last ATS packet arriving at the receiver, i.e., $T_m = \max\{t_i\}$, $i \in \{1, \ldots, L\}$. Now, optimizing the total reception time of all ATS packets is equivalent to minimizing T_m.

Suppose $\texttt{ATS}_i$ is the l^{th} of L ATS packets arriving at the receiver, then (1) can be written as:

$$w_i = T_m - (L-l)T_{\texttt{ATS}} - (t_0 + T_{\texttt{RTR}}) - 2\tau_i, \quad i \in \{1, \ldots, L\}. \tag{2}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

Note that $w_i \geq 0$, so from (2) we have

$$T_m \geq (L-l)\,T_{\mathtt{ATS}} + (t_0 + T_{\mathtt{RTR}}) + 2\tau_i, \quad i \in \{1, \ldots, L\}. \quad (3)$$

From (3) we observe that in order to minimize T_m, the order of ATS reception should follow the order of senders in $\mathcal{S}_L$. More specifically, an ATS packet from a sender with a larger propagation delay (larger τ) should come after the one from a closer sender (smaller τ), and vice versa. Therefore, let $l = i$ and we have the minimum T_m is

$$T_m = \arg\max_i \left\{(L-i)T_{\mathtt{ATS}} + (t_0 + T_{\mathtt{RTR}}) + 2\tau_i\right\}, \; i \in \{1, \ldots, L\}. \quad (4)$$

Finally, the receiver calculates t_i by $t_i = T_m - (L - i)T_{\mathtt{ATS}}$ for each sender, and then attaches the scheduled time $[t_0, t_1, \ldots, t_L]$ and the MAC addresses of corresponding senders on an RTR packet. After the RTR reception, sender s_i delays for $w_i = t_i - t_0 - 2\tau_i$ before replying its ATS.

From above descriptions we have that at the sender side, the wait time, w_i, before an ATS transmission depends on the time difference, $t_i - t_0$. Therefore, the knowledge of receiver's local time is not necessary at the sender side, which allows the above ATS scheduling to operate in a non-synchronized UCAN.

4 Spectrum Sensing

Generally speaking, a CA node may not be able to sense all frequencies in one sensing period, since a full-band spectrum sensing is not only energy and time inefficient, but also hardware demanding, which make it impractical for battery-powered underwater equipment. Thereafter, we consider the scenario that each CA node could only sense one or several subset frequency bands in one sensing period. In this section, we elaborate how nodes in a UCAN detect signals from PUs and perform collaborative spectrum sensing in an RISM system.

4.1 Spectrum Usage Realization

In an asynchronous UCAN, when a node is sensing the spectrum, other senders may be transmitting on the same channel, which will interfere with the sensing process. Nodes in RISM are thus required to distinguish signals of PUs from that of CA nodes. Here, we advocate *cyclostationary* based spectrum sensing approaches to achieve this goal.

Different man-made communication signals naturally have cyclostationary features at different cyclic frequencies [16]. By recognizing the cyclostationary pattern during spectrum sensing, CA nodes can distinguish between received signals from different systems.

However, one objective of UCANs is to share the acoustic spectrum with marine animals in an environment-friendly manner. Hence, PU in oceans may involve not only "artificial acoustic systems", like UANs and sonars, but also "natural acoustic systems", such as whales and dolphins. One important question coming up is that whether signals from "natural acoustic systems" and from CA nodes can be told apart by applying a cyclostationary based sensing technique. Here, we compare the cyclostationary based time-smoothed cyclic cross periodogram [17], $S_x^{\alpha}[k]$[1], of different acoustic signals in Fig. 3. The results may help us to answer this question.

Fig. 3 includes 4FSK signal, which is common in man-made communication systems, and voice signals from two different marine mammals. This figure illustrates that both 4FSK and voice signals from marine mammals exhibit cyclostationary feature on multiple cyclic frequencies. Moreover, it is easy to observe the obvious differences among cyclostationary patterns of different signals, which could be used in RISM for local spectrum sensing.

4.2 Collaborative Spectrum Sensing

After local sensing, senders transmit their sensed outcomes to the receiver via ATS packets. The receiver fuses these local results together for a collaborative PU detection. An example of a cyclostationary based collaborative spectrum sensing, which is well supported by the RISM system, could be found in [18].

In order to maximize the channel access opportunities while preventing intrusions to PUs, it is important to design an efficient sensing pattern [4]. Owing to the "semi-centralized" feature of an RISM system, the receiver could choose different sensing patterns to achieve varied goals:

(a) **Maximize the spectrum access opportunity.** In order to maximize the probability to discover vacant spectrum, senders in this pattern are assigned to detect different frequency bands for increasing the sensing coverage.

(b) **Maximize the sensing accuracy.** To overcome the fading and shadowing effects, a collaborative sensing strategy could be exploited for a reliable PU detection. Multiple senders, in this case, are arranged to sense the same frequencies.

(c) **Hybrid spectrum sensing.** This sensing pattern is a trade-off between maximizing the channel access opportunity and maximizing the sensing accuracy. Neighboring senders in this pattern are divided into multiple groups. Different groups take charge of different subsets of frequencies while senders in the same group sense the same frequency.

Depending on varied requirements, RISM system could easily support all above sensing patterns by leveraging its "semi-centralized" structure. More specifically, as a control center, the receiver schedules the sensing pattern for its neighbors through an RTR message, and then collects and fuses local sensing results from neighboring nodes through ATS packets.

5 Spectrum Decision

When assigning vacant frequencies for communication, receivers need to schedule the data transmission for its intended senders carefully to avoid potential collisions. In addition, CA nodes in an underwater environment usually experience severe frequency selective fading. However, a frequency which appears in deep fading to a node may be of good quality for other nodes. Therefore, a

1. $S_x^{\alpha}[k] = \frac{1}{D}\sum_{d=0}^{D-1} X_l[k]X_l^*[k-\alpha]W[k]$, where α, D, M, $W[k]$ and $X_l[k]$ are the cyclic frequency, the number of windows, the number of samples, the smoothing spectrum window and the Fourier transform of the sensed signal $x[n]$, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

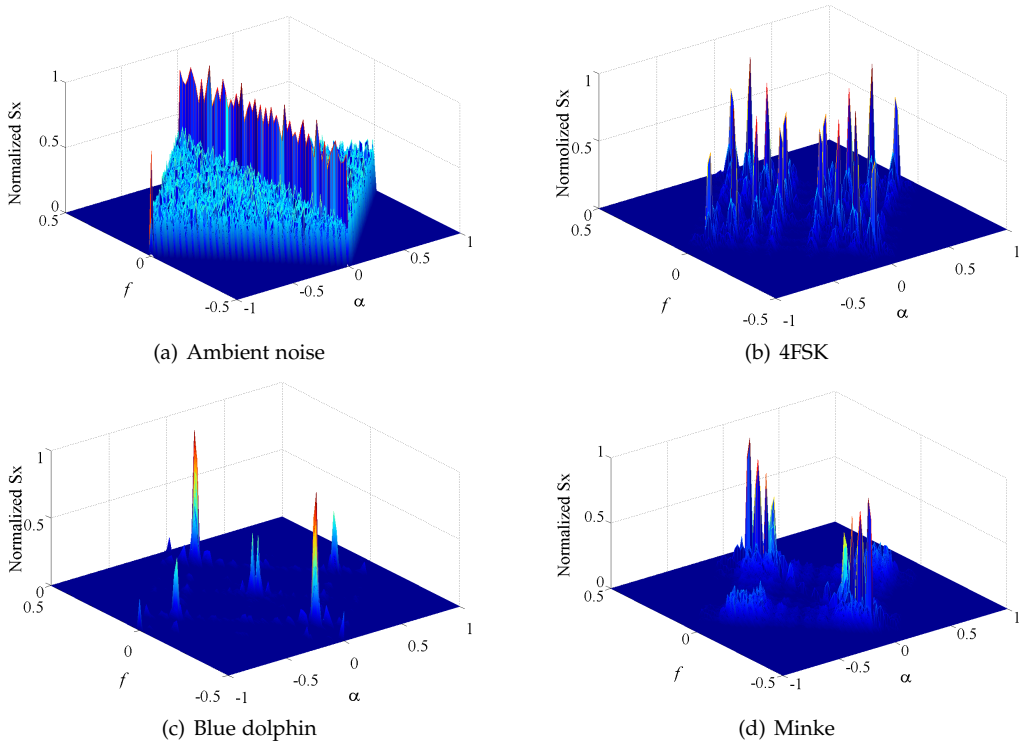


Fig. 3. **Normalized cyclic cross periodogram of different acoustic signals in oceans with Hamming window.** This figure demonstrates that the ambient noise does not exhibit the cyclostationary feature, since its cyclic cross periodogram has no peaks if $\alpha \neq 0$. By contrast, $S_x^{\alpha}[k]$ of 4FSK, blue dolphin and minke signals show additional peaks at different α, where $\alpha \neq 0$. A node could use the position of these peaks to identify sensed signals.

receiver should dynamically allocate the frequency and power to senders based on their channel situations. In this section, we introduce how to take advantage of a dedicated collision avoidance mechanism with a dynamic spectrum decision algorithm to achieve the aforementioned goals.

