

Article

A Collision-Free Hybrid MAC Protocol Based on Pipeline Parallel Transmission for Distributed Multi-Channel Underwater Acoustic Networks

Jun Zhang ^{1,2,*}, Zhi Hu ¹, Yan Xiong ³ and Gengxin Ning ¹ 

¹ School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China; zhihu2huzhi@yeah.net (Z.H.); ninggx@scut.edu.cn (G.N.)

² Key Laboratory of Marine Environmental Survey Technology and Application, Ministry of Natural Resources, Guangzhou 510300, China

³ Department of Computer Science, Guangdong University of Education, Guangzhou 510303, China; xiongy@gdei.edu.cn

* Correspondence: eejzhang@scut.edu.cn; Tel.: +86-189-9838-5259

Received: 16 March 2020; Accepted: 19 April 2020; Published: 22 April 2020



Abstract: The transmission rate between two nodes is usually very low in underwater acoustic networks due to the low available bandwidth of underwater acoustic channels. Therefore, increasing the transmission parallelism among network nodes is one of the most effective ways to improve the performance of underwater acoustic networks. In this paper, we propose a new collision-free hybrid medium access control (MAC) protocol for distributed multi-channel underwater acoustic networks. In the proposed protocol, handshaking and data transmission are implemented as a pipeline on multiple acoustic channels. Handshaking is implemented using the time division multiple access (TDMA) technique in a dedicated control channel, which can support multiple successful handshakes in a transmission cycle and avoid collision in the cost of additional delay. Data packets are transmitted in one or multiple data channels, where an algorithm for optimizing the transmission schedule according to the inter-nodal propagation delays is proposed to achieve collision-free parallel data transmission. Replication computation technique, which is usually used in parallel computation to reduce the requirement of communication or execution time, is used in the data packet scheduling to reduce communication overhead in distributed environments. Simulation results show that the proposed protocol outperforms the slotted floor acquisition multiple access (SFAMA), reverse opportunistic packet appending (ROPA), and distributed scheduling based concurrent transmission (DSCT) protocols in throughput, packet delivery rate, and average energy consumption in the price of larger end-to-end delay introduced by TDMA based handshaking.

Keywords: underwater acoustic network; medium access control; multi-channel; parallel transmission; pipeline; replication computation

1. Introduction

Underwater wireless communication and networking have attracted considerable attentions in recent years [1–3]. They are widely needed in the applications of marine research, oceanography, marine commercial exploitation, mine reconnaissance, tsunami warning, offshore oil industry, defense, etc. [1,2]. Because acoustic waves are the only known media that can propagate long distances in the water, most of the current underwater wireless communication and networking techniques are based on acoustic signals. However, underwater acoustic channels, which are characterized by limited bandwidth, high and variable propagation delay, time-varying multi-path and fading, and large

Doppler spread shift, resulting in high bit error rates and temporary losses of connectivity, creates great challenges for underwater acoustic networking [3].

Because the bandwidth of the underwater acoustic channel is narrow, the transmission rate between two nodes is usually very low in underwater acoustic networks. Therefore, increasing the transmission parallelism among network nodes is one of the most effective ways to improve the underwater acoustic network performance. Three types of parallel transmission opportunities often occur in underwater environments. The first is the time domain parallel transmission opportunity which is caused by the large propagation delay among network nodes. Because the links in the underwater acoustic network usually have large and different delays, multiple packets can be transmitted and propagate in the same channel without interfering with each other at their receivers if their transmission time is carefully scheduled. The second is the space domain parallel transmission opportunity, which is relative to the network topology. Two parts of the networks can send packets simultaneously when they are all located out of the interference regions of each other. The third is the multi-channel parallel transmission opportunity, where the sources and destinations can communicate using multiple parallel channels created by code division multiple access (CDMA) [4] or frequency division multiple access (FDMA) [5].

Usually, parallel transmission is implemented by the medium access control (MAC) techniques. In the contention-free MAC protocols, because CDMA and FDMA can provide each node an independent channel, which means that nodes can send data packets whenever needed, the researches on parallel transmission were mostly focused on time division multiple access (TDMA). Time and space domain parallel transmission opportunities were often used to optimize the TDMA time slot allocation [6–12]. Because no control information is needed in TDMA, the design of parallel transmission is simplified to the question of static communication resource allocation, which can be efficiently solved by many mathematical tools and methods. In the contention-based MAC protocols, the communication resources are allocated dynamically, so designing a parallel transmission protocol should consider not only the communication resource allocation, but also the control information exchange process. In the single channel contention-based MAC protocol [13–21], parallel transmission was mainly implemented in time and space domain since there was only one channel. In the multi-channel contention-based MAC protocol [22–27], parallel transmission can be implemented through multiple channels and most of the attentions were attracted to the channel negotiation mechanism, while time and space domain parallelism were seldom considered.

Handshaking is a common technique to reserve the channel for data transmission in contention-based MAC protocols. To implement parallel transmission, the MAC protocol should support multiple successful handshakes in a transmission cycle. In previous research, some handshaking mechanisms have been proposed to handle this problem for the single channel MAC protocols. One common technique is to use a sender- or receiver-initiated handshaking method, where when two nodes shake hands, other nodes that have data to send to the source or destination send their handshake packets to this source or destination along with the original handshake packets. However, in this method, concurrent transmissions only occur when other nodes have data to send to the current source or destination, which will sometimes reduce the chance of concurrent transmission. In [18], we proposed a handshaking method where multiple nodes sent their request to send (RTS) and clear to send (CTS) packets at a time randomly distributed in the RTS and CTS stages. This method imposes less constraint on the source and destination nodes so that can increase the opportunity of concurrent transmission in a transmission cycle and provide better performance. The major drawback of this method is that a node will fail to receive CTS packets due to collision and miscalculate its transmission time. This phenomenon becomes more severe when the number of nodes that participate in the handshaking process increases.

In this paper, we propose a new collision-free hybrid MAC protocol based on pipeline parallel transmission for distributed multi-channel underwater acoustic networks. The contributions of our work include:

- A new collision-free hybrid MAC framework is proposed, where the handshaking and data transmission are implemented as a pipeline on multiple acoustic channels. Handshaking is implemented using TDMA in a dedicated control channel and the data packets are transmitted in one or multiple data channels. This framework has two advantages. The first is that the TDMA based handshaking can support multiple successful handshakes in a transmission cycle while avoiding the control packet loss due to collision. The second is that the pipeline transmission of control and data packets can provide high throughput for the network by making use of the multi-channel transmission parallelism.
- The algorithms for optimizing the time slots of the TDMA based handshaking and the schedules of data packet transmission are proposed. These algorithms can achieve collision-free concurrent transmission by making use of the time domain parallel transmission opportunities in the underwater environment, which will let the transmission schedule more compact and improve the transmission efficiency effectively.
- The replication computation technique is employed in the data packet scheduling to reduce the control overhead in distributed environments. Replication computation, also known as redundant computing, is a technique that uses unnecessary and redundant computation to reduce the communication and/or execution time in parallel computing. By using replication computation, the data packet scheduling can be implemented via a two-way handshake, which is the simplest form of the handshaking protocol and has the least overhead.

The rest of this paper is organized as follows: in Section 2, the works related to parallel transmission in underwater acoustic networks are reviewed. In Section 3, the new collision-free hybrid MAC protocol is proposed. Sections 4 and 5 present and discuss the simulation results, respectively. Finally, Section 6 concludes the paper.

2. Related Work

2.1. Parallel Transmission Opportunities in Underwater Environments

There are mainly three types of parallel transmission opportunities in the underwater environments that have been noticed and utilized in the previous researches, i.e., the time domain, space domain, and multi-channel opportunities.

The time domain parallel transmission opportunity arises from the large propagation delay among network nodes. The sound propagation speed in underwater is around 1500 m/s, which results in a long propagation delay in the underwater communication. This propagation delay has many negative effects, but it also leads to concurrent transmission opportunities in time domain. Figure 1 gives an example of the time domain parallel transmission in underwater environments, where data packets can be sent concurrently by the source S1 and source S2 without overlapping at the destination D1 and destination D2 if their transmission time is scheduled according to the propagation delay between S1 to D1 and S2 to D2.

