J Sign Process Syst (2017) 89:3–12
DOI 10.1007/s11265-016-1138-1

A Slotted-FAMA based MAC Protocol for Underwater Wireless Sensor Networks with Data Train

Senlin Zhang[1,2] · Liangfang Qian[2] · Meiqin Liu[1,2] · Zhen Fan[2] · Qunfei Zhang[3]

Received: 8 March 2016 / Revised: 25 March 2016 / Accepted: 6 April 2016 / Published online: 15 April 2016

Abstract Underwater wireless sensor networks are significantly different from terrestrial wireless sensor networks in that sound is mainly used as the communication medium. The limited bandwidth, long propagation delay and high bit error rate pose great challenges in Media Access Control (MAC) protocol design for underwater wireless sensor networks. In this paper, we propose a Slotted-FAMA based MAC protocol for underwater wireless sensor networks with data train, called SFAMA-DT. It improves the channel utilization by forming a train of data packets of multiple transmission pairs during each round of simultaneously handshakes, which overcomes the multiple RTS attempts problem of Slotted-FAMA in high traffic environments and greatly reduces the relative proportion of time wasted due to the propagation delays of control packets. Our simulations show that the SFAMA-DT is able to achieve much higher throughput than the Slotted-FAMA protocol.

Keywords Underwater wireless sensor networks · Medium access control (MAC) · RTS/CTS sorting · Data train · Throughput

✉ Meiqin Liu
liumeiqin@zju.edu.cn

[1] State Key Laboratory of Industrial Control Technology, Hangzhou 310027, China

[2] College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China

[3] School of Marine Science and Technology, Northwestern Polytechnical University, Xi’an 710072, China

1 Introduction

Underwater Wireless Sensor Networks (UWSNs) have gained tremendous attention in recent years [1–3]. One important reason is that they can be used for a broad range scientific exploration, including but not limit to underwater data collection, environment monitoring, submarine detection, and coastline protection [4, 5].

Underwater networks use the acoustic channel and face new research challenges, such as limited bandwidth, long propagation delay, high bit error rate and high energy consumption [6–9]. As a core module of UWSNs, medium access control (MAC) is a channel control mechanism that allows multiple nodes to communicate through a shared medium. It is critical to provide valid and efficient MAC solutions for UWSNs. Many MAC protocols have been proposed for UWSNs in the past decade. In [10], authors proposed a protocol called DACAP, which allows a sender to use different handshake lengths for different receivers to improve the channel utilization. In [11], a novel handshaking-based MAC protocol named SRCR was proposed for multi-hop underwater acoustic networks. In the SRCR protocol, neighbors of the sender and the receiver are allowed to transmit packets opportunistically through the concurrent reservation of the sender and the receiver, resulting in good performance in both throughput and delay. Luo et al. proposed CT-MAC [12], which leveraged a special relay mechanism. Via this mechanism, CT-MAC significantly improves the network performance in terms of channel utilization and energy efficiency.

These underwater protocols were proposed since the first ALOHA protocol. An improvement to the original ALOHA protocol is Slotted-ALOHA. By restricting packet transmission to predetermined time slots, Slotted-ALOHA decreases the vulnerable time during which a collision can occur. However, ALOHA becomes inefficient in busty traffic, because trying to resolve collisions by retransmissions limits the throughput and increases the power consumption [13]. To achieve better performance, some random access-based MAC protocols are designed specifically for UWSNs. For example, UWAN-MAC [14] and ALOHA-CA [15] consider long propagation delays and leverage short control packets to alleviate collisions. Nevertheless, they can only effectively work in single-hop scenarios.

To provide valid solutions for multi-hop UWSNs, some handshake-based MAC protocols have been proposed. The MACA protocol was proposed by Karn [16], which uses two signaling packets called request-to-send (RTS) and clear-to-send (CTS). When a node wants to transmit a packet, it sends an RTS control packet intended for the receiver. This receiver responds with a CTS control packet. Any node that overhears a CTS packet defers its transmission to avoid collision. If a node overhears an RTS packet but not a CTS packet, it continues transmitting its own packet since it is out of range of the receiver. Therefore, both the hidden and the exposed node problems can be solved.

However, a collision in MACA protocol may occur due to different packet delays. To overcome this problem, The FAMA protocol [17] elongates the transmission delays of RTS/CTS packets to make MACA work in long propagation delay characterized networks, such as UWSNs. However, the elongated control packets are usually of seconds considering the low sound speed in water and long maximum transmission range. Also note that acoustic wave is one type of mechanical waves, and it is more energy-consuming to be generated than radio wave. As a result, it is too costly to transmit such long control packets in terms of energy consumption

Slotted-FAMA was proposed in [18] to solve this problem, which makes nodes share a global time synchronization and divides time into slots. The slot length is equal to the maximum propagation delay plus the transmission time of a control packet, which assures that only control packets may collide and that the transmission of data packets is collision-free. While Slotted-FAMA is very effective in reducing collisions, it is not suitable for high traffic environments because of the multiple RTS attempts problem which will be discussed later. Moreover, this approach severely limits throughput, because of the guard times required between subsequent slots in order to accommodate for the maximum propagation delay within a given coverage radius.

In this work, we propose a Slotted-FAMA based MAC protocol, called SFAMA-DT, to overcome the problems of Slotted-FAMA mentioned above. SFAMA-DT introduces RTS/CTS sorting scheme to set multiple handshakes simultaneously, and then form a train of data packets of multiple transmission pairs during each round of simultaneously handshakes. The main contribution of SFAMA-DT is that it overcomes the multiple RTS attempts problem and increases the network throughput by transmitting a train of data packets. Although the idea of data train has been used in several MAC proposals such as in [19] and [20], our work goes one step further as the data train is actually formed for multiple transmission pairs simultaneously.

