

A slotted floor acquisition multiple access based MAC protocol for underwater acoustic networks with RTS competition*

Liang-fang QIAN, Sen-lin ZHANG, Mei-qin LIU[‡]

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)

E-mail: qian9904@zju.edu.cn; slzhang@zju.edu.cn; liumeiqin@zju.edu.cn

Received May 16, 2014; Revision accepted Nov. 5, 2014; Crosschecked Jan. 28, 2015

Abstract: Long propagation delay, limited bandwidth, and high bit error rate pose great challenges in media access control (MAC) protocol design for underwater acoustic networks. A MAC protocol called slotted floor acquisition multiple access (slotted-FAMA) suitable for underwater acoustic networks is proposed and analyzed. This FAMA based protocol adds a time slot mechanism to avoid DATA packet collisions. However, slotted-FAMA is not suitable for dense networks since the multiple request-to-send (RTS) attempts problem in dense networks is serious and greatly limits the network throughput. To overcome this drawback, this paper proposes a slotted-FAMA based MAC protocol for underwater acoustic networks, called RC-SFAMA. RC-SFAMA introduces an RTS competition mechanism to keep the network from high frequency of backoff caused by the multiple RTS attempts problem. Via the RTS competition mechanism, useful data transmission can be completed successfully when the situation of multiple RTS attempts occurs. Simulation results show that RC-SFAMA increases the network throughput efficiency as compared with slotted-FAMA, and minimizes the energy consumption.

Key words: Underwater acoustic networks, Medium access control (MAC), Request-to-send (RTS) competition, Throughput, Energy consumption

doi:10.1631/FITEE.1400187

Document code: A

CLC number: TP393.0

1 Introduction


Recent years have seen a rapidly growing interest in underwater acoustic networks (UANs). One important reason is that they can be used for a broad

range of scientific exploration, including environmental monitoring, disaster prevention, assisted navigation, and mine reconnaissance (Cui *et al.*, 2006; Liu *et al.*, 2012).

A medium access control (MAC) protocol allows the nodes in a network to share a common broadcast channel. While wireless MAC protocols have been studied extensively for a long time, we cannot directly adapt these protocols to UANs due to the significant difference between UANs and terrestrial radio networks. The underwater acoustic channels have the characteristics of long propagation delay, low bit rates, and error-prone acoustic communication (Akyildiz *et al.*, 2005; Partan *et al.*, 2007; Liu *et al.*, 2008). The propagation speed of

[‡] Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61374021, 61222310, and 61328302), the Zhejiang Provincial Natural Science Foundation of China (No. LZ14F030003), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Nos. 20130101110109 and 20120101110115), the Zhejiang Provincial Science and Technology Planning Projects of China (No. 2012C21044), the Marine Interdisciplinary Research Guiding Funds for Zhejiang University (No. 2012HY009B), and the Fundamental Research Funds for the Central Universities, China (No. 2014XZZX003-12)

 ORCID: Liang-fang QIAN, <http://orcid.org/0000-0002-6692-931X>

©Zhejiang University and Springer-Verlag Berlin Heidelberg 2015

acoustic signals in water is about 1500 m/s. Propagation delay underwater is five orders of magnitude higher than in radio frequency (RF) terrestrial channels. The available bandwidth of acoustic channels is typically less than 15 kHz, which is much narrower compared with that of RF channels (Xie and Cui, 2007). Moreover, underwater acoustic channels are error-prone due to path-loss, noise, multi-path, and Doppler spread. These unique acoustic communication characteristics pose challenges to underwater MAC protocol design.

Because of the unique characteristics in the underwater environment, most of the existing terrestrial MAC protocols are not suitable. Many MAC protocols dedicated for UANs have been proposed in the last decade. Peleato and Stojanovic (2007) proposed a protocol called DACAP, which allows a sender to use different handshake lengths for different receivers to improve the channel utilization. In Yang *et al.* (2012), a novel handshaking-based MAC protocol named SRCR was proposed for multi-hop UANs. 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.* (2012) proposed CT-MAC, in which a special relay mechanism was leveraged. 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 (Abramson, 1970). ALOHA is based on random access of users to the medium and it allows nodes to grasp the channel whenever they have data to send. An improvement to the original ALOHA protocol is slotted-ALOHA (Roberts, 1975). In this method, the time is divided into time slots, and a node can only start transmission at the beginning of a slot. 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 bursty traffic that occurs in information networks. Also, trying to resolve collisions by retransmissions increases the power consumption of the network nodes and reduces the lifetime of the network (Sozer *et al.*, 2000).

Unlike ALOHA that allows nodes to transmit at

will, protocols that make nodes listen to the channel before they access it are called carrier sense multiple access (CSMA) protocols (Tanenbaum, 2003). These protocols require the stations to listen to the channel before starting to transmit to avoid possible collisions with other ongoing transmissions. However, since the transmission range of a node in a wireless network is limited, CSMA is subject to the hidden terminal problem, which means that carrier sensing cannot indicate the real status of the channel if competing nodes are out of the transmission range of each other (Guo *et al.*, 2009).