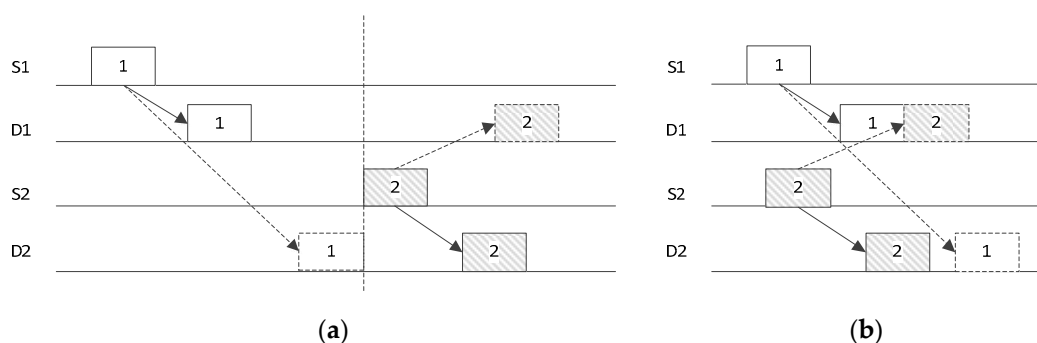


Figure 1. Serial transmission and time domain parallel transmission. (a) serial transmission; (b) time domain parallel transmission.

The space domain parallel transmission opportunity comes from the fact that two parts of the networks can send packets simultaneously when they are all located out of the interference regions of each other. Usually the interference region of a node is determined by its transmission power and the attenuation of the links, so the space domain parallel transmission opportunity can not only be explored from the location of nodes, but also be created by changing the transmission power of nodes. In Figure 2, the source S1 and source S2 can communicate with their corresponding destination D1 and destination D2 simultaneously when they reduce their transmission power.

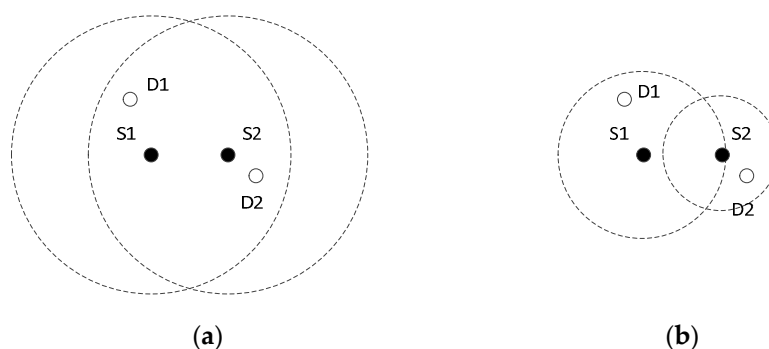


Figure 2. An example of creating a space domain parallel transmission opportunity by power control. (a) transmit with normal power; (b) transmit with reduced power.

The multi-channel parallel transmission opportunity resides in the multi-channel networks, where multiple parallel channels are created by CDMA or FDMA. Different from the pure CDMA and FDMA, the channel number is usually less than the node number in a multi-channel MAC protocol. Therefore, these kinds of protocols are mostly content-based, which means that the nodes need to compete for the channels if they have data to send.

2.2. Parallel Transmission MAC Protocols for Underwater Acoustic Networks

Parallel transmission is one of the effective ways to improve the performance of a communication network when the transmission rates of the network links are limited. Many MAC protocols based on this technique have been proposed for underwater acoustic networks in recent years.

In contention-free MAC protocols, the researches on parallel transmission were mostly focused on TDMA. Many TDMA protocols used the time domain parallel transmission to optimize their time slot allocations [6–12]. Some of them also made use of the network topology to reuse some time slots [8–12], which can be regarded as passive space domain parallel transmission. Some information such as propagation delay, link attenuation or node location needed to be known prior for calculating the communication resource in these protocols. The time and space domain parallelism help to form a more compact time slot schedule so that the channel can be utilized more efficiently in TDMA networks. Because no control information is needed in TDMA, the design of parallel transmission is simplified to the problem of static communication resource allocation. This problem can be solved by many mathematical tools and methods. Therefore, TDMA protocols with parallel transmission are easy to implement and have attracted many attentions in the previous studies.

In contention-based MAC protocols, all communication resources are allocated dynamically, so designing a parallel transmission contention-based protocol should consider not only how to optimize the allocation of communication resources, but also the exchange process of control information. The contention-based MAC protocols with parallel transmission can be further divided into single channel and multi-channel protocols.

In single channel contention-based MAC protocols, all control and data packets are transmitted through a common channel. Therefore, most of the parallel transmission opportunities in the single channel underwater acoustic networks come from the time and/or space domains. Time domain parallel transmission was usually implemented by scheduling the transmission time of the source

nodes [13–18], while space domain parallel transmission was achieved by controlling the source node's transmission power [19–21]. Optimization technique for parallel transmission in TDMA can be used to implement the communication resource allocation in the single channel contention-based MAC protocols, except that the allocation of the time or space communication resources needs to be performed every transmission cycle in the latter. The exchange process of control information is another important question in designing the parallel transmission protocol. Usually multiple successful handshakes are required before the data transmission. Some multiple handshake mechanisms have been proposed in the previous researches. One common technique is to use a sender- or receiver-initiated handshake process, where when two nodes shake hands, other nodes that have data to send to the source or destination node send their handshake packets to this source or destination along with the original handshake packets. However, in this method, additional concurrent transmissions only occur when there are nodes have data to send to the current source or destination node, which will sometimes reduce the concurrent transmission opportunities. Another method is to send the RTS and CTS packets at a time randomly distributed in the RTS and CTS phases. This method imposes less constraint on the source and destination nodes but will cause CTS packet collisions, which will lead to miscalculating the transmission time of the source node. Collisions will also occur more frequently as the number of nodes participating in the handshake process increases in this method.

The multi-channel contention-based MAC protocols use more than one channel for communication. Recent studies showed that such a parallelism can improve the performance of underwater acoustic networks effectively [22]. Multi-channel MAC protocols can be classified into two categories: single rendezvous and multiple rendezvous [23]. In single rendezvous protocols [23–26], all control packets are exchange over one common control channel. This method can solve the problem that the source may fail to shake hand with the destination since they reside on different channels. However, the control channel will become a bottleneck of the network when the traffic gets high. In multiple rendezvous protocols [27], nodes can simultaneously shake hands on multiple channels. This method hasn't bottlenecks, but special coordination is required to guarantee that the source and its correspond destination reside on the same channel when the handshaking happens.

3. Collision-Free Hybrid MAC Protocol Based on Pipeline Parallel Transmission

3.1. Motivation

The random handshaking method proposed in [18] supported arbitrary multiple successful handshakes in a transmission cycle, which can increase the parallel transmission opportunity. However, control packet collisions will occur in this method and lead to data packet collisions. Implementing the handshaking in a collision-free manner can solve the problem of control packet collision while supported arbitrary multiple successful handshakes. TDMA is a good choice for the implementation of collision-free handshaking, where the RTS and CTS packets can be transmitted in separated time slots. The disadvantage of the TDMA based handshaking is that it will bring a long delay to the handshaking process, leading to low transmission efficiency if the data packets are still transmitted in the same channel, especially when the scale of the network is large. Transmitting control and data packets through multiple channels is a possible solution to this problem, because data transmission can still be performed during handshaking. Therefore, a hybrid multi-channel MAC structure is adopted in this paper, which has the following features:

First, a hybrid multi-channel MAC framework with pipeline transmission is employed. It is a single rendezvous protocol, where handshaking is performed in a dedicated control channel while the data packets are transmitted in one or multiple data channels. The TDMA based handshaking and data transmission are implemented as a pipeline to provide high throughput for the network. The pipeline cannot reduce the transmission time of a single data packet, but can improve the transmission efficiency effectively for multiple data packets.

Second, time domain parallel transmission is employed in the handshaking time slot allocation and data packet scheduling to further improve the performance of the proposed protocol. By making use of the propagation delay differences, the time slots of the handshaking and the transmission schedules of data packets are more compact.

Third, the replication computation technique is employed in data packet scheduling to reduce the control overhead in distributed environments. By using replication computation, each source node can obtain the transmission channels and time of all source nodes via independent calculation, which can implement the data packet scheduling via two-way handshakes and reduce the requirement of control information exchange significantly.

3.2. Network Model

To simplify the discussion, we first consider a one-hop cluster in a distributed underwater acoustic network, which has the following characteristics:

- All nodes in the cluster are static, randomly distributed and can reach each other directly using an omnidirectional and half-duplex underwater acoustic modem.
- Nodes share a common clock since time synchronization among distributed nodes is achievable in underwater wireless networks [16].
- A delay table that contains all the inter-nodal propagation delays in the cluster is stored by each node, which is obtained in the network initialization stage by measuring the round-trip time and exchanging information with its neighbors [15,18].
- All nodes have one control channel and at least one data channel.
- Each node is equipped with a single transceiver, which means it can only listen, send or receive on one specific channel at the same time.