5.1 Channel Model

In this paper, we use the following multi-channel model. Each CA node has a single half-duplex acoustic transducer, which can switch to different channels for either transmission or reception. Let N and K be the number of available data channels and the amount of surrounding senders, which have replied `ATS` for data transmission, respectively. Let t represent a local time of the receiver.

The frequency allocation matrix at time t is represented as $\mathbf{A}^t$, where each entry $a_{nk}^t \in \{0, 1\}$. If channel n is assigned to node k, then we set $a_{nk}^t = 1$, otherwise $a_{nk}^t = 0$. The transmission power and data transmission rate of sender k on channel n is denoted by p_{nk}^t and R_{nk}^t, respectively, which will be jointly allocated by the receiver. Let P_k denote the total transmission power assigned to sender k.

In a communication system, if the instantaneous channel state information (CSI) is available, the receiver could schedule sender k to transmit at the maximum rate $R_{nk}^t = C_{nk}^t$. Here, C_{nk}^t is the channel capacity of node k on channel n at time t, which is expressed as

$$C_{nk}^t = a_{nk}^t B_n \log_2(1 + \frac{p_{nk}^t |h_{nk}^t|^2}{N_0 B_n a_{nk}^t}), \quad (5)$$

where h_{nk}^t is the instantaneous channel gain between sender k and the receiver on channel n, B_n is the bandwidth of channel n, and N_0 is the noise spectral density.

However, the real-time channel gain is usually unavailable in UCANs due to the long propagation delay and the high dynamic of underwater channel. Therefore, we use the *outage probability*[2], which only requires the statistical knowledge of h_{nk}^t, to calculate the channel capacity. The statistical information of h_{nk}^t is stable and easy to get in underwater communications [1].

Depending on the quality of service (QoS) requirement, the packet loss ratio between sender k and the receiver on channel n should be equal to or less than a predetermined outage probability β_{nk}. That is

$$Pr[R_{nk}^t > C_{nk}^t] = \beta_{nk}. \quad (6)$$

Now, let $f(|h_{nk}^t|^2)$ represent the probability distribution function (PDF) of $|h_{nk}^t|^2$. Depending on the real deployment of a network and the certain ocean environment, the channel gain could be modeled as Rayleigh distribution [20], K-distribution [21] or Rice distribution [22]. RISM system is generic to arbitrary channel models. In this paper, we use Rayleigh distribution as an example to show how the proposed spectrum decision mechanism works.

Assume that $|h_{nk}^t|$ follows Rayleigh distribution, then $f(|h_{nk}^t|^2)$ is an exponential distribution with mean value

2. If one packet is transmitted with spectral efficiency $S(\rho)$ (in bit/sec/Hz) and SNR ρ, the probability that this packet will be correctly decoded is $1-\beta$, i.e., $P_e(\rho, S(\rho)) = \beta$. Then $P_e(\rho, S(\rho))$ is called as the outage probability [19].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

λ_{nk}. Substituting (5) into (6) and using the exponential expression of $f(|h_{nk}^t|^2)$, we have

$$R_{nk}^t = a_{nk}^t B_n \log_2\left[1 + \frac{p_{nk}^t \lambda_{nk} \ln(\frac{1}{1-\beta_{nk}})}{N_0 B_n a_{nk}}\right], \quad (7)$$

which will be used for the spectrum decision in Section 5.3.

5.2 Collision Avoidance in Channel Allocation

In UCANs, multiple CA nodes may need to share a limited number of vacant frequency bands. When assigning frequencies to neighboring senders, a receiver in RISM should take into account the sending and receiving activities of the nodes within its two hops to mitigate the *hidden terminal problem*.

Here, we introduce a collision avoidance matrix, which will be later used in the spectrum decision algorithm. We define $\mathbf{C}^t$ to be the collision avoidance matrix, with each entry $c_{nk}^t \in \{0,1\}$. Here, $c_{nk}^t = 1$ indicates that the channel n is not available for sender k at the receiver's local time t. If any DATA packet of sender k is arranged to transmit on this channel at time t, collisions would happen. Let τ_k be the propagation delay between the sender and the receiver k. Next, we present how to calculate the value of each entry in $\mathbf{C}^t$ based on three constraints.

Packet forwarding constraint: As described in Section 3.1, a sender needs to broadcast a REPEAT packet before its DATA transmission. Therefore, when scheduling the sending time of DATA packet for a transmitter, the receiver should leave enough time to this sender for its REPEAT packet transmission. Let t_0 be the local time when a receiver sends out an ORDER packet. Denote the transmission time of ORDER packet and REPEAT packet by T_{ORDER} and T_{REPEAT}, respectively. Then, we have $c_{nk}^t = 1$, if $t < t_0 + 2\tau_k + T_{\text{ORDER}} + T_{\text{REPEAT}}$.

Transmission constraint: When a receiver schedules a DATA transmission for an intended sender, it should prevent intrusions to other receivers' receptions. As shown in Fig. 4, receiver r_2 is planing to collect DATA from sender s, where s is in the communication range of both r_1 and r_2, but r_1 and r_2 cannot hear from each other. The receiver r_1 reserved channel n from its local time t_a to t_b to receive data from neighboring senders (not s). In order to avoid interference to r_1, s cannot transmit on channel n from its local time t_a' to t_b'. Here, we call this as *transmission constraint*.

Senders in RISM need to inform the receiver of their transmission constraints by following the steps below. Upon overheard ORDER_1 from r_1, s calculates t_a' and t_b' based on $t_a' = t_0' + t_a - t_0 - 2\tau_1$ and $t_b' = t_a' + t_b - t_a$, where t_0, t_a, t_b are included in ORDER_1, t_0' is the local time when s overhears the ORDER_1, and τ_1 is the propagation delay between s and r_1. The time stamps t_1', t_a' and t_b' will be sent to r_2 through ATS so that r_2 could update the collision avoidance matrix accordingly. More precisely, we set $c_{ns}^t = 1$ on receiver r_2 if $t \in [t_a'', t_b'']$, where $t_a'' = t_1'' + t_a' - t_1'$ and $t_b'' = t_a'' + t_b' - t_a'$. Here t_1'' is the local time when r_2 receives the ATS packet.

Reception constraint: When arranging a data reception, the receiver should guarantee that it will not be interfered by neighboring senders' transmissions. As shown in Fig. 5, receiver r_1 has scheduled to receive data from sender s on channel n from its local time t_a to t_b and announced the arrangement through ORDER_1. When the sender receives ORDER_1, it transfers the time stamps (t_0, t_a and t_b) to its local time (t_1', t_a' and t_b') and attaches the information in REPEAT_1. Since r_2 is also in the communication range of s, r_2 cannot use channel n to receive DATA from its local time t_a'' to t_b'' for collision avoidance purposes. We call it as *reception constraint*. The entry in the collision avoidance matrix of r_2 is thereby set $c_{ns}^t = 1$, if $t \in [t_a'', t_b'']$, where $t_a'' = t_1'' + t_a' - t_1'$ and $t_b'' = t_a'' + t_b' - t_a'$. Here t_1'' is the local time when r_2 overhears REPEAT_1.

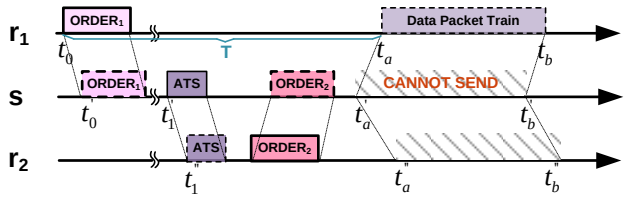


Fig. 4. When receiver r_2 schedules a data transmission for its intended sender s, it should prevent intrusions to any other receivers, such as r_1.