To solve the problem of CSMA, the MACA protocol was proposed by Karn (1990). This protocol 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 the CSMA protocol may occur due to different packet delays. To overcome this problem, the floor acquisition multiple access (FAMA) protocol was proposed. In FAMA, collision avoidance is guaranteed by holding the following two necessary conditions: (1) the duration of the RTS packet must be longer than t_{\max} , which is the propagation delay required to reach the maximum coverage radius of a node, and (2) the duration of the CTS packet must be longer than the duration of RTS plus $2t_{\max}$. However, for underwater communications, these two conditions pose a number of problems (Casari *et al.*, 2008). The long propagation delays in underwater scenarios require very long transmission of control packets, which would lead to a dramatic loss of efficiency and a useless increase in power consumption. Moreover, transmit power is significantly higher than receive or idle power in typical underwater modem hardware. This fact discourages long transmission times, severely limiting the use of the original version of FAMA in underwater networks.

Slotted-FAMA, which was proposed by Molins and Stojanovic (2006) to solve this problem, makes

nodes share a global time synchronization and divides time into slots. In slotted-FAMA, each packet (RTS, CTS, DATA, or ACK) has to be transmitted at the beginning of one slot. The slot length is determined to ensure the absence of data packet collisions. 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. If no ACK is received, the node re-transmits the whole data packet. Any neighboring node which overhears an RTS packet or a CTS packet defers its transmission to allow ongoing communications to be completed correctly.

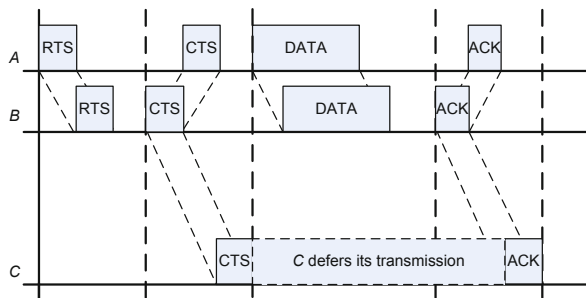


Fig. 1 A successful handshake between terminals *A* and *B* in slotted-FAMA

Although slotted-FAMA avoids DATA packet collisions, it is not suitable for dense networks since the multiple RTS attempts problem in dense networks is much more serious. To overcome this drawback, this paper proposes a slotted-FAMA based MAC protocol called RC-SFAMA. RC-SFAMA introduces an RTS competition mechanism to keep the network from the endless multiple RTS attempts problem.

2 RC-SFAMA protocol

In this section, we first discuss the multiple RTS attempts problem of slotted-FAMA in dense networks and then describe RC-SFAMA in detail. Then we focus on the RTS competition and discuss some

issues. Finally, we present the analysis of network throughput.

2.1 Multiple RTS attempts problem

As mentioned above, slotted-FAMA will have a problem of frequent multiple RTS attempts in dense networks.

In a dense network, the probability of two or more nodes sending RTS at the beginning of the same slot is high. As shown in Fig. 2, nodes *A* and *D* both send RTS to their destination nodes at the beginning of slot 1. According to the transmission rules in slotted-FAMA, neither control packet nor data packet can be transmitted in the following two slots, since (1) nodes *A* and *D* go to the backoff state for a random number of slots and (2) nodes *B*, *C*, and *E* have to wait two slots for the receiver to send a CTS and the sender to start transmitting data, which will not in fact happen. So, in the following slot 2 and slot 3, we can do nothing but wait.

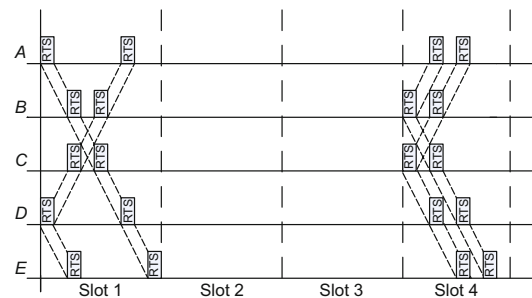


Fig. 2 The multiple RTS attempts problem

Moreover, due to the long propagation delay in UANs, a slot length, which equals 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 send. In this case, nodes *B* and *C* have a packet ready to transmit during slots 2 and 3, and thus send RTS at the beginning of slot 4. Then the same situation as slots 2 and 3 appears for slots 5 and 6.

It is true that the probability of nodes *B* and *C* both generating a new packet during slots 2 and 3 is low in this case. But when in a dense network, where there are dozens or even hundreds of nodes, the probability of two of these nodes transmitting RTS at the same slot becomes much higher. Moreover, when the situation of multiple RTS attempts occurs, the

number of the nodes that need to transmit packets increases by two at least, and during the following two slots the number may continue to increase because of the new generating packets, and no useful transmission can be completed. The more nodes that need to transmit packets, the higher the probability of multiple RTS attempts will be. Therefore, the multiple RTS attempts problem can be more and more serious as time goes on.