3.3. Protocol Description

In the concurrent transmission protocol proposed in [18], a transmission cycle includes four stages: the RTS, CTS, data, and acknowledgement (ACK) stage. Multiple nodes send their RTS and CTS packets in a random time of the RTS and CTS stages, respectively. Following this, the source nodes that receive the CTS packets schedule and transmit their data and ACK packets concurrently in the data and ACK stages, respectively. These four stages need to be performed in series in a single channel network. However, when there are multiple channels in the network and multiple data packets needed to be sent, this four-stage transmission process well meets the requirements of a pipeline, where:

- A repetitive process can be divided into several stages, each of which can be implemented by an independent device.
- These stages are connected with one another to form a pipe like structure.
- Multiple processes can be overlapped during execution so that the stages in each process can be carried out in parallel.

Figure 3 shows an example of implementing this transmission process with a four-stage pipeline, where Channel 3 to channel N-1 are parallel channels for data transmission and can be regarded as one stage in the pipeline. In Figure 3, it can be seen that when the RTS, CTS, data and ACK packets are transmitted in separated channels, a new transmission cycle can start along with the former CTS stage, which can make the whole transmission process more compact and increase the throughput of the network. Note that multiple RTS, CTS, data and ACK packets from different nodes can be transmitted in each RTS, CTS, data and ACK stages, respectively.

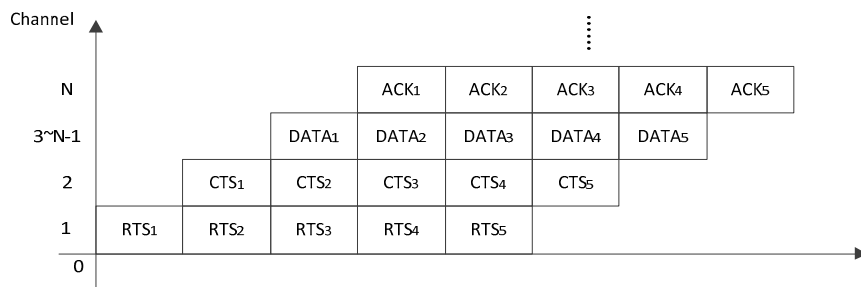


Figure 3. Implementing the typical transmission process with a four-stage pipeline.

A four-stage transmission process usually needs to be implemented with a four-stage pipeline, which means that the network needs more than four independent channels. This requirement is not always satisfied in practice and too many control channels will also reduce the transmission efficiency. However, it can be noticed that in the TDMA based handshaking, a node can only reside in the RTS, CTS or ACK stage at the same time and send a RTS, CTS or ACK packet in its time slot of the corresponding channel. There is lots of redundancy in the control packet transmission. For example, in the stage of RTS₂ in Figure 3, only the time slot for the second source node in Channel 1 is busy, while the other time slots are idle. Therefore, the RTS, CTS and ACK stages which belong to successive transmission cycles but start at the same time can be merged into one control channel and transmitted in the same TDMA period, as shown in Figure 4. In this method, the time slot assigned to each node in the control channel can be used to transmit RTS, CTS or ACK packet as needed. Two channels are sufficient for the pipeline, i.e., one control channel and one data channel. Because the control packets are still transmitted in separated time slots and each node has the chance to send control packets in its time slot, the proposed protocol can support multiple successful handshakes and cause no collision in the control packet transmission.

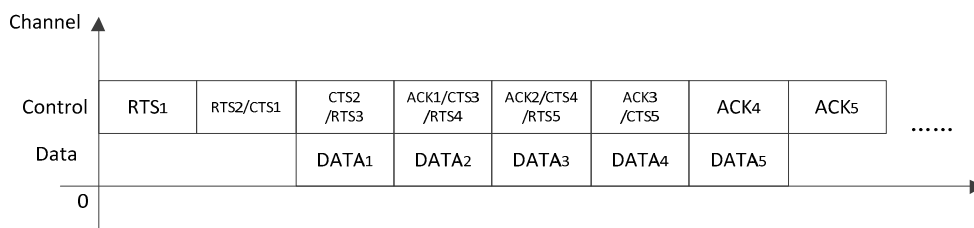


Figure 4. Merging the request to send (RTS), clear to send (CTS), and acknowledgement (ACK) stages into one control channel.

The advantage of this framework is that the handshaking is collision-free and the pipelining can increase the throughput of the network significantly. However, the TDMA based handshaking also brings a fixed delay to the data transmission, which cannot be shortened by the pipelining technique. Therefore, the time slots should be optimized to shorten the TDMA period, as well as the data packet scheduling should also be performed to make sure that no collision will occur during the data stage while the whole data transmission time is as short as possible. Theoretically, the pipeline will reach its maximum throughput when the TDMA period is equal to the duration of data transmission, so the data packet size is an important factor to the transmission efficiency and needs to be chosen properly.

3.4. Optimization of the Handshaking and Data Transmission Schedules

To make the transmission schedule more compact, the time slots of the TDMA handshaking and the transmission schedule of data packets are optimized according to the propagation delay between nodes in the proposed method. The optimizations of the TDMA time slot and the data packet schedule are similar except that there are only one channel and a fixed number of source nodes in the former while there might be more than one channel and a variable number of source nodes in the latter. Note

that the transmission sequence and time of the nodes will both affect the transmission efficiency [18]. Several methods can be used for the transmission sequence and time optimization, such as the dynamic programming and greedy algorithm. However, when the channel and source node number are large, large computational complexity is required in searching the best channel and transmission time using dynamic programming. Therefore, the greedy algorithm is employed in the proposed protocol to optimize the schedules for both the control and data packet transmission. Furthermore, two suboptimal assumptions are also adopted to simplify the discussion. The first is that a node can only transmit or receive a data packet in the same transmission cycle. The second is when node i sends out its packet before node j , the packet from node i will arrive at both destinations earlier than that from node j . These assumptions lead to the condition of collision-free transmission as:

$$T_j \geq T_i + L_i + \max(D_{i,i'} - D_{j,i'}, D_{i,j'} - D_{j,j'}), \tag{1}$$

where T_i and T_j denote the sending time of the packets from node i and node j , respectively. L_i is the duration of sending out the packets from node i . $D_{i,j'}$ denotes the propagation delay from node i to the destination of node j .

3.4.1. Time Slot Allocation for Handshaking

In a TDMA period, each node is allocated a fixed time slot, which means that each node can send out a packet as needed. Assume that a TDMA period starts at time T_0 . For a cluster consisted of N nodes, the start of the time slot for each node can be determined using the following steps according to the condition of collision-free transmission provided by Equation (1):

1. Initiate an empty queue \mathbf{Q}_0 for the nodes whose time slots are determined. Create a set $\overline{\mathbf{Q}}_0 = \{Q_i | 1 \leq i \leq N\}$ for the nodes whose time slots are not determined, where Q_i denotes node i in the cluster. Set $r = 1$.
2. Pick a node from $\overline{\mathbf{Q}}_{r-1}$ and try to add it to the tail of \mathbf{Q}_{r-1} . Assume that the selected node is Q_i . The optimal transmission time for Q_i can be calculated using the following equation:

$$T_{r,Q_i}^{slot} = \begin{cases} T_0, & r = 1 \\ \max_l (T_{r-1}^{slot} + D_{Q_{tail},Q_l} - D_{Q_i,Q_l}) + L^{slot} + C, & r > 1, 1 \leq l \leq N \end{cases} \tag{2}$$

where T_{r,Q_i}^{slot} denotes the start of the time slot for Q_i . Q_{tail} denotes the tail node of \mathbf{Q}_{r-1} . T_{r-1}^{slot} is the start of the time slot for Q_{tail} . $L^{slot} = \max(L^{RTS}, L^{CTS}, L^{ACK})$, where L^{RTS} , L^{CTS} and L^{ACK} denote the transmission durations of the RTS, CTS and ACK packets. C is the guard time required to accommodate the variations of the propagation delay.