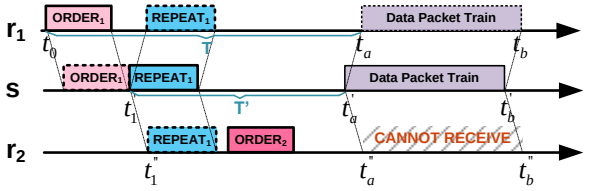


Fig. 5. When receiver r_2 schedules a data reception, it should guarantee that there is no interference from any other surrounding senders, like s.

From the above descriptions, we observe that the calculation of entries in $\mathbf{C}^t$ is completely based on the local time of the receiver, which does not necessitate the absolute time of neighboring users. Therefore, it can work in a non-synchronized UCAN.

5.3 Joint Channel and Power Assignment

Recall that from the collaborative spectrum sensing, each receiver obtains the identifications (IDs) of vacant channels. In this section, we introduce how to jointly allocate channel and power to maximize the spectrum utilization.

In RISM, senders inform the receiver, through ATS messages, of how many data packets to be sent out. Different nodes may have varied sending requests in each transmission. Now, let Q and T_r denote the total bits of data a receiver will receive and the time spent on receiving these data, respectively. Then, we have

$$Q = \int_0^{T_r} \sum_{n=1}^{N} \sum_{k=1}^{K} R_{nk}^t \, dt, \quad (8)$$

where R_{nk}^t is the assigned data rate to sender k on channel n at time t.

We aim to minimize T_r in (8), which is equivalent to maximizing $\sum_{n=1}^{N}\sum_{k=1}^{K} R_{nk}^t$ as Q is fixed. Therefore, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

formulate the joint power and frequency allocation as the following optimization problem:

$$\textbf{Prob.1} \qquad \arg \max_{\substack{p_{nk}^t>0 \\ a_{nk}^t\in\{0,1\}}} \sum_{n=1}^{N}\sum_{k=1}^{K} R_{nk}^t,$$

$$\textbf{where} \quad R_{nk}^t = a_{nk}^t B_n \log_2\left[1+\frac{p_{nk}^t \lambda_{nk} \ln(\frac{1}{1-\beta_{nk}})}{N_0 B_n a_{nk}^t}\right].$$

s.t.

$$\textbf{C1:}\ \sum_{k=1}^{K} a_{nk}^t = 1, \quad n\in\{1,\ldots,N\},$$

$$\textbf{C2:}\ \sum_{n=1}^{N} p_{nk}^t \le P_k, \quad k\in\{1,\ldots,K\},$$

$$\textbf{C3:}\ a_{nk}^t = 0,\ \text{if}\ c_{nk}^t = 1,\ n\in\{1,\ldots,N\}, k\in\{1,\ldots,K\}. \tag{9}$$

In Prob.1, C1 is the channel allocation constraint which ensures that each channel is assigned to no more than one CA sender; C2 is the power constraint to guarantee that the overall transmission power of each sender does not exceed the maximum power supply, and C3 is the collision avoidance constraint.

As introduced in Section 5.2, C3 is the combination of three constraints, namely, packet forwarding constraint, transmission constraint and reception constraint. It is a unique constraint condition in Prob.1, which integrates the collision avoidance mechanism on the MAC layer with the power and channel assignment on the physical layer closely.

If without the constraint condition, C3, we can use a similar approach proposed in [23] and [24] to solve Prob.1. We first relax the requirement $a_{nk}^t \in \{0,1\}$ to allow a_{nk}^t to be a real number within the interval $[0, 1]$. Then, the objective function of the problem follows the form of $f(x,y) = \mu x \log_2(1+\nu y/x)$, where μ and ν are constants. It is easy to prove that the Hessian matrix of this function is negative semidefinite for all x and y. Therefore, the objective function is concave. Finally, Prob.1 could be solved through the following classic method of Lagrangian multipliers:

$$\begin{aligned}\mathcal{L}(a_{nk}^t, p_{nk}^t) = &\sum_{n=1}^{N}\sum_{k=1}^{K} B_n a_{nk}^t \log_2\left[1+\frac{p_{nk}^t \lambda_{nk} \ln(\frac{1}{1-\beta_{nk}})}{N_0 B_n a_{nk}^t}\right] \\ &- \phi_k\left(\sum_{n=1}^{N} p_{nk} - P_k\right) - \varphi_n\left(\sum_{k=1}^{K} a_{nk}^t - 1\right),\end{aligned} \tag{10}$$

where ϕ_k and φ_n are the Lagrangian multipliers of the constraints C1 and C2, respectively.

Calculating the partial derivatives of $\mathcal{L}(a_{nk}^t, p_{nk}^t)$ with respect to a_{nk}^t and p_{nk}^t, we have

$$\frac{\partial \mathcal{L}}{\partial p_{nk}^t} = \frac{a_{nk}^t B_n \theta_{nk}}{\ln(2)(a_{nk}^t B_n + \theta_{nk} p_{nk}^t)} - \phi_k, \tag{11}$$

and

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial a_{nk}^t} = &B_n \log_2(1+\frac{p_{nk}^t \theta_{nk}}{a_{nk}^t B_n}) \\ &- \frac{p_{nk}^t B_n \theta_{nk}}{\ln(2)(a_{nk}^t B_n + \theta_{nk} p_{nk}^t)} - \varphi_n,\end{aligned} \tag{12}$$

where

$$\theta_{nk} = \frac{\lambda_{nk} \ln(\frac{1}{1-\beta_{nk}})}{N_0}. \tag{13}$$

Now, let (11) and (12) equal to zero, we have

$$p_{nk}^t = \frac{a_{nk}^t B_n}{\phi_k \ln(2)} - \frac{a_{nk}^t B_n}{\theta_{nk}}, \tag{14}$$

and

$$\ln\left(\frac{1}{1+X_{nk}}\right) - \frac{1}{1+X_{nk}} = -\left[\frac{\ln(2)\varphi_n}{B_n}+1\right], \tag{15}$$

where

$$X_{nk} = \frac{p_{nk}^t \theta_{nk}}{B_n a_{nk}^t}. \tag{16}$$

We note that the subscripts of variables at the left side of (15) include n and k, but those at the right side of the equation only contain n. This implies that X_{nk} is independent with k and thus, we have

$$X_{n1} = X_{n2} = \cdots = X_{nK}. \tag{17}$$

According to (16) and (17), we conclude that a_{nk}^t should be proportional to $p_{nk}^t \theta_{nk}/B_n$.

Now, we rethink the original constraint C2 of requiring a_{nk}^t to be a binary value. With this constraint, channel n would be allocated to the sender k', which has the largest a_{nk}^t. Therefore, the optimal values of a_{nk}^t and p_{nk}^t are

$$\hat{a}_{nk'}^t = 1, \quad \hat{a}_{nk}^t = 0 \quad \text{for all} \quad k \ne k', \tag{18}$$

where

$$k' = \arg\max_k \frac{p_{nk}^t \theta_{nk}}{B_n}, \quad k\in\{1,\ldots,K\}, \tag{19}$$

and

$$\hat{p}_{nk}^t = \begin{cases} 0, & \phi_k \ge \dfrac{\theta_{nk}}{\ln(2)} \ \text{ or } \ a_{nk}^t \ne 1 \ \text{ or } \ c_{nk}^t = 1, \\ \min\left\{\dfrac{B_n}{\phi_k \ln(2)} - \dfrac{B_n}{\theta_{nk}},\ P_k\right\}, & \text{otherwise.}\end{cases} \tag{20}$$

Let vector $\mathbf{A}_k^t$ be the channel assignment to sender k at time t, which is a collection of row indexes with nonzero elements in the k^{th} column of $\hat{\mathbf{A}}^t$. Substituting (20) into constraint C2 of Prob.1, we have

$$\phi_k = \frac{\sum_{n\in\mathbf{A}_k^t} B_n}{\ln(2)\left[P_k + \sum_{n\in\mathbf{A}_k^t} \dfrac{B_n}{\theta_{nk}}\right]}. \tag{21}$$

Finally, we get the optimal transmission power, $\hat{p}_{nk}^t$, by substituting (21) into (20).

Next, we take into account the constraint condition C3 and use an iterative algorithm to compute $\hat{p}_{nk}^t$ and $\hat{a}_{nk}^t$ in RISM.

Algorithm 1

Initialization: Based on the infromation of spectrum usage collected from `ATS` and `REPEAT` packets, the receiver generates its collision avoidance matrix $\mathbf{C}^t$.

