A Practical TDMA Protocol for Underwater Acoustic Networks Based on Relative Clock

Jiarong Zhang, Can Wang, and Gang Qiao
National Laboratory of Underwater Acoustic Technology, Harbin Engineering University, Harbin, Heilongjiang Province, P.R. China

Abstract - This paper proposes a practical TDMA protocol for underwater acoustic networks which are characterized by long propagation delays and limited energy. The protocol saves transmission energy by avoiding collisions while maximizing throughput. It is based on the relative clock by mapping the network schedule to every node’s local clock and taking local clock to arrange work schedule. Node synchronization is not needed and clock offset is modified during data transmitting. This protocol achieves a throughput several times higher than that of the CSMA, while offering similar savings in energy. Although CS-ALOHA offers a similar throughput, it wastes much more power on collisions.

Keywords: underwater acoustic networks; relative clock; TDMA

1 Introduction

As electromagnetic waves propagate poorly in sea water, acoustics provides the most obvious medium to enable underwater communications [1-3]. Underwater acoustic networks (UAN) have great application values in marine research, natural disaster warning, underwater vehicles navigation, offshore defense and et al [4]. Underwater acoustic channel is challenging due to limited bandwidth, extended multipath, refractive properties of the medium, severe fading, rapid time-variation and large Doppler shifts [5-6]. Communication techniques originally developed for terrestrial wired and wireless channels need significant modifications to suit underwater channels [7-9]. Even worse, acoustic modem, key facility for UAN is energy limited and hard to recharge. Energy efficient protocols are critical needed for prolonging the UAN lifetime. Although CSMA (Carrier Sense Multiple Access) can reduce the collision energy waste as much as possible by hand-shaking mechanism, it also increases end-to-end delay during hand-shaking procedures since the propagation delay is very long. TDMA (Time Division Multiple Access) assign fixed time slot to every node to avoid collision, as well as reduce time wasting in channel reservation. But, in originally developed TDMA, strict clock synchronization is needed, while this is next to impossible in UAN. This fact has been recognized recently by several authors, and some possible TDMA protocols for underwater acoustic channel are proposed [10-11], but it needs the synchronization signal to be broadcasted periodically.

Here, we propose under the name of Relative Clock based Time Division Multiple Access (RC-TDMA) a practical protocol for centralized UAN. In this protocol, clock synchronization is not needed, network schedule is mapped to the node’s local clock line and its individual work schedule is arranged according to signal arrival time and received network schedule, clock offset caused by quartz accuracy, position change or some other reasons will be modified during the data transmitting course.

The rest of this paper is organized as follows: Section 2 we detail the design of our RC-TDMA protocol. The performance of RC-TDMA is studied via extensive simulations in Section 3. In Section 4, we conducted a Lake Trail to validate the practicality of protocol. We conclude with directions for future work in Section 5.

2 Protocol description

Network topologies of UAN are usually divided into two categories: centralized and distributed. In centralized UAN, TDMA protocol is an ideal choice for avoiding collision and shortening end-to-end delay. The RC-TDMA protocol we proposed in this paper is based on the centralized topology, which can be illustrated as Fig.1 shows, consists of one gateway node (O) and some other sub-nodes (such as A, B, C), positions of these nodes are fixed.

Corresponding author: Gang Qiao, qiaogang@hrbeu.edu.cn.
The basic considerations of RC-TDMA are using the local clock to arrange individual work schedule without any complex synchronization, make the nodes transmitting data in parallel according to their propagation delays to improve throughput, and modify the clock offset during data transmitting to avoiding collision. Work progress of RC-TDMA can be divided into three portions: schedule registration, schedule mapping and data transmitting, clock offset modifying.

2.1 Schedule registration

Schedule registration needs two steps: delay estimation and schedule arrangement. Propagation delay estimation is used to arrange schedule in staggered. We do not concern the propagation delay between the sub-nodes, only the gateway to sub-nodes propagation delays are needed. Fig.2 shows the main course of estimating the propagation delays between gateway and sub-nodes.

<table>
<thead>
<tr>
<th>O</th>
<th>RIS</th>
<th>CIS_A</th>
<th>CIS_B</th>
<th>CIS_C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>RIS</td>
<td>Δt_A</td>
<td>CIS_A</td>
<td>CIS_B</td>
</tr>
<tr>
<td>B</td>
<td>RIS</td>
<td>Δt_B</td>
<td>CIS_B</td>
<td>CIS_A</td>
</tr>
<tr>
<td>C</td>
<td>RIS</td>
<td>Δt_C</td>
<td>CIS_C</td>
<td>CIS_A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>Sent frame</td>
<td>□</td>
<td>Received frame</td>
<td></td>
</tr>
</tbody>
</table>

Fig.2 Flow chart of propagation delay estimation.