2 Original Slotted-FAMA Overview

Slotted-FAMA makes nodes share a global time synchronization and divides time into slots, forcing any transmission to be initiated only at the beginning of a slot. As shown in Fig. 1, when a node has a data packet to send, it waits for the beginning of the next slot and transmits an RTS packet, which can be received by all neighbors. When the destination node receives the RTS packet, it responds with a CTS packet at the beginning of the following slot. If the source has received the CTS correctly, it starts to transmit the data packet, and backs off otherwise. Once the data packet has been sent, the source waits for the corresponding acknowledgement (ACK) to arrive in the following slot.

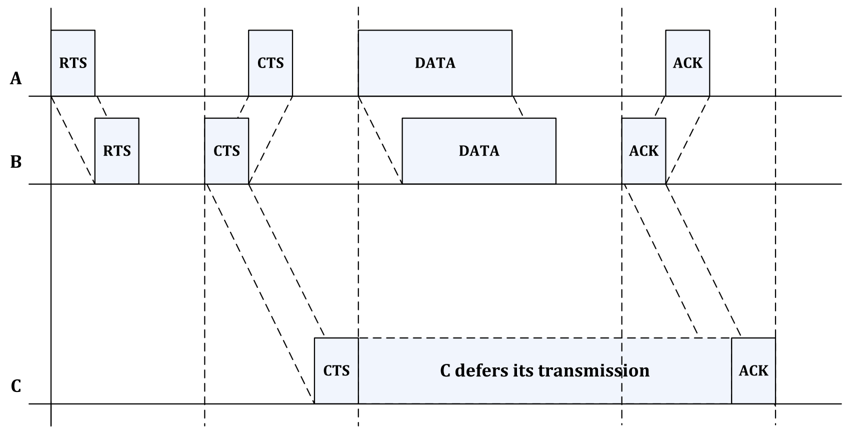


Figure 1 A successful handshake between terminals A and B in Slotted-FAMA.

When a node detects carrier on the channel it goes to a receiving state while it is receiving the packet. After receiving an RTS packet in tended for another station (called xRTS packet), the node must wait two slots (long enough for the receiver to send a CTS and the sender to start transmitting data). After receiving a CTS packet intended for another station (xCTS packet), the node must wait long enough to allow the other station to transmit the entire data packet and receive the corresponding ACK. After receiving a data packet intended for another station (xDATA packet), the node must wait long enough to allow the reception of the subsequent ACK. After hearing an ACK packet intended for another station (xACK packet), the node only has to wait until the end of the slot since the data transmission has successfully ended.

In this way, the transmission of data packets in Slotted-FAMA is collision-free. However, Slotted-FAMA will have the problem of multiple RTS attempts in high traffic situations. In these situations, the probability of two or more nodes sending RTS at the beginning of the same slot can be high. As shown in Fig. 2, node A and node D both send RTS to their destination nodes. According to the transmission rules in Slotted-FAMA, neither control packet nor data packet can be transmitted in the following two slots, since (1) node A and node D go to the backoff state for a random number of slots and (2) node B, node C and node E have to wait two slots for the receiver to send a CTS and the sender to start transmitting data, which will not happen actually. So in the following slot 2 and slot 3, nothing can be done but wait. Moreover, due to the long propagation delay in underwater acoustic networks, a slot length, which equals to the maximum propagation time plus the duration of a control packet, is not short. So, during the following two slots, it is probable that two or more nodes generate a new packet to be sent. In this case, node B and node C have a packet ready to transmit during slot 2 and slot 3, and thus send RTS at the beginning of slot 4. Then the same situation as slot 2 and slot 3 appears for slot 5 and slot 6. It is true that the probability of node B and node C both generating a new packet during slot 2 and slot 3 is low in this case. But when in high traffic environment, where there are dozens or even hundreds of nodes, the probability of two of these nodes transmitting RTS packets at the same slot becomes much higher.

In addition to this, Slotted-FAMA also suffers the low throughput problem. A successful data transmission requires at least 3 additional slots for control packets. This means a lot of time will be wasted to the propagation delays of control packets for each data transmission, which greatly limit the network throughput.

To overcome these problems, we adapt the original Slotted-FAMA for use in high traffic networks by forming a train of data packets of multiple transmission pairs during each round of simultaneously handshakes. We will call this improved MAC protocol SFAMA-DT.

3 The Proposed Protocol: SFAMA-DT

In this section, we will first brief the basic idea of SFAMA-DT. Then we describe two new schemes (i.e. RTS/CTS sorting and data train) of SFAMA-DT in detail. After that, we discuss some potential issues.

3.1 SFAMA-DT Overview

As Slotted-FAMA, SFAMA-DT is also a reservation and slotted MAC protocol, using 4 types of packets(i.e. RTS, CTS, DATA and ACK) and the same slot mechanism. Each packet should be transmitted at the beginning of a time slot.