To overcome this problem, we adapt the original slotted-FAMA for use in dense underwater networks by adding an RTS competition mechanism, as well as investigate its throughput and energy consumption performance. We will call this improved protocol RC-SFAMA.

2.2 RC-SFAMA description

The Markov chain that models the behavior of RC-SFAMA is depicted in Fig. 3. Let us start its description from the idle state. If a node generates one or more packets while in this state, it sends the RTS at the beginning of the next slot and moves to the TX RTS state. However, during this slot, this node may receive one or more xRTS packets (RTS packets sent by other nodes). Then we need to consider two cases: (a) receive no xRTS packet, and (b) receive one or more xRTS packets.

Case (a): If a node receives no xRTS packet in the TX RTS state, it goes to the state of WFCTS (wait for CTS). Then if it receives the CTS from the addressed node, it goes to the TX data state and sends the data packets. The node stays in TX data, until it eventually receives the ACK (if no ACK is received within the scheduled time which is long

enough to complete ACK transmission, the node re-transmits the whole data packet), and then returns to the idle state. If no CTS packet is received, the node backs off and reschedules a later attempt.

Case (b): If a node receives one or more xRTS packets in the TX RTS state, it goes to the RTS-C (RTS competition) state. In this state, the RTS packet of this node needs to compete with the xRTS packets. If the RTS packet wins, this node goes to the WFCTS state and goes through as case (a). If the xRTS wins, this node goes to the state of back-off 2. Then it must wait two slots (long enough for the receiver to send a CTS and the sender to start transmitting data). When in this state, if it receives the xCTS responding to the xRTS or the xData sent from the xnode (the node which sends the xRTS packet) during this time, it goes to the WFXACK state, which will be discussed later. After this time, if no carrier is sensed, the node goes to the TX RTS state and prepares to re-transmit an RTS packet.

If the node, while in the idle state, receives an RTS meant for it or an xRTS intended for other nodes, it goes to the RTS-C state. If there is only one RTS or xRTS in this state, the RTS competition is skipped and the single RTS or xRTS wins. If more than one RTS or xRTS is received, the node conducts an RTS competition. When the xRTS wins, the node goes to the state of backoff 2 and goes through as we discussed above. When the RTS wins, the node goes to the TX CTS state and prepares to send a CTS packet at the beginning of the next slot. After the CTS packet has been sent, the node goes to the RX data state, where it stays until a correct data packet is received. At this point, the node transmits an

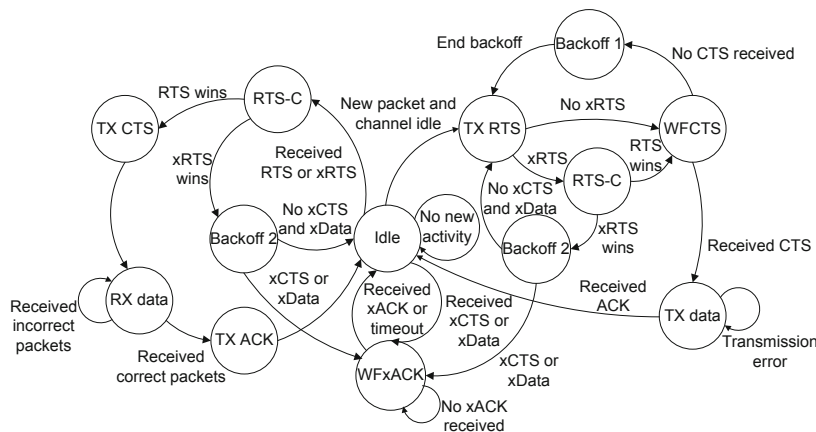


Fig. 3 State-transition diagram for RC-SFAMA

ACK (TX ACK state) and then goes back to the idle state.

If the node, while in the idle state, overhears an xCTS or xData packet addressed to another node, it goes to the WfACK (wait for xACK) state and waits for the xACK to be sent. But the waiting time is different according to the received packet. If an xCTS packet is received, the node needs to wait long enough to allow the other node to transmit the entire data packet and receive the corresponding ACK. Since this node has received the xCTS packet, it will also receive the xACK packet and will thus know that data transmission has ended successfully. If an xData packet is received, the node needs to wait long enough to allow the reception of the subsequent xACK packet. Since it is possible that the terminal cannot hear the xACK packet, it must wait an additional slot to detect whether the data packet has been re-sent or not. After hearing an xACK packet, the node has to wait until the end of the slot since the data transmission has successfully ended.

2.3 RTS competition

In the original slotted-FAMA protocol, when a node, which has already sent an RTS and is in the TX RTS state, receives an RTS packet, it defers its transmission and backs off for random slots immediately. The node which receives more than one RTS will ignore the RTS intended for it since the multiple RTS attempts problem occurs, and will wait for two slots. In RC-SFAMA, we add an RTS competition mechanism to both sending and receiving links.