Iterative Calculations:
do
for $n = 1$ to N
Step 1: Let $a_{nk}^t = 1$ for each sender, k, in turns if $c_{nk}^t \ne 1$, and calculate ϕ_k according to (21). In this step, previous channel allocation $a_{n'k}$, $n' \ne n$ remains

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

unchanged.
Step 2: Use (20) to calculate p_{nk}^t.
Step 3: Pick out the best sender, k', based on (19), and set $a_{nk'}^t = 1$.
end for
Step 4: Calculate $\sum_{n=1}^{N}\sum_{k=1}^{K} R_{nk}^t$ in Prob.1.
while The increment of $\sum_{n=1}^{N}\sum_{k=1}^{K} R_{nk}^t$ is larger than a predetermined threshold.

After running Algorithm 1, the receiver attaches the sending time of data packets, the channel assignment, transmission power and data rate into an `ORDER`, and delivers the packet to intended senders.

5.4 An Insight into Collision Avoidance Matrix

The matrix $\mathbf{C}^t$ in the RISM system is a critical "bridge" to connect the joint channel and power allocation algorithm on the physical layer with the collision avoidance mechanism on the MAC layer. In each round of communication, a receiver needs to update its $\mathbf{C}^t$ according to the occurrence of certain events. Next, we introduce how a specific event triggers a change of collision avoidance matrix by affecting the status of channel usage.

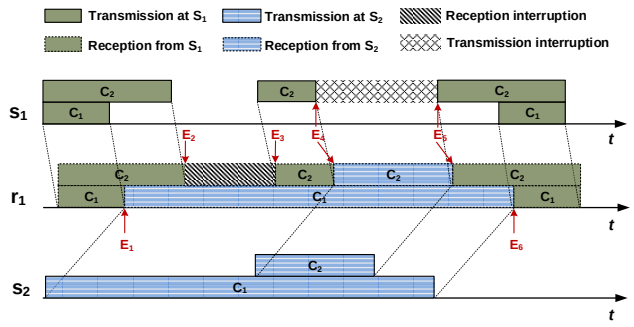


Fig. 6. **The events that affect the channel status**. $\mathbf{E_1}$: s_2 joins the channel allocation; $\mathbf{E_2}$: reception at r_1 is interrupted on c_2; $\mathbf{E_3}$: reception interruption ends on c_2; $\mathbf{E_4}$: transmission at s_1 is interrupted on c_2; $\mathbf{E_5}$: transmission interruption ends on c_2; $\mathbf{E_6}$: transmission completes and c_1 is released.

In RISM, there are a total number of six specific events that may change the channel status (idle or busy). Here, we use Fig. 6 as an example to introduce these events. In the figure, we have two senders, s_1 and s_2, and a common receiver, r_1. The propagation delay from s_1 to r_1 is shorter than that from s_2. Assume r_1 arranges s_1 and s_2 to send the data simultaneously. We further assume that s_1 has a good channel quality on c_2 whereas s_2 prefers channel c_1. The six events are introduced as follows.

(a) **Sender join ($\mathbf{E_1}$):** The event that a new sender joins the channel allocation. Particularly, in a network, the distances from senders to a common receiver vary. Hence, senders in RISM usually join a channel allocation one by one. Each time when a sender joins, the receiver redoes channel and power optimization.
(b) **Reception interruption ($\mathbf{E_2}$):** The event that a receiving process is interrupted on a specific channel. According to the reception constraint introduced in Section 5.2, a receiver may suspend its data reception on a certain channel for a while to avoid the interference from other transmitters.
(c) **Reception interruption release ($\mathbf{E_3}$):** The event that the interruption ends and a receiver can resume its data reception.
(d) **Transmission interruption ($\mathbf{E_4}$):** The event that a sender suspends its data transmission on a specific channel to prevent intrusions to the reception of neighboring receivers.
(e) **Transmission interruption release ($\mathbf{E_5}$):** The event that the potential interference to surrounding receivers ends and the sender can resume its data transmission on a given channel.
(d) **Sender leave ($\mathbf{E_6}$):** The event that a sender completes its data transmission and releases the channel. More specifically, in each round of communication, senders usually finish their data transmissions at a varied time, since both the amount of data packets and the transmission rates are different. Therefore, when a node finishes its data transmission, the channel will be available to other senders.

It is worth noting that a receiver in RISM knows the occurrence of above events before they happen, as the channel usage is up to date by overhearing `ORDER` and `REPEAT` messages and by calculating the constraints described in Section5.2. Once an event happens, the status of the channel will change. In this situation, the receiver flips the corresponding elements in its collision avoidance matrix $\mathbf{C}^t$ between 1 and 0. Thereafter it needs to run Algorithm 1 for a new round of channel and power allocation. Finally, as shown in Fig. 6, once the packet reception starts, the data streams from different senders will arrive at the receiver seamlessly. This allows CA nodes to fully utilize the channel resource for communications.

6 Adaptive Polling in RISM

In receiver-initiated approaches, there is a unique challenge that receivers need to decide when to poll the neighboring senders blindly. It becomes a big problem in a distributed network, since receivers usually lack in the current status e.g., having packets to send or not, of its intended senders. In this section, we discuss how to handle this problem in RISM with a traffic predictor.

6.1 Why Polling Senders Adaptively

Adjusting the polling frequency of a receiver will cause a trade-off between the queuing delay and the energy efficiency. Initiating handshake over frequently results in resources (spectrum, energy and time) waste for transmitting control messages, whereas slowing down the polling rate occasionally leads to larger queuing delay such that the data cannot be delivered timely. In a receiver-initiated protocol, a receiver should adjust its polling frequency, i.e., the time interval between successive `RTR` requests, to match the traffic loads on its intended senders.

However, it is a challenge for a receiver to poll surrounding senders adaptively due to the following reasons. First of all, the receiver has no prior knowledge regarding the number of data packets cumulated on the intended senders. Intuitively, a sender could inform the receiver this information by sending additional messages. However, considering the long preamble in acoustic modems and the energy constraint in UCANs, such a strategy would generate extra overhead traffic and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

reduce the lifetime of the network. Therefore, senders in RISM have to passively wait for the receiver to initiate the handshake process. Secondly, the traffic load is generally heterogeneous in a network. It becomes more difficult to decide the optimal polling frequencies for different senders. Finally, the traffic load of a node may change with the time. For example, in a target tracking network, nodes generate bursty traffic to report their observations whenever the presence of a target is detected while keeping quiet for the rest of time to save the energy.

In order to tackle the above challenges, each receiver in RISM system needs to decide its data polling frequency independently to adapt to the heterogenous and varying traffic of a network. A smart polling strategy is thus required to improve the network performance.

6.2 Smart Polling Scheme

In RISM, we adopt the traffic prediction to implement the smart polling scheme. By using the traffic predictor, a receiver could estimate the current traffic loads of its intended senders through historical traffic measurements. Whenever the total traffic of neighboring senders exceeds a threshold, the receiver sends out an RTR packet to request for data.

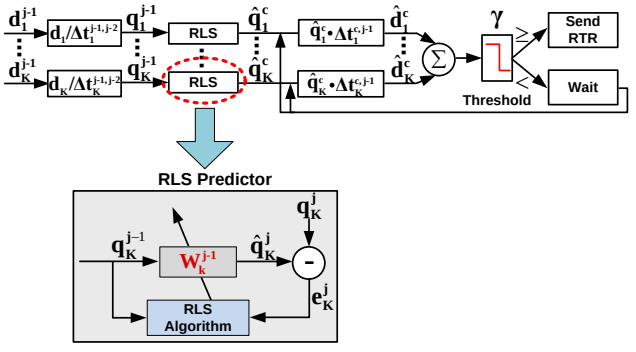


Fig. 7. An RLS based adaptive polling scheme for RISM system.

In the literature, a number of algorithms, such as the adaptive filter [25] and artificial neural network [26] have been proposed for predicting the network traffic. In RISM, we could choose an appropriate method depending on the demand of an application. For instance, in a target detection network, the traffic load of a sensor node may change quickly with the entering and leaving of a target. In this scenario, a receiver could employ a recursive least squares (RLS) filter, which has a fast convergence, to track the traffic variation of its neighbors. Moreover, if the traffic of a UCAN is non-linear, non-stationary and non-Gaussian but changes slowly, a receiver could use a finite-impulse-response artificial neural network (FIR-ANN) [27] for traffic prediction.

In Fig. 7, we use the RLS filter as an example to introduce how to design a smart polling scheme for the RISM system. Here, we call the time interval for a receiver and its intended senders completing one round of communication (Phase 1 to Phase 6 in RISS) as a period, and then the details of a smart polling scheme are as follows:

(a) Before transmitting an RTR message, the receiver first estimates the number of data packets, $\sum_{k=1}^{K} \hat{d}_k^c$, currently cumulated on all K intended senders, where $\hat{d}_k^c = \hat{q}_k^c \times \Delta t_k^{c,j-1}$. Here, $\hat{q}_k^c$ is the estimation of current traffic load on sender k, and $\Delta t_k^{c,j-1}$ is the elapsed time since receiving the $\mathtt{ATS}_k^{j-1}$ packet. The receiver initiates a handshake if $\sum_{k=1}^{K} \hat{d}_k^c$ exceeds the threshold, which is denoted by γ, otherwise it will wait for longer time.