Here we use an example to show how SFAMA-DT works. As shown in Fig. 3, there are four nodes in this network. Two of them are source nodes and the others are

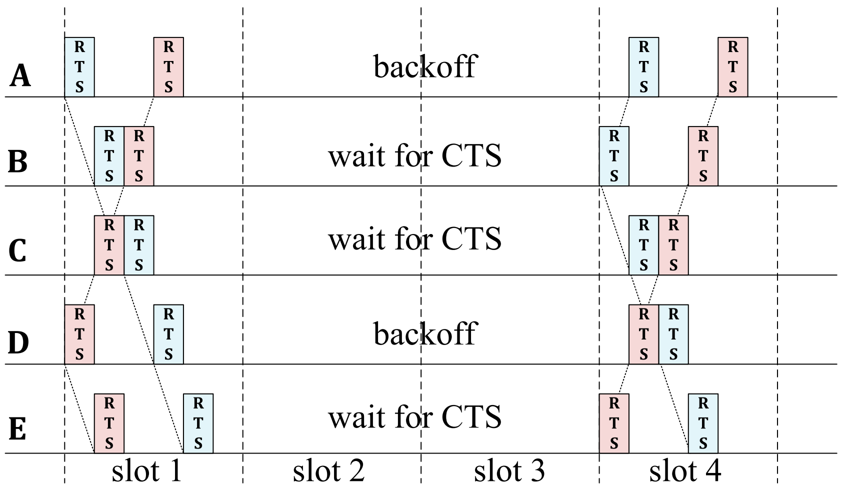


Figure 2 Multiple RTS attempts problem of Slotted-FAMA.

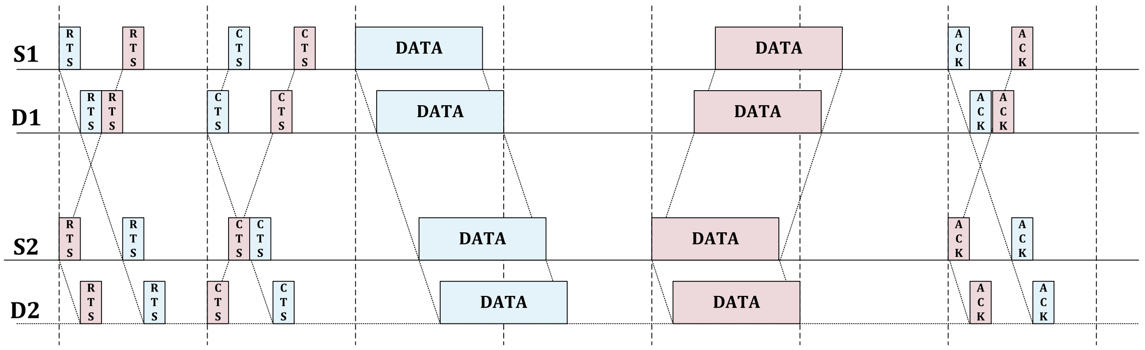


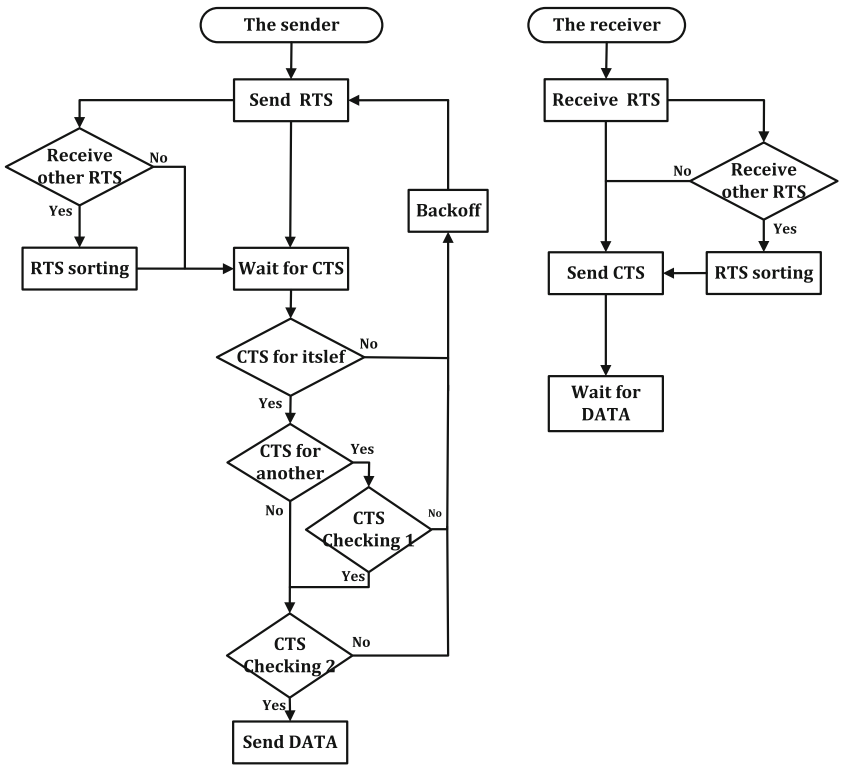
Figure 3 Timing diagram of SFAMA-DT for UWSNs.

destination nodes. At the beginning of a slot, S_1, S_2 wish to send a packet destined to D_1, D_2 respectively. These four nodes are within the maximum transmission range of each other (one-hop network).

In SFAMA-DT, when a node requires to send a packet, it will wait until the beginning of the next slot and then transmits an RTS packet if the channel is idle. So at the beginning of the first slot, S_1, S_2 send $(RTS)_{S_1 \to D_1}$, $(RTS)_{S_2 \to D_2}$ respectively. Here we use $(TYPE)_{sender \to receiver}$ to denote a packet. All the destinations (D_1, D_2) will receive both of these packets. Unlike Slotted-FAMA defers all the transmissions when multiple RTS attempts occur, SFAMA-DT introduces RTS/CTS sorting scheme(which will be described later) to keep transmissions going on. By RTS/CTS sorting scheme, When more than one nodes send RTS packets in the same slot, they can be sorted in a queue.