3. Select the node from $\overline{\mathbf{Q}}_{r-1}$ that minimizes the ending time of \mathbf{Q}_{r-1} when this node is added to its tail, i.e.,

$$Q_r^{opt} = \underset{Q_i}{\operatorname{argmin}} \left(\max_l (T_{r,Q_i}^{slot} + D_{Q_i,Q_l}) \right), 1 \leq l \leq N, Q_i \in \overline{\mathbf{Q}}_{r-1} \tag{3}$$

4. Calculate the start of the time slot for Q_r^{opt} using the following equation:

$$T_r^{slot} = \begin{cases} T_0, & r = 1 \\ \max_l (T_{r-1}^{slot} + D_{Q_{tail},Q_l} - D_{Q_r^{opt},Q_l}) + L^{slot} + C, & r > 1, 1 \leq l \leq N \end{cases} \tag{4}$$

Add Q_r^{opt} to the tail of \mathbf{Q}_{r-1} and delete it from $\overline{\mathbf{Q}}_{r-1}$.

- If $r < N$, then $r = r + 1$ and repeat steps 2 to step 5 for another iteration. Otherwise, the node order in \mathbf{Q}_N is the optimized time slot sequence. The time slot for the r -th node in \mathbf{Q}_N is $[T_r^{slot}, T_r^{slot} + L^{slot}]$. The total length of the TDMA period is given by

$$T^{TDMA} = \max_l \left(T_N^{slot} + D_{Q_N^{opt}, Q_l} \right) + L^{slot} + C, \quad 1 \leq l \leq N \quad (5)$$

3.4.2. Data Packet Scheduling

The major difference between data packet scheduling and TDMA time slot assignment is that there might be more than one channel and a variable number of source nodes in the former. Therefore, the above optimization method is modified to adapt to these changes for data packet scheduling as the following:

- Let $S_i = (s_i, d_i)$, $1 \leq i \leq K$, denotes the source and destination pairs in the current transmission cycle, where s_i , d_i , and K are the source node, destination node and the number of the source and destination pairs, respectively. Initiate M empty queue \mathbf{S}_0^m , $1 \leq m \leq M$, for S_i whose source transmission time is determined, where M is the data channel number. Create a set $\bar{\mathbf{S}}_0 = \{S_i | 1 \leq i \leq K\}$ for S_i whose source transmission time is not determined. Set $r = 1$.
- Pick S_i from $\bar{\mathbf{S}}_{r-1}$ and try to allocate it to the m -th data channel. The optimal transmission time for s_i in the m -th data channel can be calculated by:

$$T_{r,m,s_i}^{data} = \begin{cases} T_0, & r = 1 \\ \max_{s_l} \left(T_{r-1,m,s_l}^{data} + D_{s_l,d_l} - D_{s_i,d_l} \right) + L_{m,s_l}^{data} + C, & r > 1, (s_l, d_l) \in \mathbf{S}_{r-1}^m \end{cases} \quad (6)$$

where T_{r,m,s_i}^{data} denotes the transmission time of s_i when it is allocated to channel m in the r -th iteration. T_{r-1,m,s_l}^{data} is the transmission time of s_l , where $S_l = (s_l, d_l)$ is an element of \mathbf{S}_{r-1}^m , i.e., the source and destination pairs assigned to channel m in the prior iterations. L_{m,s_l}^{data} denotes the transmission duration of the data packet from s_l on channel m . C is the guard time required to accommodate the variations of the propagation delays.

- Select the node and the channel which minimizes the current data transmission time using the following formula:

$$(s_r^{opt}, m_r^{opt}) = \underset{s_i, m}{\operatorname{argmin}} \left(\max_{d_l} \left(T_{r,m,s_i}^{data} + D_{s_i,d_l} \right) + L_{m,s_i}^{data} \right), \quad (s_l, d_l) \in \bar{\mathbf{S}}_{r-1}, \quad 1 \leq m \leq M \quad (7)$$

- Calculate the transmission time for s_r^{opt} as:

$$T_{r,m_r^{opt},s_r^{opt}}^{data} = \begin{cases} T_0, & r = 1 \\ \max_{s_l} \left(T_{r-1,m_r^{opt},s_l}^{data} + D_{s_l,d_l} - D_{s_r^{opt},d_l} \right) + L_{m_r^{opt},s_l}^{data} + C, & r > 1, (s_l, d_l) \in \mathbf{S}_{r-1}^{m_r^{opt}} \end{cases} \quad (8)$$

Add S_r^{opt} to the tail of $\mathbf{S}_{r-1}^{m_r^{opt}}$ and delete it from $\bar{\mathbf{S}}_{r-1}$.

- If $r < N$, then $r = r + 1$ and repeat steps 2 to step 5 for another iteration. Otherwise, the nodes and their order in \mathbf{S}_N^m is the optimized transmission channels and sequences for all the source node in current transmission cycle. The total data transmission time is given by

$$T^{data} = \max_{m,s_i,d_l} \left(T_{m,s_i}^{data} + D_{s_i,d_l} + L_{m,s_i}^{data} \right) + C, \quad (s_i, d_i) \in \mathbf{S}_N^m, \quad (s_l, d_l) \in \mathbf{S}_N^m, \quad 1 \leq m \leq M \quad (9)$$

3.4.3. Duration of the Pipeline Stage

Obviously, the duration of data transmission varies for each transmission cycle because the source and destination nodes will change with time. When each node can obtain the data packet schedule of all nodes in the network, it can synchronize with other nodes by predicting the beginning of the next pipeline stage. It means that the duration of a pipeline stage can be variable even when the handshaking is implemented with TDMA. Therefore the best choice for the duration of a pipeline stage is:

$$T^{stage} = \max(T^{TDMA}, T^{data}) \tag{10}$$

In such cases, all data packets can be transmitted no matter the data transmission duration is longer than the TDMA period or not. On the other hand, when the data packet schedule cannot be known by all nodes in the network, like in a multi-hop situation, the pipeline stage needs to be consistent with the TDMA period for synchronizing, i.e.;

$$T^{stage} = T^{TDMA} \tag{11}$$

It means that some data packets need to be postponed when the duration of data transmission is larger than the TDMA period, which will lead to performance degradation.

3.5. Replication Computation

Because the source and destination nodes will change in each transmission cycle, the optimization result for data packet scheduling should be known by all source nodes. If the optimization algorithm is carried out by one node, additional overhead is required for broadcasting the optimization result to other source nodes. To reduce the control overhead for data packet scheduling, replication computation technique is employed in this paper.

Replication computation, also known as redundant computing, is a technique that uses unnecessary and redundant computation to reduce the communication and/or execution time in parallel computing. Figure 5 shows an example of replication computation. In this example, the sum of the two inputs needs to be stored in both node A and node B. In Figure 5a, the sum is first calculated by A, and then passed from A to B. In Figure 5b, the two inputs are first broadcasted to A and B, and then the sum is calculated in both A and B simultaneously. The latter method can omit the communication between A and B in the price of repeating the same computation in B.

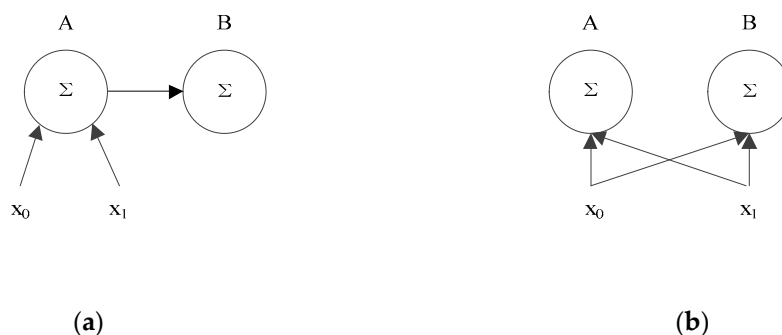


Figure 5. An example of replication computation. (a) common method; (b) replication computation.

There are several conditions that should be met when applying replication computation to an operation in a parallel process:

- The input of the operation is variable, or else the output of the operation can be calculated in advance and there is no need for replication computation.
- The output of the operation can be planned or predicted.
- Multiple nodes need to know the output of the operation.

- The operation can produce the same outputs when the inputs are equal.

The data transmission scheduling in the proposed protocol has well met the requirements of replication computation. First, the nodes needed to transmit data vary in each transmission cycle. Second, the optimal scheduling can be calculated by an algorithm, which means its output is predictable. Third, all source nodes need to know the scheduling results. Forth, when the scheduling algorithm is deterministic, its can ensure that the same inputs result in the same outputs. Figure 6 shows an implementation of the data packet scheduling with and without replication computation. The conventional method without replication computation is shown in Figure 6a, where the information of successful handshakes is first collected by node A, and then node A calculates the optimal scheduling result and broadcasts it to the other nodes. Figure 6b shows the scheduling with replication computation, where the information of successful handshakes is collected by all nodes, and then each node calculates the scheduling result using the same deterministic algorithm. Note that in Figure 6b, the broadcasting ensures that the scheduling algorithms in all nodes have the same inputs so that they can generate the same outputs, i.e., the same transmission schedules. It can be seen that by using replication computation, the data packet scheduling can be implemented via a two-way handshake, which is the simplest form of the handshaking protocol and has the least overhead.