(b) Based upon $\mathtt{ATS}_k^j$, the receiver calculates the average traffic rate, denoted by q_k^j, of sender k during the last period according to $d_k^j / \Delta t_k^{j,j-1}$, where $\Delta t_k^{j,j-1}$ is the time interval between the receptions of $\mathtt{ATS}_k^j$ and $\mathtt{ATS}_k^{j-1}$. Then the receiver uses the latest l measurements, $\{q_k^j, \ldots, q_k^{j-l+1}\}$, to update the weight vector, which is denoted by $\mathbf{W}_k^j$, of its RLS filter. Here, l is referred to as the order of a RLS filter.

(c) The receiver applies the new weight, $\mathbf{W}_k^j$, to predict $\hat{q}_k^{j+1}$ — the traffic rate of sender k in the next period.

By adjusting the threshold γ in the traffic predictor, a UCAN could achieve a tradeoff between the queuing delay and the energy efficiency. More specifically, with a small γ, the data produced by each sender could be sent out timely, whereas, with a large γ, each round of communication could carry more data packets, which improves the energy efficiency.

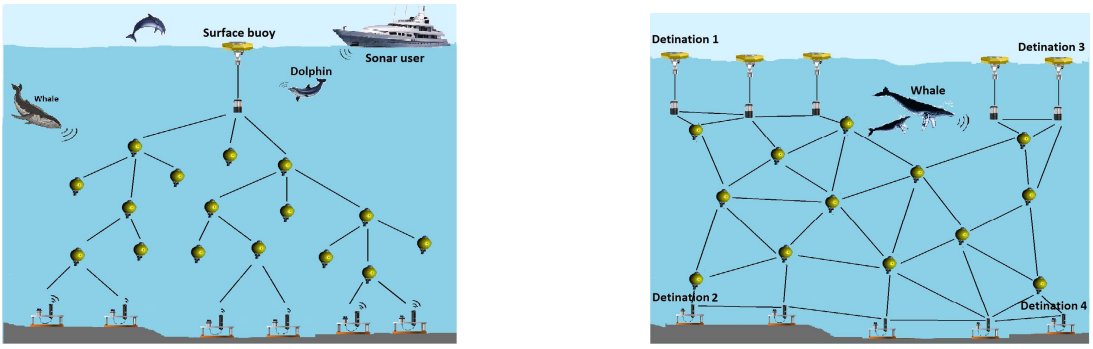
7 Simulations and Analysis

In this section, we conduct simulations to evaluate the performance of RISM system. The simulation platform is Aqua-Sim [28], an NS-2 based underwater network simulator. We extend Aqua-Sim to support multi-channel communications, dynamic power and channel assignments. Simulation results verify the enhanced network throughput, delivery delay and energy efficiency of RISM with smart polling. We also compare RISM with MMAC-CR [11] in different network settings. All results presented in this section are the average outcomes of multiple tests.

One of the real applications we considered in the simulation is an underwater target tracking network, where a UCAN with the bottom nodes, autonomous underwater vehicles (AUVs) and surface nodes is deployed in oceans to detect the presence of interesting targets, e.g., marine animals or submarines. If the nodes have not sensed any specific target yet, they stay in the sleep mode to save power. In this case, each node generates only a few packets periodically to update the routing table or to synchronize time. The sensor nodes become active once any target enters the network, thereby creating a bursty traffic. In order to prevent the UAN from interfering surrounding underwater systems (PUs), like sonar or marine animals, we apply the cognitive technique for environment-friendly data transmissions.

We evaluate the performance of RISM in two typical network topologies, as shown in Fig. 8. The tree network is usually applied for underwater data collection while the mesh topology network is a general example of ad hoc UANs. In the tree topology, we have 20 sub-sea nodes collecting and forwarding data to the sink node on the surface of the ocean, as illustrated in Fig. 8(a). In the partially connected mesh network, we deploy four hydrophones at four corners as sink nodes, as shown in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing



(a) Tree topology (b) Partially connected mesh topology

Fig. 8. **Two network topologies used in simulations.** (a) In the tree topology, 20 underwater nodes in the network deliver their data to a surface buoy. (b) In the partially connected mesh topology, 16 underwater nodes randomly select a node at four corners as the destination for data transmission.

Fig. 8(b). In the network, 16 sub-sea nodes generate and deliver data to a randomly selected hydrophone. In both topologies, the size of data packets is 250 bytes.

The average distance between neighboring nodes is 1 kilometer (uniformly distributed between 800 meters to 1200 meters) in both topologies. We set the maximum transmission range and the maximum transmission power of each node as 1.5 kilometers and 20 watts, respectively. The overall available channel bandwidth in the simulation is 30 kHz (from 1 kHz to 31 kHz), which is evenly divided into six subbands. We assign the lowest subband (1 kHz to 6 kHz) to be the CCC and assume this channel is not occupied by any PUs in the area. The remaining five subbands are used as data channel. In the network, we also deploy two PUs. Each of them randomly selects one amongst five data channels for its communication, and switches the communication channel every 60 seconds.

7.1 Performance Evaluation

We first evaluate the joint channel and power assignment in RISM by comparing the transmission rates in scenarios with and without optimal resource allocation. In this comparison, we let three intended senders transmit on four vacant channels to a common receiver.

Fig. 9 illustrates the average transmission rate of three senders. In the random allocation strategy, the receiver randomly allocates one channel to each sender with the maximum transmission power. The optimal assignment with continuous a_{nk}^t supposes that multiple senders could share one channel simultaneously, which is unrealistic in the real world, but could be considered as an upper bound of the transmission rate. The channel allocation algorithm applied in RISM is a sub-optimal solution, in which one channel is allocated to no more than one sender. From this figure, it is easy to observe a significant improvement on the average transmission rate in RISM system over the random assignment. In addition, the sub-optimal channel allocation is only slightly lower than the optimal solution. This verifies that RISM could utilize the channel and energy resources efficiently on the physical layer.

As we introduced in Section 5.2, RISM aims to eliminate the data collisions caused by the hidden terminal problems. However, we observe the existence of data

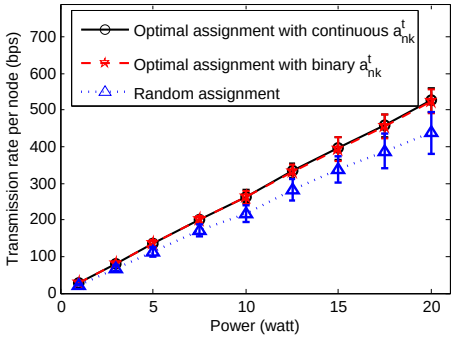


Fig. 9. The average transmission rate comparison among different resource allocation approaches.

interference in RISM when (a) the node fails to overhear the `ORDER` or `REPEAT` from surrounding users due to the collisions on the CCC; and (b) the node does not overhear the `ORDER` or `REPEAT` packets on time due to the long propagation delay in underwater communications. We plot the delivery ratio of data packets with respect to the variation of traffic load in different networks. Here, the packet delivery ratio is defined as the number of packets successfully received by receivers divided by the total number of packets sent by senders.

Fig. 10 demonstrates that RISM achieves 90% – 95% delivery ratio when there is no packet loss caused by the factors other than collisions. RISM can get higher delivery ratios, which means lower data collisions, in the mesh topology than in the tree network. In the tree topology, the data flow is gradually aggregated to the upper nodes in the network causing higher collision probabilities than the case in the mesh topology where the destination of data packets is one of a random node located in the four corners resulting in the scattered data flow. When the probability of packet decoding failure caused by the poor channel quality is increased to 10% or 20% in our simulations, the high channel loss becomes the dominated reason for a low packet delivery ratio, as shown in Fig. 10. However, it is worth noting that the end-to-end reliability of RISM can be guaranteed by the acknowledgment and retransmission mechanism regardless the packet loss on the hop-by-hop delivery.