The node which receives the RTS intended for itself always responds with a CTS packet, no matter how many RTS packets have been received. In this case, D_1 receives $(RTS)_{S_1 \to D_1}$ and thus responds with $(CTS)_{D_1 \to S_1}$, and D_2 receives $(RTS)_{S_2 \to D_2}$ and thus responds with $(CTS)_{D_2 \to S_2}$. When S_1 and S_2 have received the CTS packets correctly, they start to transmit the data packets orderly according to the sorted data train. In this case, we assume that S_1 transmits in front of S_2. As shown in Fig. 3, after S_1 completing its data transmission, S_2 starts to send its data packet at the beginning of the arranged slot. When the last sender finishes its data transmission, all the receives which

Figure 4 State machine of RTS/CTS sorting scheme.



receive the data packets correctly respond with the ACK packets to complete the entire transmission.

Based on RTS/CTS sorting and data train schemes, more data transmissions can be completed during each round of simultaneously handshakes, which overcomes the multiple RTS attempts problem and greatly increases the network throughput. After giving the overall picture of SFAMA-DT, we will discuss the detailed design of protocol.

3.2 RTS/CTS Sorting

In the original Slotted-FAMA protocol, when a node, which has already sent an RTS, receives another RTS packet, it defers its transmission and backs off for random slots immediately. The node which receives more than one RTS packets will ignore the RTS packet intended for itself, and will wait for two slots. In SFAMA-DT, we add RTS/CTS sorting scheme in both send and receive links to handle multiple RTS attempts problem.

The RTS/CTS sorting scheme has two phases, namely, RTS sorting and CTS checking. Figure 4 depicts how these two phases work.

In the RTS sorting phase, when a node has a new packet to send and the channel is idle, it makes an RTS packet and adds a sorting number, called S-number, into the RTS packet. In SFAMA-DT, we obtain S-numbers by random number generation. Then the node sends the RTS packet with S-number at the beginning of the following slot. In this way, each RTS packet will have a S-number with it. The RTS packet also contains the transmission time of the data packet, which will be used in transmitting the data train. For a sender, which has already sent an RTS packet, when it receives one or more RTS packets, it makes its own S-number sort with the S-numbers of other RTS packets to form a sorted data train. For a receiver, when it receives more than one RTS packets and one of them is intended for itself, it also makes the S-number in the RTS packet intended for itself sort with the S-numbers of other RTS packets to form a sorted data train. Then each sender or receiver obtains the sorted train of transmission pairs in the RTS attempt slot.

However, because of the hidden terminal problem (which will be discussed later), a node which has received an RTS packet may not receive its responding CTS packet, or a node may receive an CTS packet without a previous RTS packet received, or different nodes may receive different numbers of RTS/CTS packets in the same slot. Such mismatches of RTS and CTS packets may result in different data trains at different nodes. So we need the CTS checking phase to keep away from these mismatches. As shown in Fig. 4, we have two steps in CTS checking phase. In step one, when a receiver has received an RTS packet intended for itself, it makes an CTS packet and adds the information of its sorted train into the CTS packet. When the senders receive this CTS, each of them check if the train's information of this CTS packet is the same as its own or not.If the trains are the same, the transmissions keep going on, otherwise they shall be canceled. In addition, we need checking step two to completely eliminate the mismatches. At the end of CTS attempt slot, each of senders and receivers checks the RTS and CTS packets it received. If they can entirely match, the sender or the receiver continues to completing the data transmission, otherwise it cancels its data transmission. This two checking steps make sure all the senders and receivers have the same sorted train.

After RTS/CTS sorting, each transmission pair sends data according to the sorted data train.

3.3 Data Train

In this section, we explain how data transmissions are sent in sequence and when receivers send ACK or NACK packets to acknowledge data receptions. Table 1 shows the notations that are used.

As mentioned above, in addition to S-number, the RTS packet also contains the transmission time of the data packet, $T_{data,i}$. So all the nodes in the train would know the slot number that each send-receive pair takes to complete its data transmission, $N_{data,i}$.

Let us denote the beginning time of the entire transmission (including RTS/CTS attempts, data transmissions and ACK corresponding) t_0. The first node in the train sends data packet at $t_{data,1}$, which is $t_{data,1} = t_0 + 2T_{slot}$.

For node i, it transmit its data packet at $t_{data,i}$, which can be calculated by using the following equations:

$$t_{data,i} = t_0 + 2T_{slot} + \sum\nolimits_{1}^{i-1} \cdot N_i \cdot T_{slot} \qquad (1)$$

Then it needs to wait for other nodes in the train to finish their data transmissions and thus responds with the ACK or

Table 1 Notations used for explaining the protocol.

Notation	Description
n	Total number of transmission pairs
i	The i-th node in the sorted train
T_{slot}	The duration of one slot
T_{CTS}	The duration of a control packet
$T_{data,i}$	The duration of a data packet of node i
$N_{data,i}$	The slot number of the data packet of node i
t_0	Time to send the RTS packet
$t_{data,i}$	Time at which node i sends the data packet
t_{ACK}	Time to send the ACK or NACK packet

NACK packet to acknowledge data reception. The time at with the receivers send their corresponding ACK or NACK is as follows:

$$t_{ACK} = t_0 + 2T_{slot} + \sum\nolimits_1^n \cdot N_i \cdot T_{slot} \qquad (2)$$

3.4 Scheduling for Neighbors

To avoid data transmission interference, all the neighbors should keep quiet for the ongoing transmission. When a neighboring node detects carrier on the channel it goes to a receiving state while it is receiving the packet. The type of packet received will determine the neighbor actions as listed blew.

If the received packet is an RTS packet (or more than one RTS packets) intended for another node (xRTS packet), the node must wait two slots (long enough for the receiver to send a CTS and the sender to start transmitting data). If after this time no carrier is sensed, the node returns to the idle state.

If the received packet is a CTS packet intended for another node (xCTS packet), the node must wait long enough to allow the other node to transmit the entire data packet and receive the corresponding ACK. If it receives more than one CTS packets, it waits for all the senders to complete their data transmissions in order and receive the ACK packets.