Let’s take gateway O and sub-node A to illustrating the propagation delay estimation procedure. On network initializing, O broadcasts the Require Initialize Signal (RIS) to A and records the send time $t_s$. After RIS arrived, A chooses a random interval time to send back the Clear Initialize Signal (CIS), which contains the interval $\Delta t_A$ from RIS arrived to CIS sent. Arrive time $t_{r_A}$ of CIS from A will be recorded by O at the moment of CIS arrived, and the propagation delay between gateway O and sub-node A can be depicted as:

$$T_{\text{rpd}_A} = t_{r_A} - t_s - \Delta t_A \quad (1)$$

Collision of the CIS may occur at the gateway, for this problem, we propose two solutions:

If the IDs of sub-nodes are not certain, we use multi-stage registration. In first stage, the gateway broadcasts RIS to all sub-nodes. As the first stage complete, gateway records the IDs of successful registered sub-nodes and the corresponding delays in a list. Then, starts the next stage registration with the successful IDs contained in RIS, the corresponding sub-nodes keep silence. After this stage complete, new successful IDs and corresponding delays will be add to the list. This course will be repeated until no sub-node responds the RIS, which considered as the end of initialization.

If the IDs of sub-nodes are confirmed, we use polling registration. Gateway sends RIS to sub-nodes one by one and adds the succeeded IDs and corresponding delays to list sequentially until the last sub-node has been requested.

After all propagation delays have been obtained, the gateway arranges the schedule in interleaved to make the sub-nodes working in parallel according to the delays. For example, the receive time-window on gateway for sub-node A and B are respectively $(0,10)$ s and $(10,20)$ s as Fig.3 shows, propagation delay of sub-node B is 10 s longer than that of B, thus A and B can transmit data simultaneously without collision.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>......</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>40</td>
</tr>
</tbody>
</table>

Fig.3 Receive time rectangle on gateway clock line.

Network schedule is as Fig.4 shows, each row filled with sub-node ID and its corresponding parameters including the working period $T$, working time-window $T_{wi}$, propagation delay $T_{pd}$ and sending time modified value $t_{mi}$. After schedule arrangement has been finished, the gateway dispatches it to all sub-nodes.

<table>
<thead>
<tr>
<th>ID</th>
<th>T</th>
<th>$T_{wi}$</th>
<th>$T_{pd}$</th>
<th>$t_{mi}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID_1</td>
<td>T</td>
<td>$T_{wi1}$</td>
<td>$T_{pd1}$</td>
<td>$t_{mi1}$</td>
</tr>
<tr>
<td>ID_2</td>
<td>T</td>
<td>$T_{wi2}$</td>
<td>$T_{pd2}$</td>
<td>$t_{mi2}$</td>
</tr>
<tr>
<td>ID_3</td>
<td>T</td>
<td>$T_{wi3}$</td>
<td>$T_{pd3}$</td>
<td>$t_{mi3}$</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>

Fig.4 Network schedule information.

2.2 Schedule mapping and data transmitting

On receiving the schedule, each sub-nodes mark the arrival moment as the clock line start point, and find its corresponding parameters from the schedule, then, the data sending time $t_{si}$ can be expressed as:

$$t_{si} = t_{ir} + t_{mi} + kT \quad (2)$$

Sub-nodes send data in their working windows, and keep a $T_{di}$ length monitoring for acknowledgement (ACK) from gateway. If AKC is received, next send time will be modified according to the information from ACK, otherwise, the sub-node go to sleep and wake up automatically at next working period. This course can be illustrated as Fig.5 shows.
2.3 Clock offset modifying

Clock offset may occur caused by quartz accuracy or node position change drifted by flows. This need to be modified immediately or data collision would occur at gateway. Receive time-windows for each sub-nodes are created at the schedule registration stage on gateway’s local clock line, guard interval \( T_g \) are also made between the adjacent time-windows allows the data arrival time has some offsets. If the offset exceeds the critical, ACK would be sent to the corresponding sub-node with the modified value contained. Fig.6 shows the receive time-windows setting.

\[
\begin{align*}
\text{data}_0 & \quad T_r \quad \text{data}_1 \\
& \quad T_r \quad \text{data}_2 \quad T_r \quad \text{data}_{s+1}
\end{align*}
\]

Fig.6 Receiving time-window setting on gateway’s clock line.