When a node in the idle state has a new packet to send, it makes an RTS packet and adds a competing number, called C-number, into the RTS packet. In RC-SFAMA, we obtain C-numbers by random number generation. Then the node sends the RTS packet with the C-number at the beginning of the following slot. In this way, each RTS packet will have a C-number with it. Then we can conduct RTS competitions. For a sender, when it receives one or more RTS packets in the TX RTS state, it makes its own C-number compete with the C-numbers of other RTS packets one by one. If the sender wins, which means that the C-number of the sender is the largest one, it ignores other RTS packets and continues completing its transmission. Otherwise, the sender defers its transmission and backs off. For a receiver, when it receives more than one RTS packet

and one of them is intended for itself, it also makes the C-number in the RTS packet intended for it compete with the C-numbers of other RTS packets. If the receiver wins, it ignores the other RTS packets and sends a CTS packet at the beginning of the following slot. Otherwise, the receiver goes to the waiting state. For the nodes neither a sender nor a receiver, when receiving more than one RTS packet, they need not conduct an RTS competition and go through as receiving one RTS packet intended for another node. The procedure of the receiving RTS process is shown in Algorithm 1.

By adding an RTS competition mechanism, the transmission of the winning node can keep going.

2.4 RTS collision

As mentioned above, both slotted-FAMA and RS-SFAMA allow more than one RTS to be transmitted in the same slot. In this case, the RTS packets could arrive at a node at the same time or have an overlapping period at the receiving node, which will

Algorithm 1 Receiving RTS

```

1: if the STATE is WFCTS then
2:   if the RTS (one or more) is for this node then
3:     Compete C-numbers
4:     if this node wins then
5:       Continue to WFCTS
6:     else
7:       Wait to send CTS
8:     end if
9:   else
10:    Compete C-numbers
11:    if this node wins then
12:      Continue to WFCTS
13:    else
14:      BACKOFF
15:    end if
16:  end if
17: else
18:   if the RTS (one or more) is for this node then
19:     Compete C-numbers
20:     if this node wins then
21:       Wait to send CTS
22:     else
23:       BACKOFF
24:     end if
25:   else
26:     BACKOFF
27:   end if
28: end if

```

create collisions, i.e., RTS collisions. Although the probability of RTS collision is low since the duration of a control packet is very short with respect to the slot, it cannot be ignored when the traffic of the network increases.

Fortunately, the RTS collision problem will not cause much trouble in RC-SFAMA. When an RTS collision occurs in RC-SFAMA, the receiving node may not decode the packets correctly and does not know which packets have collided. Then the worst assumption is made, and the node acts as if it had received an xCTS packet and it defers its transmission. If this receiving node is the winning node, since it defers its transmission, no data transmission is attempted. Otherwise, the RTS collision would not impact the winning node's transmission. However, when an RTS collision occurs in slotted-FAMA, all the nodes defer their transmissions. Therefore, although the RTS collision problem is still not solved completely in RC-SFAMA, it is better than in slotted-FAMA.

2.5 Time synchronization

Time synchronization is an important requirement for RC-SFAMA. Although our work does not focus on this issue, there are various time synchronization algorithms already proposed for UANs. Liu *et al.* (2013) presented Mobi-Sync, a time synchronization scheme for UANs. Mobi-Sync is the first time synchronization algorithm to utilize the spatial correlation characteristics of underwater objects, improving the synchronization accuracy as well as the energy efficiency. Another time synchronization algorithm called E²DTS was proposed by Li *et al.* (2013). The authors investigated the relationship between time-varying propagation delay and nodes mobility, and then estimated the clock skew. Skew-corrected nodes send local time stamp to the beacon node to estimate its clock offset. The E²DTS can achieve high level time synchronization precision with minimal energy cost. Both time synchronization algorithms are accurate and energy-efficient, and can be introduced to RC-SFAMA to achieve the requirement of time synchronization.

2.6 Throughput analysis

Similar to slotted-FAMA (Molins and Stojanovic, 2006), we assume node A has N neighbors.

Each of them has Q neighbors which are hidden from node A . Each node generates packets according to a Poisson process of a rate of λ packets per second per node. The packets are distributed evenly among the neighbors, i.e., λ/N directed to each of the neighbor nodes.

Throughput per node (S) can be defined as

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}}, \quad (1)$$

where \bar{U} is the average useful data transmission time, \bar{B} is the average busy time while the channel is being used, and \bar{I} is the average idle time.