If the received packet is a data packet intended for another node (xDATA packet), the node must wait long enough to allow the reception of the subsequent ACK packet. If more than one data packets are received, the node wait until the ACK packets are received.

If the received packet is an ACK packet intended for another node (xACK packet), the node only has to wait until the end of the slot since the data transmission has successful ended.

If a terminal senses interference in the channel, a collision is assumed. Since it doesn't know which packets have collided, the worst assumption is made, and it acts as if it had received an xCTS packet which is the situation that requires a longer wait.

3.5 Discussions

Besides the overall designs illustrated above, there are some special problems that should be discussed, including the hidden terminal problem and the RTS/CTS collision problem.

3.5.1 Hidden Terminal Problem

In general, a situation of a hidden terminal occurs when one node cannot sense one or more nodes that can interfere with its transmission. In our work, we regard the situation that one node cannot sense one or more transmissions which should be involved in the data train as the hidden terminal problem. Figure 5 shows the hidden terminal problem in SFAMA-DT.

As we mentioned in Section 3, without CTS checking phase, only RTS sorting phase could not handle the hidden terminal problem which may result in incomplete data train. As show in Fig. 5, without CTS checking phase, both sender $S1$ and sender $S2$ cannot sense the transmission of each other. So each of them makes the data train only consist of itself and thus sends data at the beginning of the following slot. Then data collisions occur. With CTS checking phase, both $S1$ and $S2$ cancel their transmissions to void data collisions.

However, simply canceling the transmissions is not conducive to improving the network throughput. So we wish to find a more effective way to handle the hidden terminal problem in the future. But in this paper, we only defer the

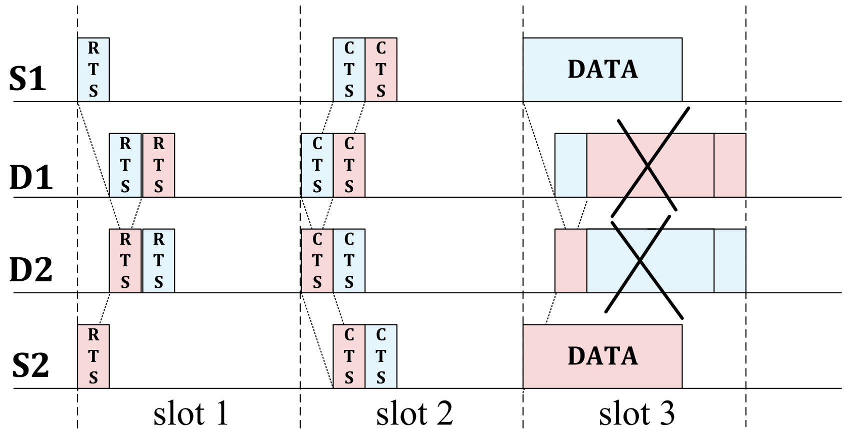


Figure 5 Data collisions caused by hidden terminal problem without CTS checking phase.

transmissions to void data collisions once a hidden terminal problem occurs.

3.5.2 RTS/CTS Collision Problem

In SFAMA-DT, it is allowed that more than one RTS or CTS packets are transmitted in the same slot. Therefore, it is possible that RTS or CTS collisions occur. Fortunately the probability of the RTS or CTS collision is low since the duration of a control packet is very short with respect to the slot. However, while the traffic of the network becomes higher, more RTS or CTS packets are sent in the same slot, thus this kind of collision may occur frequently which will limit the network throughput. Therefore, in dense networks we may restrict the max length of the data train in order to relieve the problem of RTS or CTS collisions. In our simulation, the max length of the data train is set to 4.

4 Simulation

In this section, we evaluate the performance of SFAMA-DT using simulations. By comparing with Slotted-FAMA, we demonstrate the aggregate throughput achieved by SFAMA-DT.

4.1 Simulation Settings

We implement SFAMA-DT and Slotted-FAMA in Aqua-Sim, an NS-2 based simulator for underwater sensor networks, developed at the Underwater Sensor Network (UWSN) Lab at the University of Connecticut [21].

Unless specified otherwise, we use the following parameters in the simulations. We assume that nodes generate traffic according to a Poisson process of rate λ packets per second per node. We also assume that all nodes transmit at the same rate, within the same bandwidth, and using the same transmit power. The bit rate is 800 bps. The DATA packet size is 500B and the control packet size is 50B. The power consumption on transmission mode is 2 Watts and the power consumption on receive mode is 0.75 Watts. The maximum transmission range is 1500 meters and the interference range is the same as the transmission range. The speed of sound is always 1500m/s. The simulation time is 10^3s.

The network throughput is measured as the number of successful data transmissions per unit time.

4.2 One Hop Network Scenario

In the one-hop network scenario, every node is within the transmission ranges of all the other nodes in the network. We simulate with 8 nodes and 16 nodes. The nodes are deployed in an area by 1500m$\times$1500m. Half of these nodes are source nodes and the others are destinations. Each source node generates packets according to the Possion process with a rate λ (packets/s). For each generated packet, the destination is randomly selected from the destination nodes.