Through changing the polling frequency of receivers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

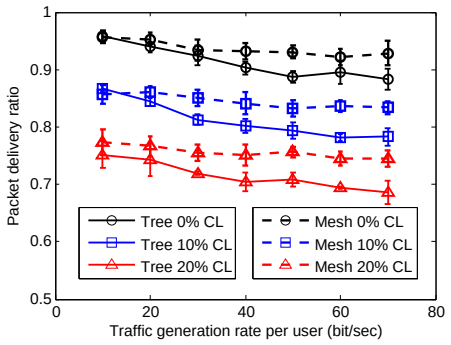


Fig. 10. Packet delivery ratio of RISM.

in RISM, a tradeoff exists between the packet delivery delay and the overhead of control messages. More specifically, if a receiver polls its intended senders frequently, data packets on a seder will have a short queuing delay, but at the cost of a low handshake efficiency, since each round of communication carries only a small number of data packets. On the other hand, if a receiver waits for a long time before requesting data, the queuing delay would be significant. However, it guarantees that enough data packets could be delivered in each round of handshake process, which improves the energy efficiency.

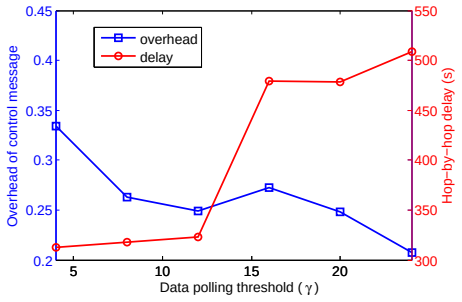


Fig. 11. Tradeoff between the delay and the overhead of control messages.

For RISM with the smart polling scheme, a receiver would not send RTR until it predicts that the total number of packets accumulated on its intended senders goes over the polling threshold. Therefore, we could change the polling frequency of a receiver by adjusting γ, the expected number of data to receive. In Fig. 11 we set the traffic generation rate as 40 bit/sec and show the tradeoff between the overhead of control messages and the hop-by-hop delay of RISM in the tree topology with respect to γ. In this figure, the overhead of control messages represents the percentage of energy consumption on control packets for each successful data transmission. It is the ratio of energy consumption on transmitting control messages to that on all packets (control plus data).

The results in Fig. 11 depict that when γ is small, frequent handshake consumes considerable energy on control message transmissions. The energy consumption remarkably reduces with the increment of γ, but results in a larger delivery delay, especially when γ is over 13. Hence, in the rest of this paper, we set the threshold for the adaptive polling as $\gamma = 13$ unless stated otherwise. Many factors, such as the node density, the traffic load and the requirements on network performance (energy efficiency and delivery delay), may affect the setting of γ. Unfortunately, we can not provide a theoretical result on how to optimize γ. A similar problem has been studied in the signal detection area [29].

In order to give an insight into the overhead of RISM, we show the average number of control packets sent for each successful data transmission in Fig. 12. We could observe a considerable decrease in the average number of control messages with the growth of traffic generation rate. Intuitively, when the traffic is light, the time it takes for senders to cumulate enough data packets for being polled will be inevitably long, which is not desirable in a delay-sensitive application. To tackle this problem, we set a maximal polling interval for the RISM system. When either the polling interval exceeds the maximal value or the number of cumulated packets reaches γ, the receiver will initiate a request for the data reception. For this reason, in a situation of low traffic rate, a handshake is most likely triggered by the maximal polling interval, at which time senders may have only a few data to send. On the other hand, when the traffic rate is high, the threshold, γ, controls the polling frequency. The average number of control packets used for each data transmission becomes stable as traffic generation rate grows.

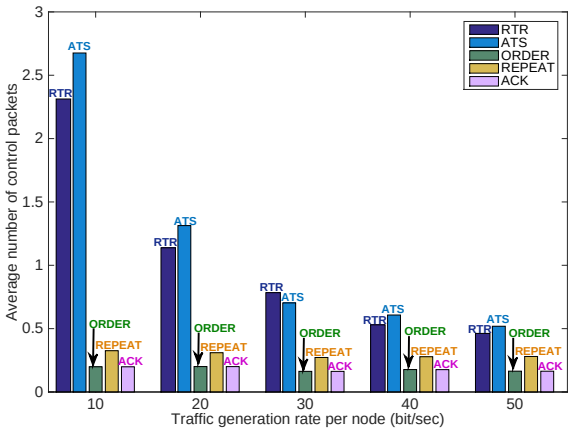


Fig. 12. The average number of control packets used for each successful data transmission in the tree topology.

7.2 Performance Comparison

To verify the advantages of the proposed system, we compare RISM with MMAC-CR, a representative MAC protocol for CR networks. A brief introduction on MMAC-CR has been presented in Section 2. We also run RISM with and without smart polling to validate the effectiveness of the adaptive data retrieving mechanism. In RISM with constant polling, the polling frequency is determined based upon the traffic rate measured at the initialization stage of a network. In this scheme, receivers send RTR periodically, and thus could not self-adapt to the variation of the network traffic.

The performance metrics we focus on in the comparison are network throughput, delay and overhead of control packet. The throughput is bits per second successfully received by each node in a network. The packet delivery delay consists of queueing delay, transmission delay and propagation delay. Considering the low throughput and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

the high collision probability among control messages in UANs, the queuing delay waiting for channel access is considerable and dominates the packet delivery delay. The overhead is calculated as the ratio of energy consumption on transmitting control messages to that on all packets (control plus data).

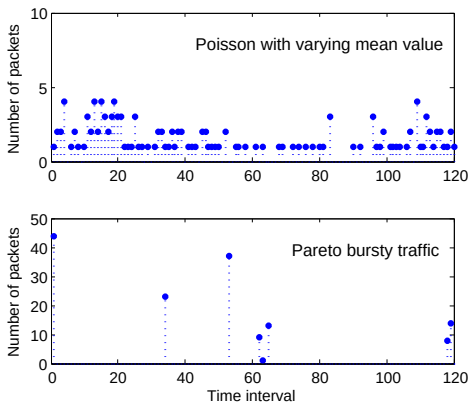


Fig. 13. An example of traffic patterns. (Average traffic generation rate is 0.02 packet per second for both scenarios. Each time interval is 50 seconds.)

In order to give comprehensive comparisons, we test the protocols with two different traffic patterns, namely, the Poisson process with slowly varying mean value and a Pareto bursty traffic, as shown in Fig. 13. The Poisson traffic is generally used to model the arrival process of a traffic in sensor networks where the data traffic is barely bursty. A varying mean value in the Poisson process can simulate the temporal variation of data collection rate. The Pareto traffic generator could well capture the traffic features of an event-driven sensor network, where sensor nodes generate a large amount of observations whenever a target event is detected.

Fig. 14 demonstrates the performance comparison in the tree network (Fig. 8(a)) where the data generation rate follows Poisson process. From Fig. 14 we observe that RISM outperforms the conventional sender-initiated MMAC-CR in all aspects. More specifically, smart polling assisted RISM achieves the highest throughput benefiting from the parallel reservation. By allowing receivers to negotiate with multiple senders in parallel, RISM mitigates the problem of a low handshake efficiency in UCANs caused by the long propagation delay and the long preamble in acoustic modems. Moreover, as shown in Fig. 14(a), the throughput improvement of RISM dramatically increases with the growth of network traffic generation rate. In applications with heavy traffic loads, RISM could provide high network throughput, which makes it a promising solution for efficient spectrum management. Compared to RISM without smart polling, the traffic prediction scheme grants receivers the capability to dynamically poll senders with the varying mean value of Poisson traffic, thereby leading to a significant throughput enhancement.

Furthermore, RISM with smart polling allows a network to accommodate a high traffic rate with relatively low delivery delays, as shown in Fig. 14(b). In a UCAN with heavy traffic load, the packet queuing delay is the main source of packet delivery delays in RISM without smart polling. The traffic prediction enables each receiver to retrieve data from surrounding senders not only more efficiently but also more timely than the RISM without smart polling, thereby significantly reducing the delivery delay in scenarios with high traffic generation rates.

Another advantage of RISM over sender-initiated protocols is the low overhead of control messages. When receivers start negotiation with their surrounding senders, RISM works as a "semi-centralized" system, where the spectrum sensing, spectrum sharing and dynamic power control could efficiently share control packets with each other. As illustrated in Fig. 14(c), RISM with and without traffic prediction have comparable control overhead, since both schemes tend to wait for enough cumulated data packets before starting a handshake process for better energy efficiency. MMAC-CR, by contrast, has almost twice larger overhead than RISM, as MMAC-CR has to schedule separate control messages for spectrum sensing, multi-channel rendezvous, and dynamic power control.