Now, we take \( \text{data}_i \) as an example to detail the clock offset modification. Let’s assume that the data duration is \( T_{\text{data}} \), and its arranged arrival time is \( t_0 \). \( T_s \) is the safe time that means if the data has been received completely in \( T_s \), collision will not occur. \( T_j \) can be expressed as:

\[
t_0 - \frac{1}{2} T_g < T_s < t_0 + T_{\text{data}} + \frac{1}{2} T_g
\]

If \( \text{data}_i \)’s arrival time \( t_n \) is in \((t_0+0.5T_g, t_0+0.5T_g)\), then collision will not occur, gateway sends nothing, otherwise, ACK will be sent to sub-node \( i \) including modifying value \( \Delta t_m \), which can be expressed as:

\[
\Delta t_m = t_n - t_0
\]

The sub-node \( i \) modifies its send time on receiving the ACK. If \( \Delta t_m > 0 \), that means the arrival time for \( \text{data}_i \) is lagged, and its next send time needs \( \Delta t_m \) earlier. In contrast, if \( \Delta t_m < 0 \), sub-node \( i \) needs leg its send time \( \Delta t_m \) later.

3 Performance analyzing and simulating

To assess the protocol performance, we define the end-to-end delay as the duration of a packet generated to receive successfully, and the throughput as the total successful received packet numbers per second.

3.1 Performance analyzing

The end-to-end delay can be expressed as:

\[
T_{e2e} = \frac{L_{\text{data}}}{R} + T_{p_d} + t_d + mT
\]

\[m = \left\lfloor \frac{N_{\text{load}}}{C} \right\rfloor\]

Where, \( L_{\text{data}}, L_{\text{ack}} \) means the frame length of data and ACK, \( R \) means the transmission rate, \( t_d \) means the waiting time for a sending course in a work-window, \( N_{\text{load}} \) means the net load of UAN and \( C \) means the sending capacity in a work-window, \( T \) and \( T_{p_d} \) has the same meaning as defined before. \( \lfloor \cdot \rfloor \) means rounding a number to its nearest integer. As equation (5) shows, end-to-end delay \( (T_{e2e}) \) for RC-TDMA relates to the net load, work-window and sending capacity in each work-windows.

Throughput can be expressed as:

\[
N_{\text{thrt}} = \frac{[T - (N - 1)T_g] \cdot R \cdot (1 - p_n)}{L_{\text{data}} \cdot T}
\]

Where, \( N \) is the sub-node numbers, \( p_n \) is the packets missing rate, \( T, R, T_g \) has the same meaning as defined before. The throughput will be affected by the guard interval length \( T_g \) and the packets missing rate \( p_n \) which depends on the channel conditions.

Now, we investigate the energy consumption of the RC-TDMA protocol for transmitting a packet. Assume that the modem has three states: sending, receiving and sleeping. As the energy consumption in sleeping state is extremely lower than the other two, we just consider the send and receive energy consumption. For one packet transmitting, include sending and receiving, energy consumption can be expressed as:

\[
E = \left( P_{\text{snd}} T_{\text{snd}} + P_{\text{rcv}} T_{\text{rcv}} \right)
\]

\[
T_{\text{snd}} = T_{\text{rcv}} = \frac{L_{\text{data}} + L_{\text{ack}}}{R}
\]

Where, \( P_{\text{snd}} \) and \( P_{\text{rcv}} \) is the send power and receive power. In RC-TDMA protocol, there is neither hand-shaking energy wasting as in CSMA, nor collision energy wasting as in ALOHA.

3.2 Simulation results

Simulations were run in OMNeT++ 4.0, gateway was deployed in the center of the simulation region and the sub-
nodes are distributed around the gateway randomly, other parameters are shown in Tab.1.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>simulation region</td>
<td>1.5 km × 1.5 km</td>
</tr>
<tr>
<td>gateway numbers</td>
<td>1</td>
</tr>
<tr>
<td>sub-node numbers</td>
<td>10</td>
</tr>
<tr>
<td>max transmitting range</td>
<td>1 km</td>
</tr>
<tr>
<td>send power</td>
<td>80 mw</td>
</tr>
<tr>
<td>receive power</td>
<td>10 mw</td>
</tr>
<tr>
<td>control packet length</td>
<td>64 bits</td>
</tr>
<tr>
<td>data packet length</td>
<td>512 bits</td>
</tr>
<tr>
<td>transmission rate</td>
<td>4800 bps</td>
</tr>
</tbody>
</table>

The performance of the protocol was compared to that of CSMA, including Carrier Sensing ALOHA (CS-ALOHA) as a benchmark. In CS-ALOHA, nodes transmit packets whenever they see the channel idle, and therefore do not waste time on hand-shaking. CSMA is a previously proposed protocol for the underwater environment, based on RTS/CTS hand-shaking, nodes use RTS/CTS to reserve the channel to transmit data.