Figure 6 shows the network throughput with varying packet generation rate for the two target MAC protocols, Slotted-FAMA and SFAMA-DT in one-hop networks. This figure shows that SFAMA-DT achieves a great improvement in network throughput over Slotted-FAMA both in the 8 nodes network and the 16 nodes network. When the packet generation rate is low, the situation of multiple RTS attempts almost does not happen, and the transmissions with data train also rarely occur in SFAMA-DT. But when the packet generation rate becomes higher, the multiple RTS attempts problem becomes more serious in Slotted-FAMA. As shown in Fig. 6, Slotted-FAMA suffers the degradation of throughput when the traffic becomes higher, since Slotted-FAMA does nothing but backs off when the multiple RTS attempts problem occurs and the high frequency of backoff greatly limits the throughput. On the contrary, when the traffic becomes higher in SFAMA-DT, more useful data transmissions can be successfully transmitted. This is because that nodes in SFAMA-DT complete the data transmissions via a sorted data train, utilizing the situation of multiple RTS attempts, and more data packets can be transmitted during each round of simultaneously handshakes.

By comparing the throughput of the 8 nodes network and the 16 nodes network, we can also find SFAMA-DT and Slotted-FAMA perform totally different when the networks is denser. Comparing Slotted-FAMA in an 8 node network

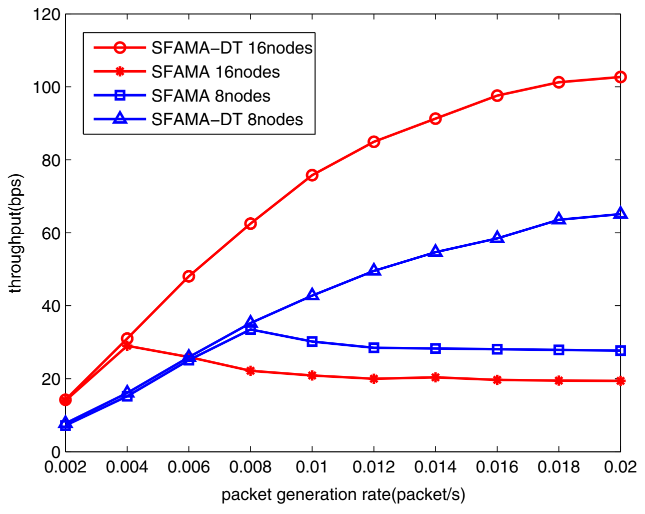


Figure 6 One-hop throughput.

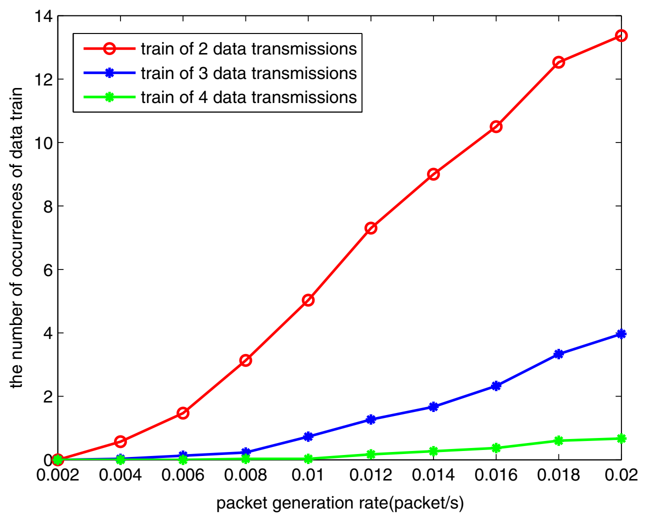


Figure 7 The number of occurrences of data train in the 8 nodes one-hop network.

with a 16 node network, the network throughput becomes lower since the multiple RTS attempts problem is more serious. This situation does not happen in SFAMA-DT, because SFAMA-DT can handle the multiple RTS attempts problem and more transmission with data train occur. As shown in Fig. 6, in a 16 nodes network, SFAMA-DT achieves a better improvement than in an 8 node network in network throughput.

Figures 7 and 8 depict the situations of data trains in the 8 nodes network and the 16 nodes network. Both two figures show that more data packets are transmitted via a data train when the traffic becomes higher, and the longer data trains will replace the shorter ones. This is essentially why SFAMA-DT achieves higher throughput.

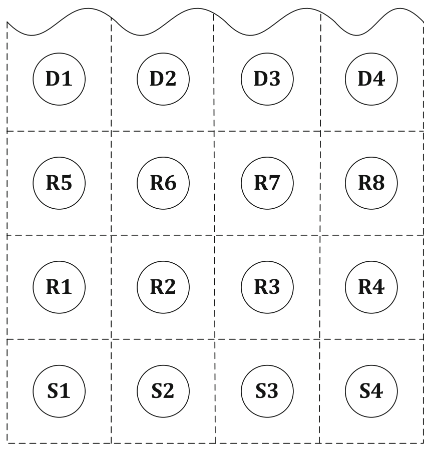


Figure 9 Multi-hop topology.

4.3 Multi-Hop Network Scenario

Since multi-hop UWSNs are generally more useful for underwater applications, it is necessary to evaluate the performance of the RC-SFAMA in a generic scenario.

As shown in Fig. 9, it is a four-level multi-sink network. Nodes S1-S4 are four data source nodes in level 1, R1-R8 are relay nodes in level 2 and 3 and D1-D3 are four destination nodes in level 4. For the node with a packet in the lower level, it randomly selects one node in the upper lever to send the packet. All the nodes are deployed in an area 3000m × 3000m. Each source node generates packets according to the Poisson process with a rate λ (packets/s).