Define $P_{s(\text{SFAMA})}$ as the probability of successful data transmission (no collisions) of slotted-FAMA and $P_{s(\text{RC-SFAMA})}$ of RC-SFAMA. In slotted-FAMA, the probability of no collisions is the probability that only the given node A sends an RTS and no neighbors send CTS whose corresponding RTS has not been heard. For each A 's neighbor, the number of neighbors hidden from it equals Q . Then the probability of successful data transmission of slotted-FAMA is

$$P_{s(\text{SFAMA})} = \prod_1^N e^{-\lambda T_{\text{slot}}} \cdot \prod_1^N \prod_1^Q e^{-\frac{\lambda}{N} T_{\text{slot}}} = e^{-\lambda(N+Q)T_{\text{slot}}}. \quad (2)$$

In RC-SFAMA, the probability of no collisions is the probability that the given node A sends an RTS and wins the RTS competition if the neighbors of A also send RTS in this slot, and no neighbors send CTS whose corresponding RTS has not been heard. The probability of successful data transmission of RC-SFAMA is

$$P_{s(\text{RC-SFAMA})} = \sum_{i=0}^N \left(\prod_1^{N-i} e^{-\lambda T_{\text{slot}}} \cdot (1 - e^{-\lambda T_{\text{slot}}})^i \frac{1}{i+1} \right) \prod_1^N \prod_1^Q e^{-\frac{\lambda}{N} T_{\text{slot}}} = e^{-\lambda(N+Q)T_{\text{slot}}} + \sum_{i=1}^N \left(e^{-\lambda(N+Q-i)T_{\text{slot}}} \cdot (1 - e^{-\lambda T_{\text{slot}}})^i \frac{1}{i+1} \right). \quad (3)$$

Then

$$\begin{aligned} \Delta P_s &= P_{s(\text{RC-SFAMA})} - P_{s(\text{SFAMA})} \\ &= \sum_{i=1}^N \left(e^{-\lambda(N+Q-i)T_{\text{slot}}} (1 - e^{-\lambda T_{\text{slot}}})^i \frac{1}{i+1} \right) > 0. \end{aligned} \quad (4)$$

Therefore, the probability of successful data transmission of RC-SFAMA is higher than that of slotted-FAMA in the same network.

Given the bit error rate (BER), the probability of error in a data packet containing L bits, assuming independent error, is

$$P_e = 1 - (1 - \text{BER})^L \approx L \cdot \text{BER}. \quad (5)$$

The average busy time \bar{B} can be defined as

$$\bar{B} = \bar{T}_{\text{success}} + \bar{T}_{\text{fail}} + \bar{T}_{\text{defer}}, \quad (6)$$

where \bar{T}_{success} is the time during which data is being successfully sent, \bar{T}_{fail} is a period of collisions on the channel, and \bar{T}_{defer} is the time during which we cannot transmit due to transmissions from other nodes.

The duration of a successful transmission consists of the RTS, CTS, DATA, and ACK. RTS, CTS, and ACK need only one slot and DATA needs more slots, defined as T_{data} . Define T as the time between the start of DATA and the end of ACK. Then

$$T = \sum_{n=1}^{\infty} n(T_{\text{data}} + T_{\text{slot}})P_e^{n-1}(1 - P_e) = \frac{T_{\text{data}} + T_{\text{slot}}}{1 - P_e}. \quad (7)$$

Then the duration of a successful transmission is

$$\begin{aligned} T_{\text{Tot}} &= 2T_{\text{slot}} + T \\ &= 2T_{\text{slot}} + (T_{\text{data}} + T_{\text{slot}})/(1 - P_e). \end{aligned}$$

Hence,

$$\bar{T}_{\text{success}} = P_s T_{\text{Tot}}. \quad (8)$$

The duration of a failed transmission is two slots (the slot in which the RTS packet is sent, and the next slot spent in waiting for the CTS that will not arrive). The probability that a given RTS was transmitted by A is $1/(N + 1)$ because all the $N + 1$ nodes transmit at the same rate. So,

$$\bar{T}_{\text{fail}} = \frac{2T_{\text{slot}}(1 - P_s)}{N + 1}. \quad (9)$$

A node defers its transmission when it overhears a CTS sent by one of its neighbors or there is a collision on the channel.

The probability of overhearing a CTS is

$$\text{Prob}(\text{CTSoverheard}) = \frac{Q \frac{\lambda}{N} N P_s}{(N + 1)\lambda} = \frac{Q}{N + 1} P_s. \quad (10)$$

In this situation, the deferral time equals $(T_{\text{data}} + T_{\text{slot}})/(1 - P_e)$.

The probability of hearing a collision on the channel is

$$\text{Prob}(\text{collision}) = \frac{N}{N + 1}(1 - P_s). \quad (11)$$

In this situation, the deferral time equals $T_{\text{data}} + T_{\text{slot}}$. So, the average deferral time is

$$\begin{aligned} \bar{T}_{\text{defer}} &= (T_{\text{data}} + T_{\text{slot}}) \left(\frac{Q P_s}{(N + 1)(1 - P_e)} \right. \\ &\quad \left. + \frac{N}{N + 1}(1 - P_s) \right). \end{aligned} \quad (12)$$

The average idle time on the channel is

$$\bar{I} = \frac{1}{\lambda(N + 1)}. \quad (13)$$

The average useful data transmission time is

$$\bar{U} = \frac{\delta}{N + 1} P_s, \quad (14)$$

where δ is the transmission time of the DATA packet.