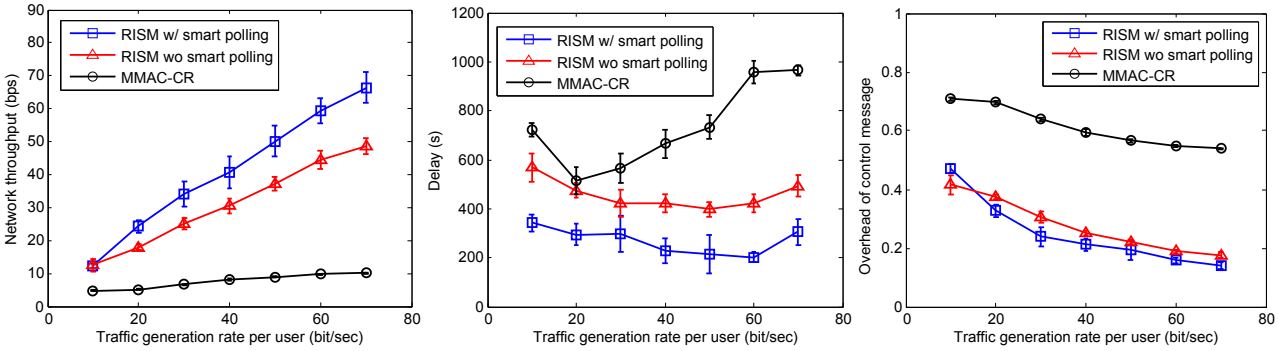
Fig. 15 uses the same setting as Fig. 14, but the network topology is changed from the tree network to the mesh network as shown in Fig. 8(b). By comparing the results in Fig. 14 and Fig. 15, we could observe that RISM has lower throughput in the mesh network than in the tree network. This is because, in the tree topology, underwater nodes generate and forward data packets to a common destination, namely, the surface node resulting in the aggregated data flow. Therefore, a receiver could easily get plenty of packets from nodes beneath it in each period, which improves the handshake efficiency. In the mesh network, on the other hand, the destination of data packets is one of a random node located in the four corners, which "diluents" the traffic. Therefore, a receiver usually retrieves fewer data packets within a given period in a mesh network than that in a tree network, which reduces the handshake efficiency and leads to a lower nodal throughput for RISM.

By contrast, MMAC-CR can achieve higher throughput in a mesh networks than that in tree networks. Intuitively, in tree networks, multiple senders choosing the same relay node causes heavy congestion for the channel access. In mesh network, however, the data packets are scattered resulting in lower collision probability on control messages. Due to the similar reason, RISM has a much longer delay in the mesh network than that in the tree network, whereas MMAC-CR can achieve a shorter delay in the mesh network. Although RISM works more efficiently in a tree network, it still outperforms the sender-initiated MMAC-CR in terms of network throughput and energy efficiency in the mesh network.

By comparing the results in Fig. 14 and Fig. 16, we could realize how a traffic pattern affects the performance of different protocols. For Pareto bursty traffic, the instantaneous traffic is bursty but the average data generation rate is constant in the simulation. As for Poisson traffic, we periodically change the average traffic generation rate from $0.5\times$ to $1.5\times$ of the mean value in each test.

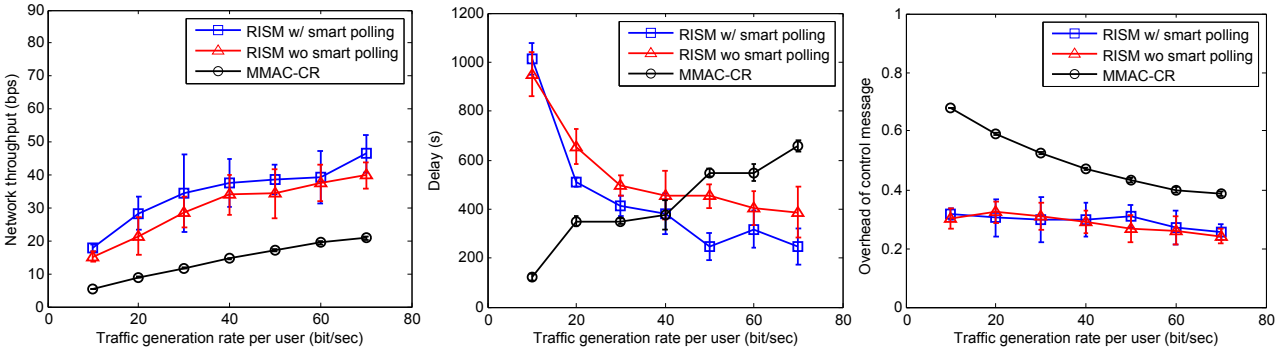
We observe that RISM with smart polling has comparable performance with both Poisson traffic and bursty traffic, i.e. RISM is barely affected by the traffic pattern. With an assistance of traffic prediction, the smart polling mechanism could capture the variations in the network traffic, thereby making receivers in RISM request data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing



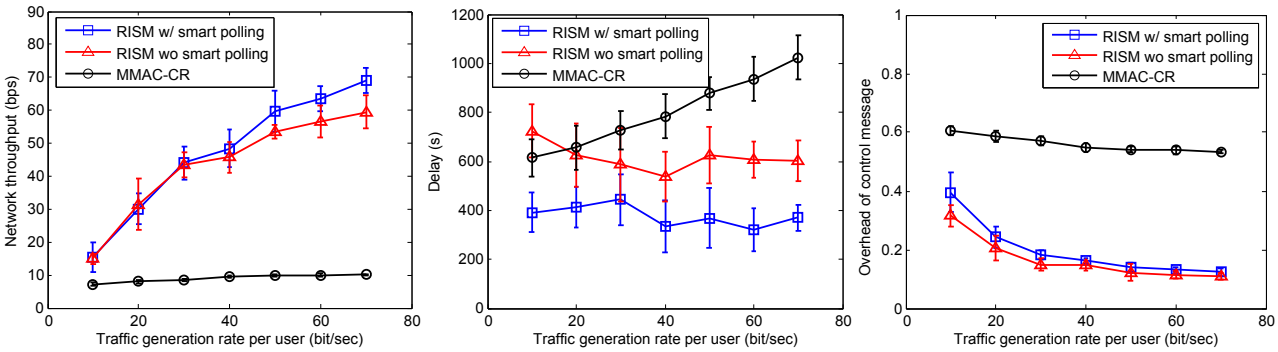
(a) Network throughput (b) Delay (c) Overhead

Fig. 14. Performance comparison for Poisson traffic with slowly varying mean value in tree topology.



(a) Network throughput (b) Delay (c) Overhead

Fig. 15. Performance comparison for Poisson traffic with slowly varying mean value in mesh topology.



(a) Network throughput (b) Delay (c) Overhead

Fig. 16. Performance comparison for Pareto bursty traffic in tree topology.

at a proper time. RISM without smart polling, on the other hand, has much lower throughput than RISM with smart polling when the network traffic varies with time, as shown in Fig. 14(a). The difference on the throughput of RISM with and without smart polling becomes less significant with bursty traffic. This indicates that the adaptive polling could bring more enhancement to RISM when the average traffic rate is more dynamic.

It is worth noting that in simulations we do not take the guard time into consideration. More specifically, in order to prevent unexpected collisions among control messages and data packets from an inaccurate distance measurement, a synchronization error and the drifting of acoustic nodes with an ocean current, a guard time among sending packets is usually required in a UCAN. In this situation, the absolute values of throughput, delay and overhead of MAC protocols showed in this section may be lower than that in a real world. The relative performance among different protocols, however, will not change a lot.

8 Conclusions

In this paper, we designed a receiver-initiated spectrum management system, RISM, to achieve environment-friendly and spectrum-efficient communications for underwater cognitive acoustic networks. In RISM, receivers replace the role of senders in conventional protocols as

an initializer of a handshake process. This allows receiver in each round of handshake to request packets from multiple senders in parallel.

In addition, in RISM, the three components, i.e., the cooperative spectrum sensing, spectrum sharing and spectrum decision, do not generate control messages separately as independent pieces. Instead, they share control packets with each other without incurring additional control overhead, which significantly improves the negotiation efficiency considering the high latency in UCANs and the long preamble in acoustic modems.

However, there is a unique challenge in receiver-initiated approaches for receivers to decide when to poll without prior knowledge of the data cumulation on senders. This issue is tackled by adopting the traffic predictor. A receiver in RISM could smartly poll senders adapting to the variation of senders' traffic loads. By employing the smart polling scheme, the receiver could initiate handshakes timely to reduce the packet queuing delay while constraining the energy consumption on transmitting control packets.