The network throughput and the situation of data trains of the multi-hop network are shown in Figs. 10 and 11 with varying packet generation rate. As the same trend of

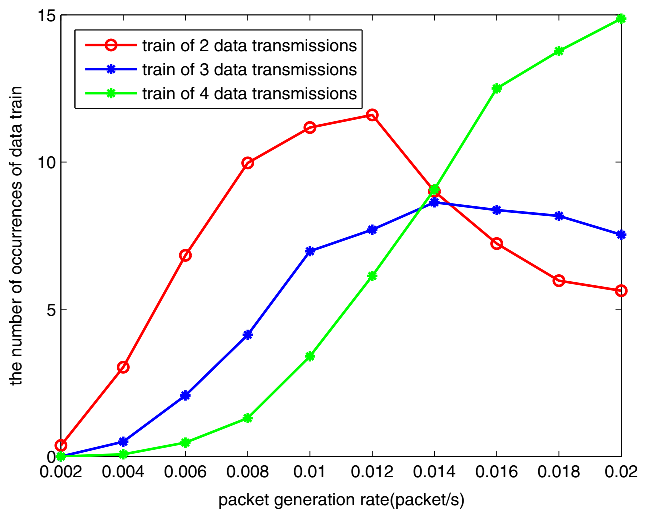


Figure 8 The number of occurrences of data train in the 16 nodes one-hop network.

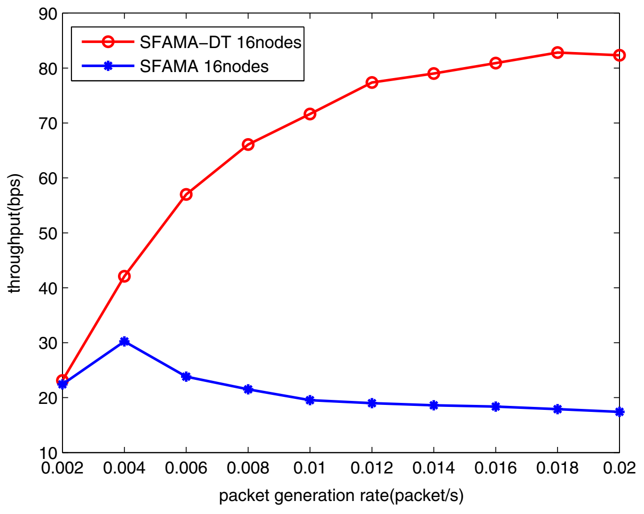


Figure 10 Multi-hop throughput.

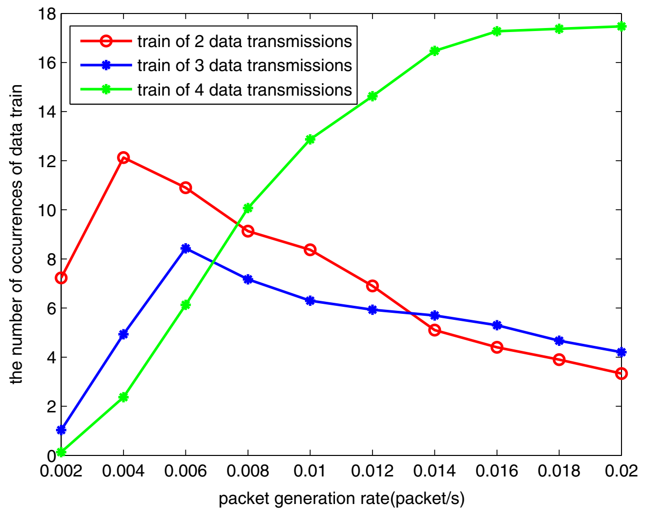


Figure 11 The number of occurrences of data train in the multi-hop network.

one-hop network, these figures show that SFAMA-DT achieves a better network throughput performance than Slotted-FAMA and more data train are formed when the traffic becomes higher.

5 Conclusions

In this paper, we first present the original Slotted-FAMA and indicate its limitation in high traffic networks. Then we propose a Slotted-FAMA based MAC protocol, SFAMA-DT, to solve the multiple RTS attempts problem and low throughput problem of Slotted-FAMA. SFAMA-DT introduces RTS/CTS sorting scheme and data train scheme to form a train of data packets of multiple transmission pairs during each round of simultaneously handshakes, which greatly improves the channel utilization. Simulation results show that the SFAMA-DT is able to achieve much higher throughput than the Slotted-FAMA protocol. Our future work for SFAMA-DT includes a more efficient way to solve the hidden terminal problem and RTS/CTS collision problem, as well as the theoretical analysis. Moreover, we also need to consider possible problems in real implementation.

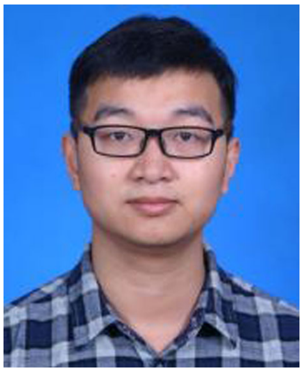
Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grants 61374021 and 61531015, the Zhejiang Provincial Natural Science Foundation of China under Grants LZ14F030003 and LY15F030007, the Specialized Research Fund for the Doctoral Program of Higher Education of China (SRFDP) under Grants 20130101110109 and 20120101110115, the ASFC under Grant 2015ZC76006, and the special electronic fund TD-LTE of national ministry of industry and information in 2014.