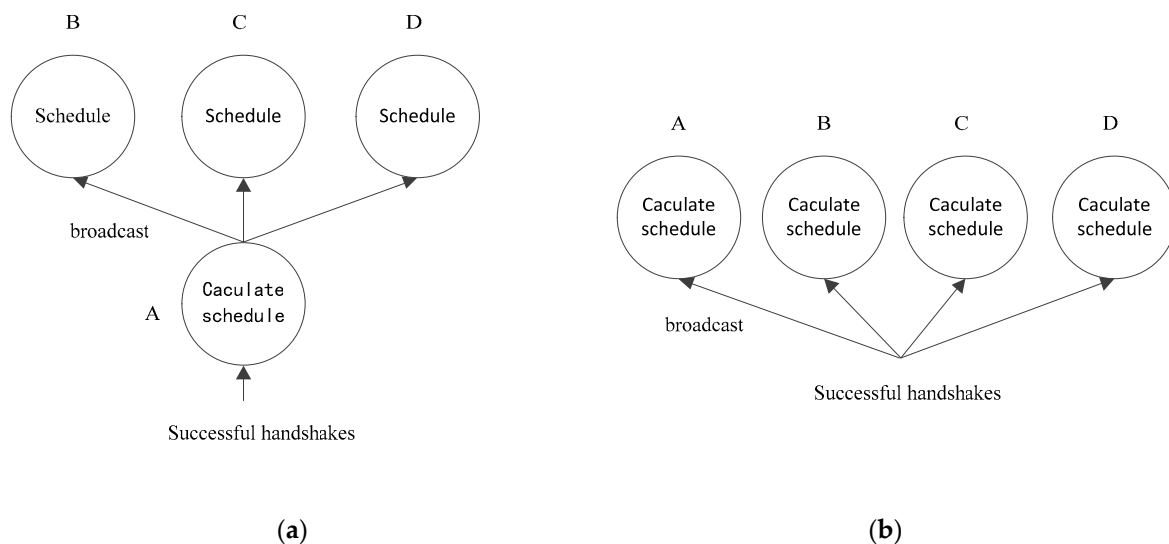


Figure 6. Applying replication computation to parallel data transmission. (a) conventional method; (b) replication computation.

3.6. Multi-Hop Extension

The proposed method can be easily extended to multi-hop networks. First, divide the multi-hop network into several one-hop clusters. Then allocate the TDMA time slots in each cluster. The proposed time slot allocation method can be performed in each cluster independently, except that when a node belongs to more than one cluster, all nodes in these clusters should be considered in calculating its time slot using Equation (2). The replication computation can also be used to calculate the data packet schedules inside a cluster. However, because nodes without direct links cannot obtain the handshaking result from each other through broadcasting, the nodes belong to multiple clusters can only joint the communication of one cluster. Furthermore, while other nodes can communicate normally, the nodes belong to multiple clusters should stop their communication when they hear any RTS packets from different clusters in the same transmission cycle to avoid mutual interference. The transmission cycle should also be fixed in all clusters for the purpose of synchronizing, which means that in some cases not all the data packets can be sent out in the current transmission cycle as in the single-hop situation. Therefore, the transmission efficiency in the multi-hop network is not as high as that in the one-hop network.

4. Simulations and Results

The simulations were implemented by Aqua-Sim [28], which is an NS-2 based simulator for underwater sensor networks and developed at the Underwater Sensor Network (UWSN) Lab at the University of Connecticut. In the simulations, the nodes are randomly located in a square area $1.5 \times 1.5 \text{ km}^2$ and 500 m deep. The single-hop static network is tested in the simulations, i.e., all nodes in the network are static and can reach the other nodes directly. In the network initialization stage, each node measures the round-trip time with other nodes, and broadcasts this information to the others to obtain an inter-nodal propagation delay table. The acoustic propagation speed is set to 1500 m/s. Transmission rate is set to 1200 bps. Control packet size is 10 bytes and the data packet size varies as required. The powers of sending, receiving, and idle states are set to 20, 1 and 0.5 Watt, respectively. Data packets generation for each node follows a Poisson distribution, and the destination of each packet is selected randomly with equal probability. Channel-related packet loss is ignored, i.e., packet loss only occurs due to packet collisions. A two channel network is tested for the proposed protocol, where one channel for control packet transmission and another channel for data packet transmission. The ratio of the control and data channel's transmission rates is set to 0.7, while the total transmission rate of the two channels maintains 1200 bps as the other protocols.

In the simulations, the throughput, packet delivery rate, average end-to-end delay, and average energy consumption of the proposed method are compared with the slotted floor acquisition multiple access (SFAMA), reverse opportunistic packet appending (ROPA) [14] and distributed scheduling based concurrent transmission (DSCT) [18] protocols with different network loads, data packet sizes and node numbers. SFAMA is a typical serial access protocol while the others are all parallel access protocols. In ROPA, concurrent transmissions will occur when other nodes have data to send to the source node. DSCT supports multiple handshakes and concurrent transmissions for arbitrary source and destination nodes in one transmission cycle, which can utilize more concurrent transmission opportunities than ROPA. The proposed protocol also supports multiple handshakes and concurrent transmissions as DSCT, except that the handshaking is implemented in a dedicated control channel by TDMA and the transmission rate of the data channel is reduced. The throughput, packet delivery rate, average end-to-end delay, average energy consumption and normalized offered load are defined as [29]:

- Throughput: the average number of data bits successfully received by the intended destinations per second (in bit/s).
- Packet delivery rate: the ratio of the number of data packets successfully delivered at the intended destinations to the total number of data packets generated (dimensionless).
- Average end-to-end delay: the average time interval between generation and successful delivery of data packets at the intended destinations (in ms).
- Average energy consumption: the ratio of the total energy consumption to the number of data packets successfully received by the intended destinations (in Joule/packet or J/pkt).
- Offered load: the average number of data packet generated per second for each node (in pkt/s).

4.1. Performance Under Different Network Loads

First SFAMA, ROPA, DSCT, and the proposed protocols are tested under different network loads with the node number of 10 and the data packet size of 400 bytes. The simulation results are shown in Figure 7.

Figure 7a shows the throughputs of all protocols under the offered loads from 0.3 to 5.0. It can be seen that SFAMA gives the lowest throughputs among the four protocols due to its serial transmission mode. ROPA has a similar throughput as SFAMA when the offered load is below 1.5, but becomes higher when the offered load increases. This is because that the opportunity for other nodes to send data to the source node increases as the offered load rises, which increases the concurrent transmission probability in ROPA accordingly. DSCT gives the second highest throughput in the simulation.

It is superior that ROPA because it imposes less constraints on the source and destination nodes and supports arbitrary multiple successful handshakes in a transmission cycle, which increases the concurrent transmission opportunity. The proposed method achieves the best performance. It reaches its maximum throughput around 4700 bit/s, which is significantly higher than DSCT and ROPA. This is because that the proposed method can support arbitrary multiple successful handshakes in a transmission cycle as DSCT, while eliminates the control packet losses due to collision by using a collision-free handshaking method. The pipeline parallel transmission also helps to increase the throughput of the proposed protocol. It can also be noticed that the throughput of the proposed method reaches its maximum quickly in Figure 7a. It is because that the throughput of the proposed method will become saturated when the duration of data transmission is equal or larger than the handshaking period.

Figure 7b shows the delivery rates of the four protocols. Because there is no control packet loss due to collision and almost all transmission requests can be satisfied, the delivery rate of the proposed method is nearly 100% in the simulations. The delivery rates of the other three protocols decrease as the offered load increases. DSCT gives higher delivery rates than ROPA because it increases the concurrent transmission opportunity. The SFAMA is inferior to the three parallel transmission protocols.