Simulation results show that the performance of RISM with smart polling scheme outperforms MMAC-CR, a representative CR based MAC protocol. Specifically, the throughput of RISM is $6\times$ higher than MMAC-CR, while the hop-by-hop delay and overhead of control packet are only $0.25\times$ and $0.3\times$ of MMAC-CR. Moreover, RISM could work better in a tree network than in a mesh network. The throughput in the former scenario is near $2\times$ than that in the later one, while maintaining comparable hop-by-hop delay and overhead of control packet.

We believe that RISM is a promising system that enables cognitive technique to work efficiently and environment-friendly in UANs.

Acknowledgments

This work is supported in part by the US National Science Foundation under Grant No.1331851. The authors would like to thank the editors and reviewers for their thoughtful, constructive suggestions and comments. Dr. Zheng Peng is the corresponding author.

References

[1] Y. Luo, L. Pu, M. Zuba, Z. Peng, and J.-H. Cui, "Challenges and opportunities of underwater cognitive acoustic networks," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–14, 2014.

[2] N. Baldo, P. Casari, and M. Zorzi, "Cognitive spectrum access for underwater acoustic communications," in *Proceedings of IEEE ICC Workshops*, 2008, pp. 518–523.

[3] W. Yonggang, T. Jiansheng, P. Yue, and H. Li, "Underwater communication goes cognitive," in *Proceedings of IEEE OCEANS-Quebec City*, 2008.

[4] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: a POMDP framework," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 589–600, 2007.

[5] J. Jia, Q. Zhang, and X. Shen, "HC-MAC: a hardware-constrained cognitive MAC for efficient spectrum management," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 106–117, 2008.

[6] H.-B. Chang and K.-C. Chen, "Auction-based spectrum management of cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1923–1935, 2010.

[7] W. Hu, D. Willkomm, M. Abusubaih, J. Gross, G. Vlantis, M. Gerla, and A. Wolisz, "Cognitive radios for dynamic spectrum access-dynamic frequency hopping communities for efficient IEEE 802.22 operation," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 80–87, 2007.

[8] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 533–545, 2008.

[9] C. Cordeiro and K. Challapali, "C-MAC: A cognitive MAC protocol for multi-channel wireless networks," in *Proceedings of IEEE DySPAN*, 2007, pp. 147–157.

[10] Q. Chen, Y.-C. Liang, M. Motani, and W.-C. Wong, "A two-level MAC protocol strategy for opportunistic spectrum access in cognitive radio networks," *IEEE Transactions on vehicular technology*, vol. 60, no. 5, pp. 2164–2180, 2011.

[11] M. Timmers, S. Pollin, A. Dejonghe, L. Van der Perre, and F. Catthoor, "A distributed multichannel MAC protocol for multihop cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 446–459, 2010.

[12] Z. Zhou, Z. Peng, J.-H. Cui, and Z. Jiang, "Handling triple hidden terminal problems for multichannel MAC in long-delay underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 139–154, 2012.

[13] L. Pu, Y. Luo, Y. Zhu, Z. Peng, J.-H. Cui, S. Khare, L. Wang, and B. Liu, "Impact of real modem characteristics on practical underwater MAC design," in *Proceedings of IEEE OCEANS-Yeosu*, 2012.

[14] S. Le, Y. Zhu, J.-H. Cui, and Z. Jiang, "Pipelined transmission MAC for string underwater acoustic networks," in *Proceedings of ACM WUWNet*, 2013.

[15] A. De Domenico, E. C. Strinati, and M.-G. Di Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 21–44, 2012.

[16] K. Kim, I. A. Akbar, K. K. Bae, J.-S. Um, C. M. Spooner, and J. H. Reed, "Cyclostationary approaches to signal detection and classification in cognitive radio," in *Proceedings of IEEE DySPAN*, 2007, pp. 212–215.

[17] B. M. Sadler and A. V. Dandawate, "Nonparametric estimation of the cyclic cross spectrum," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 351–358, 1998.

[18] J. Lundén, V. Koivunen, A. Huttunen, and H. V. Poor, "Collaborative cyclostationary spectrum sensing for cognitive radio systems," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4182–4195, 2009.

[19] I. E. Telatar, "Capacity of multi-antenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, pp. 585–595, 1999.

[20] M. Chitre, J. Potter, and O. S. Heng, "Underwater acoustic channel characterisation for medium-range shallow water communications," in *Proceedings of IEEE OCEANS*, 2004.

[21] W. B. Yang and T. C. Yang, "Characterization and modeling of underwater acoustic communications channels for frequency-shift-keying signals," in *Proceedings of IEEE OCEANS-Boston*, 2006.

[22] F. Ruiz-Vega, M. C. Clemente, P. Otero, and J. F. Paris, "Ricean shadowed statistical characterization of shallow water acoustic channels for wireless communications," *arXiv preprint arXiv:1112.4410*, 2011.

[23] C. Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch, "Multiuser OFDM with adaptive subcarrier, bit, and power allocation," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 10, pp. 1747–1758, 1999.

[24] F. F. Digham, "Joint power and channel allocation for cognitive radios," in *Proceedings of IEEE WCNC*, 2008, pp. 882–887.

[25] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.

[26] B. Yegnanarayana, *Artificial neural networks*. Prentice Hall of India Private Limited, 2009.

[27] E. A. Wan, "Finite impulse response neural networks with applications in time series prediction," Ph.D. dissertation, Stanford University, 1993.

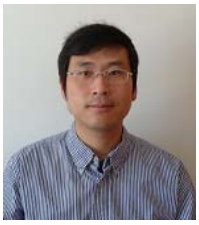
[28] Y. Zhu, X. Lu, L. Pu, Y. Su, R. Martin, M. Zuba, Z. Peng, and J.-H. Cui, "Aqua-Sim: an NS-2 based simulator for underwater sensor networks," in *Proceedings of ACM WUWNet*, 2013.

[29] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.

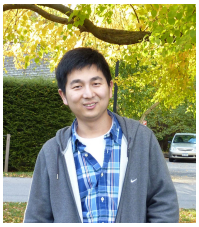
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2544757, IEEE Transactions on Mobile Computing

Yu Luo received the B.S. degree and the M.S. degree in electrical engineering from the Northwestern Polytechnical University, China, in 2009 and 2012, respectively. In 2015, he received the Ph.D. degree in computer science and engineering from University of Connecticut, Storrs. His major research focus on the cross-layer design for cognitive acoustic networks and underwater acoustic network. He is a Co-recipient of IFIP Networking 2013 best paper award.

Lina Pu received the B.S. degree in electrical engineering from the Northwestern Polytechnical University, Xi'an, China in 2009 and the Ph.D. degree in Computer Science and Engineering from University of Connecticut, Storrs. Her research interests lie in the area of MAC design, performance evaluation and experimental study for underwater acoustic networks. She owned IFIP Networking 2013 best paper award.



Haining Mo received his M.E. from Computer Science and Technology, Harbin Institute of Technology, Harbin, China in 2006. In 2014, he received his Ph.D. degree from Computer Science and Engineering, University of Connecticut. His research interests include reliable data transfer, medium access control and protocol design, implementation and test in Underwater Acoustic Networks.



Yibo Zhu received the BEng and MEng degrees in computer science, both from Xi'an Jiaotong University, China, in 2006 and 2009, respectively. In 2014, he received his Ph.D degree in computer science and engineering from University of Connecticut. His main research interests include MAC/Routing protocol designs for underwater acoustic networks.

Zheng Peng received a Bachelors degree in Control Theory and a Bachelors degree in Computer Science from Zhejiang University, Hangzhou, China in 2002. He obtained a Masters degree in Computer Science from University of Electrical Science and Technology of China in 2005, and later his Ph.D. from University of Connecticut, Storrs, USA. His main research interests cover the design, modeling, optimization, development and experimental evaluation of the wireless sensor networks, embedded/distributed systems, cyber security and their applications in the unique underwater environment

Jun-Hong Cui received her B.S. degree in Computer Science from Jilin University, China in 1995, her M.S. degree in Computer Engineering from Chinese Academy of Sciences in 1998, and her Ph.D. degree in Computer Science from UCLA in 2003. Currently, she is on the faculty of the Computer Science and Engineering Department at University of Connecticut.

Her research interests cover the design, modeling, and performance evaluation of networks and distributed systems. Recently, her research mainly focuses on exploiting the spatial properties in the modeling of network mobility, scalable and efficient communication support in overlay and peer-to-peer networks, algorithm and protocol design in underwater sensor networks. She co-founded ACM WUWNet. Jun-Hong is a member of ACM, ACM SIGCOMM, ACM SIGMOBILE, IEEE, IEEE Computer Society, and IEEE Communications Society.