References

1. Akyildiz, I.F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks Journal*, *3*(3), 257–281.
2. Cui, J.-H., Kong, J.J., Gerla, M., & Zhou, S.L. (2006). The challenges of building mobile underwater wireless networks for aquatic applications. *IEEE Network*, *20*(3), 12–18.
3. Heidemann, J., Ye, W., Wills, J., Syed, A., & Li, Y. (2006). Research challenges and applications for underwater sensor networking.In *IEEE wireless communications and networking conference* (pp. 228–235).
4. Liu, J., Wang, Z., Zuba, M., Peng, Z., Cui, J.-H., & Zhou S.L (2012). JSL: Joint time synchronization and localization design with stratification compensation in mobile underwater sensor networks. In *Annual IEEE communications society conference* (pp. 317–325).
5. Liu, L.B., Zhou, S.L., & Cui, J.-H. (2008). Prospects and problems of wireless communication for underwater sensor networks. *Wireless Commun. Mobile Comput.*, *8*(8), 977–994.
6. Qiu, M.K., Ming, Z., Li, J.Y., Liu, J.N., Quan, G., & Zhu, Y. (2013). Informer homed routing fault tolerance mechanism for wireless sensor networks. *Journal of System Architecture*, *59*(4–5), 260–270.
7. Qiu, M., Ming, Z., Li, J., & Gai, K. (2015). Phase-change memory optimization for green cloud with genetic algorithm. *IEEE Transactions on Computers*, *64*(12), 3528–3540.
8. Niu, J., Gao, Y., Qiu, M.K., & Ming, Z. (2012). Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability. *Journal of Parallel and Distributed Computing*, *72*(12), 1565–1575.
9. Wu, G., Zhang, H.X., Qiu, M.K., Ming, Z., Li, J.Y., & Qin, X. (2013). A decentralized approach for mining event correlations in distributed system monitoring. *Journal of Parallel and Distributed Computing*, *73*(3), 330–340.
10. Peleato, B., & Stojanovic, M. (2007). Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks. *IEEE Communications Letters*, *11*(12), 1025–1027.
11. Yang, J., Guo, P., Jiang, T., & Zhang, K. (2012). SRCR: A novel MAC protocol for underwater acoustic networks with concurrent reservation. In *IEEE international conference on communications* (pp. 435–439).
12. Luo, Y., Pu, L., Peng, Z., Zhou, Z., & Cui, J.-H. (2012). CT-MAC: A MAC protocol for underwater mimo based network uplink communications. In *Proceedings of ACM international conference on underwater networks and systems* (pp. 1–8).
13. Sozer, E.M., Stojanovic, M., & Proakis, J.G. (2000). Underwater acoustic networks. *IEEE Journal of Oceanic Engineering*, *25*(1), 72–83.
14. Park, M.K., & Rodoplu, V. (2007). UWAN-MAC: An energy-efficient MAC protocol for underwater acoustic wireless sensor networks. *IEEE Journal of Oceanic Engineering*, *32*(3), 710–720.
15. Chirdchoo, N., Soh, W.-S., & Chua, K.C. (2007). Aloha-based MAC protocols with collision avoidance for underwater acoustic networks. In *Proceedings of IEEE INFOCOM* (pp. 2271–2275).
16. Karn, P. (1990). MACA: A new channel access method for packet radio. In *ARRL/CRRL amateur radio 9th computer networking conference* (pp. 134–140).
17. Fullmer, C.L., & Garcia-Luna-Aceves, J.J. (1995). Floor acquisition multiple access (fama) for packet-radio networks. In *Proceedings of the conference on applications, technologies, architectures, and protocols for computer communication* (pp. 262–273).

18. Molins, M., & Stojanovic, M. (2006). Slotted FAMA: A MAC protocol for underwater acoustic networks. In *IEEE OCEANS* (pp. 1–7).
19. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM* (pp. 1567–1576).
20. Chirdchoo, N., Soh, W.-S., & Chua, K.C. (2008). MACA-MN: A MACA-based MAC protocol for underwater acoustic networks with packet train for multiple neighbors. In *IEEE vehicular technology vonference* (pp. 46–50).
21. Xie, P., Zhou, Z., Peng, Z., Yan, H., Hu, T., Cui, J.-H., Shi, Z., Fei, Y., & Zhou, S.L. (2009). Aqua-sim: an NS-2 based simulator for underwater sensor networks. In *IEEE OCEANS* (pp. 1–7).

Senlin Zhang received the B.E. degree in control theory and control engineering from the Wuhan University of Technology, Wuhan, China, and the M.E. degree in control theory and control engineering from Zhejiang University, Hangzhou, China, in 1984 and 1991, respectively. He is currently a Professor with the College of Electrical Engineering, Zhejiang University, Hangzhou. His current research interests include underwater wireless sensor networks, textile automation, and intelligent systems


Liangfang Qian received the BE degree in control theory and control engineering from Zhejiang University, Hangzhou, China, in 2011. He is currently a PhD student in the College of Electrical Engineering, Zhejiang University, Hangzhou, China. His research focuses on MAC protocol for underwater acoustic netwroks.

Meiqin Liu received the B. E. and Ph.D. degrees in control theory and control engineering from Central South University, Changsha, China, in 1994 and 1999, respectively. She was a Post-Doctoral Research Fellow with the Huazhong University of Science and Technology, Wuhan, China, from 1999 to 2001. She was a Visiting Scholar with the University of New Orleans, New Orleans, LA, USA, from 2008 to 2009. She is currently a Professor with the College of Electrical Engineering, Zhejiang University, Hangzhou, China. She has authored more than 150 papers in major journals and international conferences. Her current research interests include robust control, multi-sensor networks, and information fusion.

Zhen Fan received the B.E., M.E., and Ph.D. degrees in control theory and control engineering from Zhejiang University, Hangzhou, China, in 1992, 1995, and 2005, respectively. He is currently a Lecturer with the College of Electrical Engineering, Zhejiang University, Hangzhou. His current research interests include textile automation, intelligent systems, and multi-sensor networks.

Qunfei Zhang received the B.E., M.E. degree in electronic and information engineering from the Northeastern Polytechnical University, Xi'an, China, and the Ph.D. degree in signal and information processing from Xidian University, Xi'an, China, in 1990, 1993 and 2003, respectively. He is currently a Professor with the School of Marine Science and Technology, Northeastern Polytechnical University, Xi'an. His current research interests include array signal processing, underwater acoustic communication, and underwater wireless sensor networks.