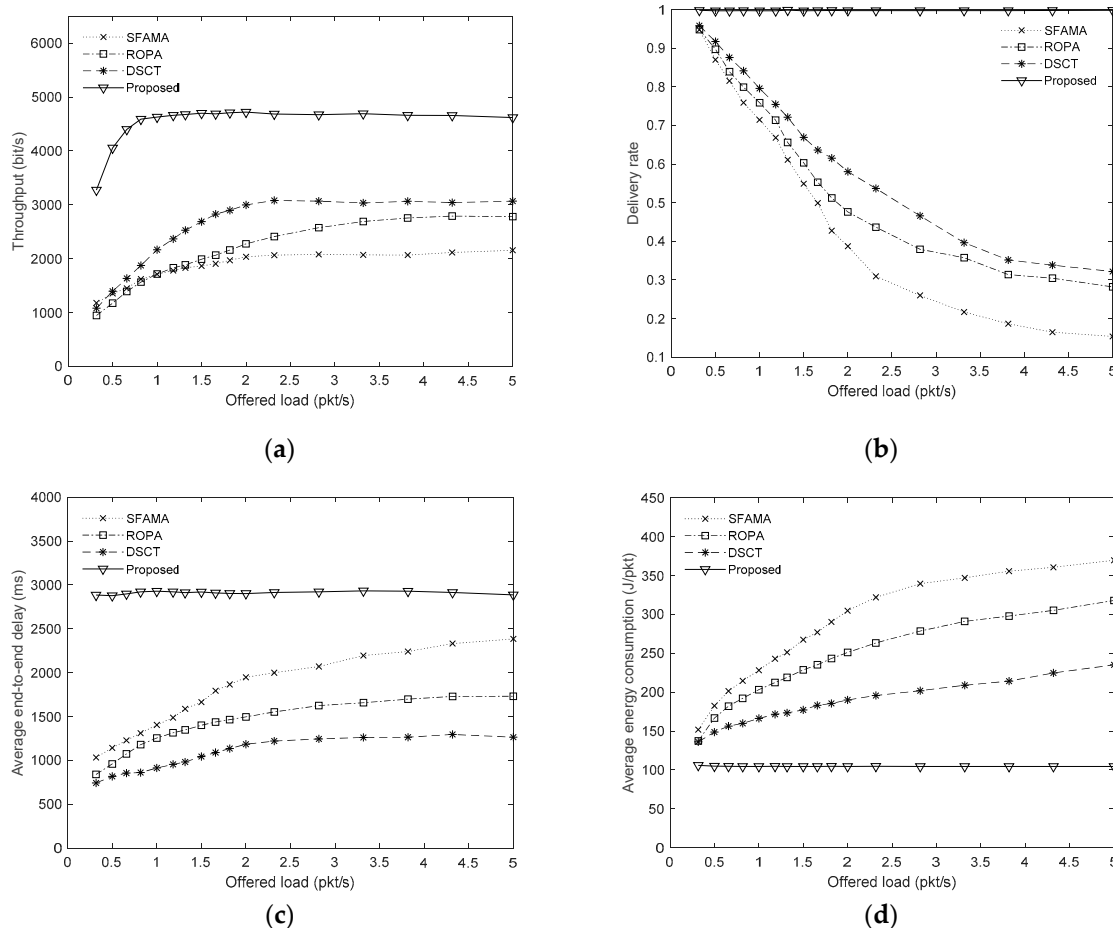


Figure 7. Performance of the slotted floor acquisition multiple access (SFAMA), reverse opportunistic packet appending (ROPA), and distributed scheduling based concurrent transmission (DSCT) protocols and that of the proposed method under different network loads. (a) throughput; (b) packet delivery rate; (c) average end-to-end delay; and (d) average energy consumption.

Figure 7c shows the average end-to-end delays of the four protocols. It can be seen that the average end-to-end delay decreases as the transmission parallelism increases for SFAMA, ROPA, and DSCT, where DSCT gives the lowest average end-to-end delay while SFAMA the highest among the

three protocols. The proposed protocol shows stable but larger average end-to-end delays under different offered loads than SFAMA, ROPA, and DSCT. It is because that the TDMA handshaking introduces a fixed delay to the transmission, which is usually larger than the normal handshaking delays and cannot be shortened by the pipeline technique. It is the price for eliminating the collision of control and data packets.

Figure 7d shows the average energy consumption of the four protocols. By eliminating the collision of control and data packets, the proposed protocol achieves the lowest and stable average energy consumption among the four protocols. The average energy consumption also decreases as the transmission parallelism increases for SFAMA, ROPA, and DSCT because higher transmission parallelism can reduce the handshake attempts for a data transmission, which will lead to fewer handshake packets wasted.

4.2. Influence of Packet Size

Because the handshaking and data transmission are implemented as a pipeline in the proposed protocol, the best performance will be achieved when the period of handshaking is equal to the duration of data transmission theoretically. Therefore, the data packet length is an important factor to the transmission efficiency and needs to be chosen properly.

Figure 8a shows the durations for the handshaking period and data transmission under different packet sizes, where the node number is set to 10 and the offered load is set to 5.0 pkt/s. It can be seen that the period of TDMA based handshaking is quite stable even when the network nodes are randomly distributed in the test zoom. The duration of data transmission increases linearly as the packet size becomes large. The duration for handshaking and data transmission is nearly equal when the packet size is around 400 bytes.

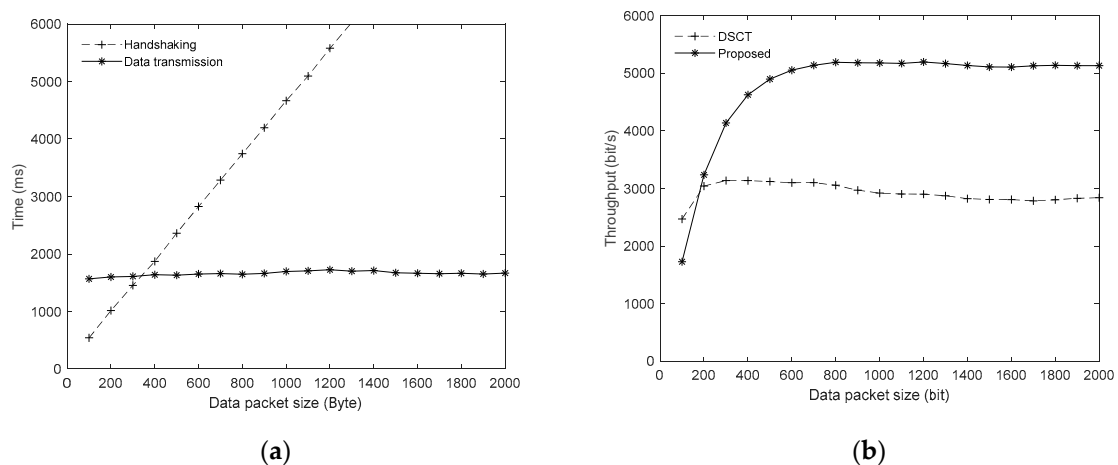


Figure 8. Influence of packet size. (a) durations for handshaking and data transmission under different packet sizes; (b) throughputs of DSCT and the proposed protocols under different packet sizes.

Figure 8b shows the throughputs of DSCT and the proposed protocols under different packet sizes. It can be seen that when the packet size is smaller than 200 bytes, the throughput of the proposed protocol is lower than DSCT. This is because when the packet size is small, the drawback of long handshaking period becomes significant and will reduce the throughput of the proposed protocol. On the other hand, when the packet size is larger than 200 bytes, the data transmission efficiency is large enough to compensate the negative effect of the long handshaking period and will become much higher than DSCT. It should be noted that the maximum throughput of the proposed protocol is achieved when the packet size is larger than 600, which does not coincide with the packet size at which the time for handshaking period and data transmission is nearly equal. This is because that although the total data transmission time increases with the packet size linearly, the number of packets

transmitted is not a linear function of the data transmission time due to concurrent transmission. A proper packet size might fill the vacant time better during the concurrent transmission so that the throughput can still increase when the packet size is larger than 400 bytes.

4.3. Performance Under Different Network Sizes

The network size is another important factor that affects the performance of the proposed protocol because it determines the duration of the TDMA based handshaking. The performance of SFAMA, ROPA, DSCT and that of the proposed protocol under different network sizes are shown in Figure 9. The packet size is set to 400 bytes and the offered load is set to 5.0 pkt/s in the simulations.

Figure 9a shows that the throughputs of SFAMA, ROPA and DSCT rise as the network size increases and become saturated at the network size above 10. The throughput of the proposed protocol is much higher than SFAMA, ROPA and DSCT. This means that the pipeline transmission and collision-free parallel handshaking and data transmission help to increase the network throughput. The throughput curve of the proposed protocol also shows a different trend. It decreases slightly as the network size increases. This is because when the network size increase, the duration of handshaking is prolonged. On the other hand, the optimization of the transmission schedules is not so effective as the node number increases, which will causes more vacant time in the transmission and decreases the transmission efficiency.