Linking all the parts, the final result of throughput is

$$S = \frac{\delta P_s}{\bar{C} P_s + \bar{D}}, \quad (15)$$

where

$$\begin{aligned} \bar{C} &= (N + 1)T_{\text{Tot}} - 2T_{\text{slot}} + \frac{T_{\text{data}} + T_{\text{slot}}}{1 - P_e} Q \\ &\quad - N(T_{\text{data}} + T_{\text{slot}}), \end{aligned} \quad (16)$$

$$\bar{D} = 2T_{\text{slot}} + N(T_{\text{data}} + T_{\text{slot}}) + \frac{1}{\lambda}. \quad (17)$$

Then we can obtain the derivative of S with respect to P_s :

$$\frac{dS}{dP_s} = \frac{\delta \bar{D}}{(\bar{C} P_s + \bar{D})^2} > 0. \quad (18)$$

Since $\Delta P_s = P_{s(\text{RC-SFAMA})} - P_{s(\text{SFAMA})} > 0$, RC-SFAMA achieves higher throughput than slotted-FAMA in the same network.

3 Simulation results

In this section, we evaluate the performance of RC-SFAMA using simulations. By comparing with slotted-FAMA, we demonstrate the aggregate throughput achieved by RC-SFAMA. We also conducted experiments to explore the energy efficiency of RC-SFAMA.

3.1 Simulation settings

We implemented RC-SFAMA 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 (Xie *et al.*, 2009).

Unless specified otherwise, we used the following parameters in the simulations. We assumed that nodes generate traffic according to a Poisson process of a rate of λ packets per second per node. We also assumed that all nodes transmit at the same rate, within the same bandwidth, and using the same transmit power. The bit rate was 800 bits/s. The maximum transmission range was 3000 m. The interference range was the same as the transmission range. In our simulations, we set the energy consumption parameters based on a commercial underwater acoustic modem, UMW1000, from LinkQuest: the power consumption on transmission mode was 2 W and the power consumption on receive mode was 0.75 W. The simulation time was 1000 s.

We used network throughput and energy consumption as performance metrics. The network throughput was measured as the number of successful data transmissions per unit time. The energy consumption was measured by the average energy consumed for each successfully delivered packet.

3.2 One-hop network scenario

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

Fig. 4 depicts the network throughput with a varying packet generation rate for the two target MAC protocols, slotted-FAMA and RC-SFAMA. This figure shows that RC-SFAMA achieves a great improvement in network throughput over slotted-FAMA when both of them reach the maximum throughput in an 8-node network. As shown in Figs. 4 and 5, when the packet generation rate is low, the multiple RTS attempts problem rarely occurs

and would not impact much on data transmission. However, when the packet generation rate becomes higher, the multiple RTS attempts problem will be more serious. As slotted-FAMA does nothing but backs off when the multiple RTS attempts problem occurs, the nodes need to defer their data transmissions. When more multiple RTS attempts occur, the frequency of backoff becomes high, which limits the network throughput. In RC-SFAMA, though the multiple RTS attempts problem is even more serious because of the reduced backoff, it does not limit the network throughput. This is because the winning node continues completing its data transmission. Thus, the network does not waste time and more useful data transmission can be completed. That is why RC-SFAMA achieves higher throughput than slotted-FAMA.

When 16 nodes are deployed in the network, the network is denser and the performances of RC-SFAMA and slotted-FAMA are very different. Comparing slotted-FAMA in an 8-node network with that in a 16-node network, the network throughput becomes lower since the multiple RTS attempts problem is more serious. This situation does not occur in RC-SFAMA, because RC-SFAMA can handle the multiple RTS attempts problem since data

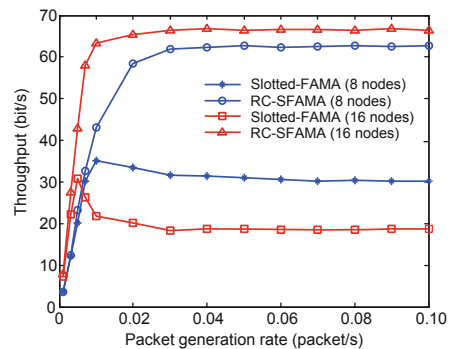


Fig. 4 One-hop throughput

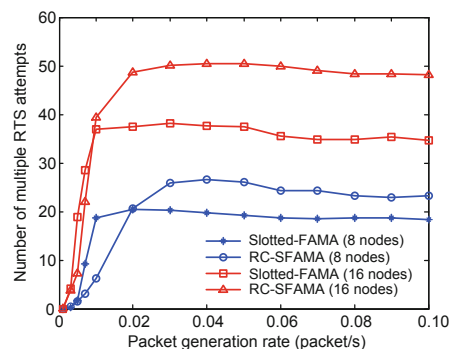


Fig. 5 One-hop multiple RTS attempts

transmission can be completed when it occurs. As shown in Fig. 4, in a 16-node network, RC-SFAMA achieves a better improvement in network throughput than slotted-FAMA in an 8-node network.