Fig.7 illustrates the results of \( T_{e2e} \) for these three protocols. As we can see, RC-TDMA delay depends on the \( T \) as equation (5) shows. In low net load, shorter \( T \) gets shorter \( T_{e2e} \), while it is in contrast in high net load. Select a propel \( T \) can make the \( T_{e2e} \) of RC-TDMA shorter than CSMA and CS-ALOHA in any net load.

4 Lake Trail results

In order to validate the practicability of this protocol, we conducted a Lake Trail in April 2013 at Qiandao Lake in Zhejiang province, China. Experiment region and nodes distribution schematic is shown in Fig.10, as well as the nodes pictures we used in this experiment. Depth of the experiment region is about 30-50 meters, the gateway is deployed near the ship and the 5 sub-nodes were randomly deployed around the gateway in range of 0.5-3 km. Nodes were deployed about 20 m below the surface with cement block and float. We used FSK to transmit control packets and OFDM to transmit data packets, frequency band is in 6-10
kHz. In schedule registration procedure we use polling registration, it cost about 5 minutes to complete the initialization, then the network start working automatically.

Fig.10 Lake Trail schematic and modem pictures.

The actual distance (obtained from GPS) of each sub-nodes to gateway and the measured propagation delay (including the modulate/demodulate time) are shown in Tab.2, as well as the \( t_{mi} \) (arranged by gateway while \( T=60, T_p=2 \)).

<table>
<thead>
<tr>
<th>node</th>
<th>Distance (km)</th>
<th>( T_{e2e} ) (s)</th>
<th>( t_{mi} ) (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.56</td>
<td>4.61</td>
<td>7.39</td>
</tr>
<tr>
<td>B</td>
<td>0.65</td>
<td>3.99</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>2.51</td>
<td>5.23</td>
<td>30.77</td>
</tr>
<tr>
<td>D</td>
<td>2.89</td>
<td>5.48</td>
<td>42.52</td>
</tr>
<tr>
<td>E</td>
<td>2.01</td>
<td>4.92</td>
<td>19.08</td>
</tr>
</tbody>
</table>

The test was last almost 8 hours, statistics of each sub-node sending packets, modifying times and the gateway successfully received packets for each sub-nodes are shown in Tab.3.

<table>
<thead>
<tr>
<th>node</th>
<th>Snd(pkts)</th>
<th>Rcv(pkts)</th>
<th>Modify(times)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1232</td>
<td>1205</td>
<td>109</td>
</tr>
<tr>
<td>B</td>
<td>1179</td>
<td>1130</td>
<td>113</td>
</tr>
<tr>
<td>C</td>
<td>1355</td>
<td>1341</td>
<td>98</td>
</tr>
<tr>
<td>D</td>
<td>1530</td>
<td>1525</td>
<td>76</td>
</tr>
<tr>
<td>E</td>
<td>1425</td>
<td>1403</td>
<td>89</td>
</tr>
</tbody>
</table>

Lake Trail results are well demonstrated that the protocol is practical and easy to be implemented, although no clock synchronize was made and no synchronization signal was periodically broadcasted, the network can work well without collision and modify the clock offset automatically during data transmitting.

5 Conclusions

In this paper, we proposed a practical TDMA protocol for UAN based on relative clock, this protocol do not need neither clock synchronization nor synchronize signal broadcasting periodically. It maps the network schedule to local clock line, and arrange individual work schedule according the schedule arrival time and its corresponding information in the network schedule. Gateway makes the sub-nodes work in parallel according their propagation delays and fixes the clock offset during data transmission. Simulation results show that it can achieve a throughput several times higher than that of the CSMA, while offering similar savings in energy. Although CS-ALOHA offers a similar throughput, it wastes much more power on collisions. The Lake Trail well validate that the proposed protocol is practical and easy to be implemented.

In future work, we will investigate the problem of old node missing or new node incoming, as well as its application in distributed network.

6 Acknowledgement

This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation, and is supported in part by the National Natural Science Foundation of China under Grants No. 11274079, and the National High Technology Research and Development Program of China under Grant No. 2009 AA093601-2.

7 References