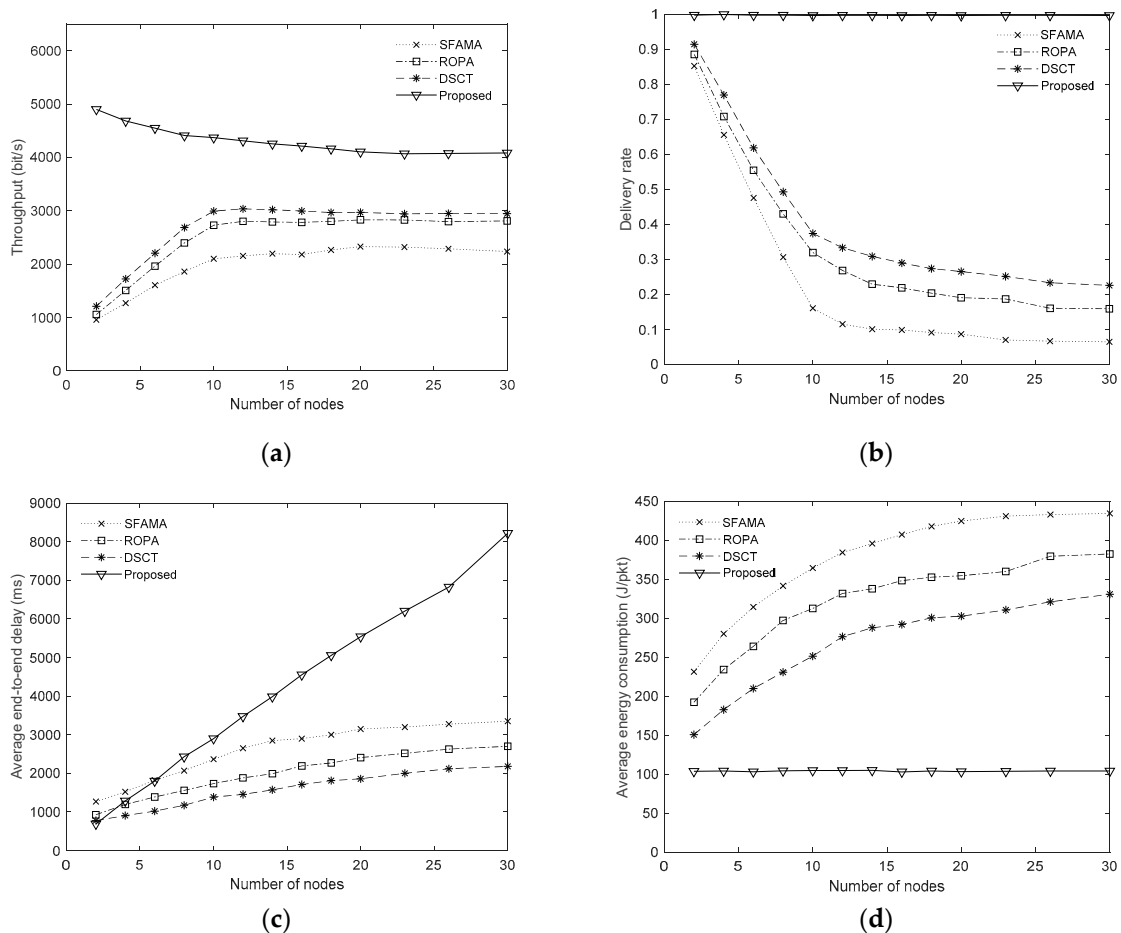


Figure 9. Performance of the SFAMA, ROPA, and DSCT protocols and that of the proposed method under different network sizes. (a) throughput; (b) packet delivery rate; (c) average end-to-end delay; and (d) average energy consumption.

In Figure 9b, the delivery rates of SFAMA, ROPA and DSCT decrease as the network size increases because the network load increases with the network size. The delivery rate of the proposed method is nearly 100% for in all the simulations due to the collision-free data transmission.

Figure 9c shows the average end-to-end delays of the four protocols under different network sizes. The average delays of all protocol rise as the network size increases. DSCT and the proposed protocol give the lowest average delay when the node number is two, while the former maintains the lowest average delay among the four protocols and the latter increases quickly as the node number increases. The proposed protocol gives the highest average delay when the node number is larger than eight. This disadvantage is caused by the TDMA based handshaking.

Figure 9d shows the average energy consumption of the four protocols under different network sizes. The average energy consumption of SFAMA, ROPA, and DSCT become larger as the network size increases. This is because the useless handshake attempts and data collisions increase as the network size increases under the same offered load. It can also be noticed that the average energy consumption decreases as the transmission parallelism increases for SFAMA, ROPA, and DSCT. The average energy consumption of the proposed protocol is the lowest among the four protocols and maintains the same level for different node numbers. This is because that there is no energy wasted due to collision in the propose protocol.

5. Discussion

SFAMA, ROPA, DSCT, and the proposed protocols are all handshaking based protocols, which have to content for the channel to transmit data. SFAMA is the only serial access protocol among the four protocols. Simulation results show that it achieves the worst performance in throughput, packet delivery rate, and average energy consumption. It is also worse than ROPA and DSCT in average end-to-end delay. This means that parallel transmission is an effective way to improve the performance of underwater acoustic networks. In ROPA, when nodes have data to send to the current source node, they can append their data transmission to the current data transmission according to the propagation delays. Therefore, ROPA allows concurrent transmission but sets a constraint to the source and destination nodes. On the other hand, DSCT supports multiple handshakes and concurrent transmissions for arbitrary source and destination nodes in one transmission cycle, which can utilize more concurrent transmission opportunity than ROPA. So DSCT is superior to ROPA in throughput, packet delivery rate, average end-to-end delay, and average energy consumption under different network loads and network sizes. The proposed protocol also supports multiple handshakes and concurrent transmissions as DSCT, except that the handshaking is implemented in a dedicated control channel by TDMA and the transmission rate of the data channels is reduced to maintain the same total transmission rate as the other protocols. The proposed protocol eliminates the control packet collision by using a content-free method for handshaking, therefore achieves significant improvements over DSCT in throughput, packet delivery rate, and average energy consumption when the packet size is properly selected.

The major disadvantage of the proposed protocol is its large average end-to-end delay. As shown in Figures 3 and 4, a transmission cycle in the proposed protocol consists of four pipeline stages. A message needs three pipeline stages to be passed from the source to its destination, i.e., the propagation delay of a message in a cluster is $3 \times T^{stage}$. After sending a message, the source and destination also need another pipeline stage for receiving or sending the ACK packet before processing the next message. Because $T^{stage} \geq T^{TDMA}$ and T^{TDMA} is determined by the size of a cluster, larger cluster size will lead to longer T^{stage} , as shown in Figure 9c. Therefore, the size of a cluster should be chosen carefully to keep the end-to-end delay in an acceptable range.

Except the large average end-to-end delay, the proposed protocol can also be improved in several aspects. First, using TDMA makes the proposed protocol inflexible to the network size change. Additional mechanism should be adopted to handle the situation when a node join or leave the network. Second, while replication computation allows a flexible duration of the pipeline stage in

a one-hop cluster, it cannot work so well in a multi-hop environment as in the single-hop situation because extra information exchange is needed to keep the inputs of the scheduling algorithm equal for two non-adjacent nodes, which is contradicted with the idea of replication computation. Therefore, the duration of the pipeline stage has to be fixed to the TDMA period in multi-hop networks, which might not be long enough to transmit all data packets in a transmission cycle so as to decrease the transmission efficiency of the proposed protocol. Third, in Equation (1), we assumed that if s_i sends out its data packet before s_j , then the data packets from s_i will arrive at d_i and d_j earlier than the data packets from s_j . This assumption sets a constraint on the packet arrival sequence at the destination node, which simplifies the question but leads to a sub-optimal solution. Further improvement can be expected by removing this constraint. The greedy algorithm is also sub-optimal, which can be improved by using more sophisticated optimal methods.

Finally, although the proposed protocol uses TDMA for handshaking and has some common features of TDMA, such as collision-free transmission and stable end-to-end delay, it is different from the TDMA-like protocols because it is basically content-based, where the data channels are dynamically allocated to different nodes in different transmission cycles. It is well known that the content-free protocols are more efficient for heavy, constant and balance traffics, while the content-based protocols are more suitable for light, burst and unbalance traffics. The proposed protocol can be viewed as a compromise of the content-free and content-based methods. Therefore, it outperforms the conventional handshaking based protocols but will not be as good as the TDMA-like protocols in the former situations, because the control packet exchanging will occupy a part of bandwidth while there is no control traffic in TDMA. However, the proposed protocol has the potential ability to outperform TDMA-like protocols for burst and unbalance traffics.