3.3 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.

Fig. 6 shows a four-level multi-sink network. Nodes S_1-S_6 are six data source nodes at level 1, R_1-R_{12} are relay nodes at levels 2 and 3, and D_1-D_6 are six destination nodes at level 4. For the node with a packet at the lower level, it randomly selects one node in the upper lever to send the packet. All the nodes are deployed in a 1500 m×1500 m area. Each source node generates packets according to the Poisson process with a rate λ (packet/s).

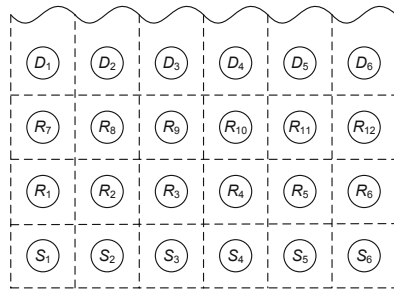


Fig. 6 Multi-hop topology

The network throughput with multiple RTS attempts is shown in Figs. 7 and 8 with a varying packet generation rate. Similar to the one-hop network, these figures show that RC-SFAMA achieves a better network throughput performance

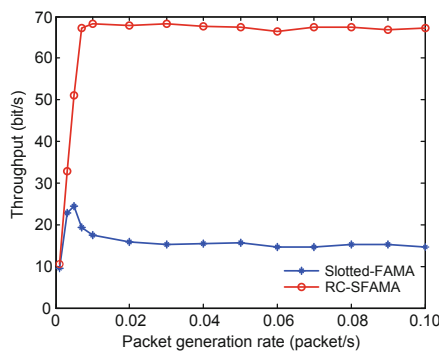


Fig. 7 Multi-hop throughput

than slotted-FAMA when the multiple RTS attempts problem is serious.

As discussed in Section 2.6, the network throughput of RC-SFAMA should be higher than that of slotted-FAMA. The simulation results also show that RC-SFAMA performs much better in network throughput than slotted-FAMA, which matches the theoretical analysis. Thus, we conclude that RC-SFAMA increases the network throughput efficiency more than slotted-FAMA. Also, the denser the network, the better the improvement.

3.4 Energy consumption

Energy efficiency is not the major target of RC-SFAMA. However, by adding RTS competition, RC-SFAMA benefits not only the improvement of network throughput but also energy conservation.

As shown in Fig. 9, in slotted-FAMA, the energy consumption per useful data transmission significantly increases when the network becomes denser, which is caused by the frequent occurrences of multiple RTS attempts, since a successful data transmission needs more RTS attempts, which wastes a lot of energy. While in RC-SFAMA, though the multiple RTS attempts also occur, the useful data

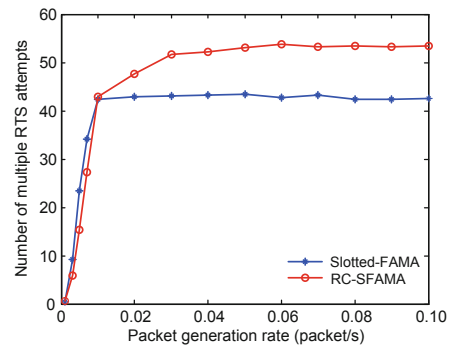


Fig. 8 Multi-hop multiple RTS attempts

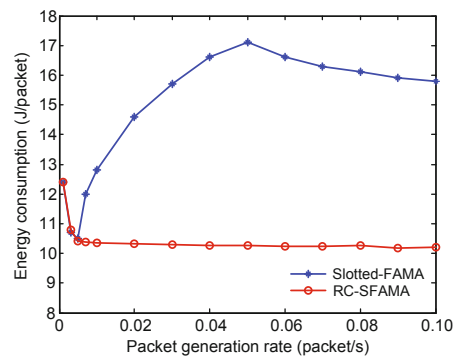


Fig. 9 Energy consumption comparison

transmission would not be limited. Within the same period of time, RC-SFAMA can complete more data transmissions than slotted-FAMA, without wasting time to re-send RTS over and over. Thus, RC-SFAMA can be more energy efficient.

4 Conclusions

In this paper, we first discuss the multiple RTS attempts problem of slotted-FAMA in dense networks and show how this problem limits network throughput. Then we propose a slotted-FAMA based MAC protocol, RC-SFAMA, to solve this problem. RC-SFAMA introduces an RTS competition mechanism to choose a winning node when the multiple RTS attempts occur, and then makes the winning node continue completing its data transmission. Simulation results show that RC-SFAMA increases the throughput efficiency compared with slotted-FAMA, and minimizes energy consumption.