6. Conclusions

In this paper, we propose a new collision-free hybrid MAC protocol based on pipeline parallel transmission for distributed multi-channel underwater acoustic networks. Three new techniques are adopted in the proposed protocol. First, a collision-free hybrid MAC framework is proposed, where the handshaking and data transmission are implemented as a pipeline on multiple acoustic channels. Handshaking is implemented using the TDMA technique in a dedicated control channel to allow multiple successful handshakes in a transmission cycle and avoid collision. Second, the replication computation technique is employed in the data packet scheduling to reduce the control overhead in distributed environments. Third, greedy algorithms for TDMA time slot assignment and data packet scheduling is proposed to optimize the control and data packet transmission using the underwater time domain parallelism. Simulation results show that the proposed protocol outperforms SFAMA, ROPA and DSCT in throughput, packet delivery rate, and average energy consumption in the price of longer end-to-end delay introduced by the TDMA based handshaking.

Author Contributions: Conceptualization and methodology, J.Z., G.N.; software and validation, Z.H.; formal analysis and data curation, Z.H., Y.X., and J.Z.; writing—original draft preparation, Y.X. and J.Z.; supervision, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61871191, the Science and Technology Planning Project of Guangzhou, grant number 201804010209, and the Key Program of Marine Economy Development, Department of Natural Resources of Guangdong Province, grant number YZRZH[2020]009.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jiang, S. State-of-the-art medium access control (MAC) protocols for underwater acoustic networks: A survey based on a MAC reference model. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 96–131. [[CrossRef](#)]
2. Chen, K.; Ma, M.; Cheng, E.; Yuan, F.; Su, W. A survey on MAC protocols for under water wireless sensor networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1433–1447. [[CrossRef](#)]

3. Pompili, D.; Akyildiz, I.F. Overview of networking protocols for underwater wireless communications. *IEEE Commun. Mag.* **2009**, *47*, 97–102. [[CrossRef](#)]
4. Bo-Min, S.; Junho, C.; Ho-Shin, C. A Signaling-Free Underwater Code Division Multiple Access Scheme. *Electronics* **2019**, *8*, 880.
5. Qiao, G.; Liu, L.; Ma, L.; Yin, Y. Adaptive downlink OFDMA system with low-overhead and limited feedback in time-varying underwater acoustic channel. *IEEE Access* **2019**, *7*, 12729–12741. [[CrossRef](#)]
6. Lmai, S.; Chitre, M.; Laot, C.; Houcke, S. Throughput-efficient super-TDMA MAC transmission schedules in Ad Hoc linear underwater acoustic networks. *IEEE J. Ocean. Eng.* **2017**, *42*, 156–174. [[CrossRef](#)]
7. Luque-Nieto, M.A.; Moreno-Roldán, J.M.; Otero, P.; Poncela, J. Optimal scheduling and fair service policy for STDMA in underwater networks with acoustic communications. *Sensors* **2018**, *18*, 612. [[CrossRef](#)]
8. Zhang, R.; Cheng, X.; Cheng, X.; Yang, L. Interference-free graph based TDMA protocol for underwater acoustic sensor networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 4008–4019. [[CrossRef](#)]
9. Alfouzan, F.A.; Shahrabi, A.; Ghoreyshi, S.M.; Boutaleb, T. A collision-free graph coloring MAC protocol for underwater sensor networks. *IEEE Access* **2019**, *7*, 39862–39878. [[CrossRef](#)]
10. Alfouzan, F.A.; Shahrabi, A.; Ghoreyshi, S.M.; Boutaleb, T. An Efficient Scalable Scheduling MAC Protocol for Underwater Sensor Networks. *Sensors* **2018**, *18*, 2806. [[CrossRef](#)]
11. Alfouzan, F.A.; Shahrabi, A.; Ghoreyshi, S.M.; Boutaleb, T. An Energy-conserving Collision-free MAC Protocol for Underwater Sensor Networks. *IEEE Access* **2019**, *99*, 2169–3536. [[CrossRef](#)]
12. Bai, W.; Wang, H.; Shen, X.; Zhao, R. Link scheduling method for underwater acoustic sensor networks based on correlation matrix. *IEEE Sens. J.* **2016**, *16*, 4015–4022. [[CrossRef](#)]
13. Diamant, R.; Shi, W.; Soh, W.S.; Lampe, L. Joint time and spatial reuse handshake protocol for underwater acoustic communication networks. *IEEE J. Ocean. Eng.* **2013**, *38*, 470–483. [[CrossRef](#)]
14. Ng, H.H.; Soh, W.S.; Montani, M. An underwater acoustic MAC protocol using reverse opportunistic packet appending. *Comput. Netw.* **2013**, *57*, 2733–2751. [[CrossRef](#)]
15. Lee, J.W.; Cho, H.S. A hybrid sender- and receiver-initiated protocol scheme in underwater acoustic sensor networks. *Sensors* **2015**, *15*, 28052–28069. [[CrossRef](#)]
16. Han, Y.; Fei, Y. DAP-MAC: A delay-aware probability-based MAC protocol for underwater acoustic sensor networks. *Ad Hoc Netw.* **2016**, *48*, 80–92. [[CrossRef](#)]
17. Cho, J.; Shitiri, E.; Cho, H.S. Network allocation vector (NAV) optimization for underwater handshaking-based protocols. *Sensors* **2016**, *17*, 32. [[CrossRef](#)]
18. Zhang, J.; Lai, H.; Xiong, Y. Concurrent Transmission Based on Distributed Scheduling for Underwater Acoustic Networks. *Sensors* **2019**, *19*, 1871. [[CrossRef](#)]
19. Sun, S.; Xin, Y.; Li, Y.; Tan, Z.; Xing, Y. Interference rejection power control scheme for underwater acoustic sensor networks. In Proceedings of the 2018 International Conference on Virtual Reality and Intelligent Systems, Taichung, Taiwan, 10–12 December 2018.
20. Qian, L.; Zhang, S.; Liu, M.; Zhang, Q. A MACA-based power control MAC protocol for underwater wireless sensor networks. In Proceedings of the 2016 IEEE/OES China Ocean Acoustics, Harbin, Heilongjiang, China, 9–11 January 2016.
21. Han, Y.; Fei, Y.; Ding, A. SMARP: A stochastic MAC protocol with randomized power control for underwater sensor networks. In Proceedings of the 13th IEEE International Conference on Sensing, Communication, and Networking, London, UK, 27–30 June 2016.
22. Han, Y.S.; Deng, J.; Haas, Z.J. Analyzing multi-channel medium access control schemes with ALOHA reservation. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 2143–2152. [[CrossRef](#)]
23. Bouabdallah, F.; Zidi, C.; Boutaba, R.; Mehaoua, A. Collision avoidance energy efficient multi-channel MAC protocol for underwater acoustic sensor networks. *IEEE Trans. Mobile Comput.* **2019**, *18*, 2298–2314. [[CrossRef](#)]
24. Su, Y.; Jin, Z. UMMAC: A multi-channel MAC protocol for underwater acoustic networks. *J. Commun. Netw.* **2016**, *18*, 75–83.
25. Dang, D.N.M.; Nguyen, V.; Le, H.T.; Hong, C.S.; Choe, J. An efficient multi-channel MAC protocol for wireless ad hoc networks. *Ad Hoc Netw.* **2016**, *44*, 46–57. [[CrossRef](#)]
26. Zidi, C.; Bouabdallah, F.; Boutaba, R.; Mehaoua, A. MC-UWMAC: A multi-channel MAC protocol for underwater sensor networks. In Proceedings of the 2017 International Conference on Wireless Networks and Mobile Communications, Rome, Italy, 9–11 October 2017.

27. Chao, C.M.; Lu, M.W.; Lin, Y.C. Energy-efficient multichannel MAC protocol design for bursty data traffic in underwater sensor networks. *IEEE J. Ocean. Eng.* **2015**, *40*, 269–276. [[CrossRef](#)]
28. Xie, P.; Zhou, Z.; Peng, Z.; Yan, H.; Hu, T.; Cui, J.; Shi, Z.; Fei, Y.; Zhou, S. Aqua-Sim: An NS-2 based simulator for underwater sensor networks. In Proceedings of the MTS/IEEE Conference on Oceans, Biloxi, MI, USA, 26–29 October 2009.
29. Kim, H.; Im, T.H.; Cho, H. UCMAC: A cooperative MAC protocol for underwater wireless sensor networks. *Sensors* **2018**, *18*, 1969. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).