Our future work for RC-SFAMA includes an investigation of unfairness in the RTS competition algorithm, as well as the comparison of theoretical analysis and simulations. Moreover, we need to consider possible problems in real implementation. For example, the maximum transmission range of each node may be different and vary in different regions, which will lead to problems to RC-SFAMA (since node *A* may be hidden from node *B* while node *B* can reach node *A*). Thus, real-world scenarios need further consideration.

References

- Abramson, N., 1970. THE ALOHA SYSTEM: another alternative for computer communications. *Proc. AFIPS*, p.281-285. [doi:10.1145/1478462.1478502]
- Akyildiz, I.F., Pompili, D., Melodia, T., 2005. Underwater acoustic sensor networks: research challenges. *Ad Hoc Netw.*, **3**(3):257-279. [doi:10.1016/j.adhoc.2005.01.004]
- Casari, P., Tomasi, B., Zorzi, M., 2008. A comparison between the Tone-Lohi and slotted FAMA MAC protocols for underwater networks. *Proc. OCEANS*, p.1-8. [doi:10.1109/OCEANS.2008.5152038]
- Cui, J.H., Kong, J., Gerla, M., et al., 2006. The challenges of building mobile underwater wireless networks for aquatic applications. *IEEE Network*, **20**(3):12-18. [doi:10.1109/MNET.2006.1637927]
- Guo, X., Frater, M.R., Rya, M.J., 2009. Design of a propagation-delay-tolerant MAC protocol for underwater acoustic sensor networks. *IEEE J. Ocean. Eng.*, **34**(2):170-180. [doi:10.1109/JOE.2009.2015164]
- Karn, P., 1990. MACA—a new channel access method for packet radio. *Proc. ARRL/CRRL Amateur Radio 9th Computer Networking Conf.*, p.134-140.
- Li, Z.B., Guo, Z.W., Hong, F., et al., 2013. E²DTS: an energy efficiency distributed time synchronization algorithm for underwater acoustic mobile sensor networks. *Ad Hoc Netw.*, **11**(4):1372-1380. [doi:10.1016/j.adhoc.2011.03.008]
- Liu, J., Wang, Z., Zuba, M., et al., 2012. JSL: joint time synchronization and localization design with stratification compensation in mobile underwater sensor networks. *Proc. 9th Annual IEEE Communications Society Conf. on Sensor, Mesh Ad Hoc Communications and Networks*, p.317-325. [doi:10.1109/SECON.2012.6275793]
- Liu, J., Zhou, Z., Peng, Z., et al., 2013. Mobi-Sync: efficient time synchronization for mobile underwater sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, **24**(2):406-416. [doi:10.1109/TPDS.2012.71]
- Liu, L., Zhou, S., Cui, J.H., 2008. Prospects and problems of wireless communication for underwater sensor networks. *Wirel. Commun. Mob. Comput.*, **8**(8):977-994. [doi:10.1002/wcm.654]
- Luo, Y., Pu, L., Peng, Z., et al., 2012. CT-MAC: a MAC protocol for underwater MIMO based network uplink communications. *Proc. 7th ACM Int. Conf. on Underwater Networks and Systems*, p.23.1-23.8. [doi:10.1145/2398936.2398966]
- Molins, M., Stojanovic, M., 2006. Slotted FAMA: a MAC protocol for underwater acoustic networks. *Proc. OCEANS*, p.1-7. [doi:10.1109/OCEANSAP.2006.4393832]
- Partan, J., Kurose, J., Levin, B.N., 2007. A survey of practical issues in underwater networks. *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, **11**(4):23-33. [doi:10.1145/1347364.1347372]
- Peleato, B., Stojanovic, M., 2007. Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks. *IEEE Commun. Lett.*, **11**(12):1025-1027. [doi:10.1109/LCOMM.2007.071160]
- Roberts, L.G., 1975. ALOHA packet system with and without slots and capture. *ACM SIGCOMM Comput. Commun. Rev.*, **5**(2):28-42. [doi:10.1145/1024916.1024920]
- Sozer, E.M., Stojanovic, M., Proakis, J.G., 2000. Underwater acoustic networks. *IEEE J. Ocean. Eng.*, **25**(1):72-83. [doi:10.1109/48.820738]
- Tanenbaum, A.S., 2003. *Computer Networks* (4th Ed.). Prentice Hall, New Jersey.
- Xie, P., Cui, J.H., 2007. R-MAC: an energy-efficient MAC protocol for underwater sensor networks. *Proc. Int. Conf. on Wireless Algorithms, Systems and Applications*, p.187-198. [doi:10.1109/WASA.2007.126]
- Xie, P., Zhou, Z., Peng, Z., et al., 2009. Aqua-Sim: an NS-2 based simulator for underwater sensor networks. *Proc. OCEANS*, p.1-7.
- Yang, J., Guo, P., Jiang, T., et al., 2012. SRCR: a novel MAC protocol for underwater acoustic networks with concurrent reservation. *Proc. IEEE Int. Conf. on Communications*, p.435-439. [doi:10.1109/ICC.2012.6363852]