Enhancement of Receiver-initiated Packet Train Protocol with Slotted Random Access for Underwater Acoustic Networks

Nuttarit Leelapisut[1], Muhammad Saadi[2],
Lunchakorn Wuttisittikulkij[3]
Department of Electrical Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand
Leelapisut.n@gmail.com[1], muhammadsaadi@gmail.com[2],
Lunchakorn.W@chula.ac.th[3]

Nitthita Chirdchoo
Sensor Network and Embedded System Research Unit
Nakhon Pathom Rajabhat University
Nakhon Pathom, Thailand
nitthita@npru.ac.th

***Abstract*— Underwater media access control (MAC) protocols based on receiver-initiated handshaking are more effective in terms of alleviating the hidden terminal problems than sender-initiated one's. Receiver-Initiated Packet Train (RIPT) is one such protocol that was designed to allow multi-receiving packets from multiple neighboring nodes in one handshake round to achieve high and stable throughput. However, RIPT requires that every node must know every other inter-node's propagation delays. Moreover, the frame size for data transmission period is set based on the information of the past traffic demand, not taking into account of the present demand.**

This paper aims to enhance the performance of RIPT by introducing slotted random access during the reservation period and applying adaptive frame size in response to present traffic demand. Our simulation results have confirmed that the enhanced RIPT protocol can improve throughput of the original RIPT in all load and decease delay when the load is low while keeping the delay unchanged at high load.

Keywords*— *Underwater acoustic communication; Access protocols; Multi access communication

I. INTRODUCTION

Underwater Acoustic Networks (UANs) have many applications such as pollution monitoring, offshore exploration, and oceanographic data collection. However, scarce bandwidth availability leads to low data rate in addition to low propagation speed (approximately 1500 m/s) results in longer propagation delay. These unique characteristics of underwater acoustic channel differentiate them from radio communication channel used in terrestrial wireless networks [1]. For these aforementioned characteristics, well established MAC protocols suitable for wired/wireless networks cannot be applied directly to UANs. In response to this, Chirdchoo et al. proposed a MAC protocol designed specifically for underwater network called RIPT, which outperforms the conventional MACA [4].

RIPT is essentially a receiver-initiated 4-way handshake protocol, which is conceptually distinct to other known sender-initiated MAC protocols for UANs, such as SF-MAC [3], MACA-U [5]. An important feature of such a protocol is the ability to increase the spatial reuse of UANs [6] and to alleviate the hidden terminal problem. A brief description of RIPT is as follows:

- A prospective receiver broadcasts an RTR packet to inform all neighboring nodes which can possibly act as a receiver.
- Each neighboring node responds to RTR packet by sending a SIZE packet containing the number of packets that it wants to send to the receiver.
- The receiver sends both time for data packet transmission and the number of packets allowed to send to the receiver to each corresponding neighboring node in the ORDER packet.
- Finally, the respective neighboring nodes transmit their DATA packets.

However, RIPT protocol requires that each node must know the inter-node's propagation delay of all other nodes, so that the packet transmission time of each node can be determined to avoid overlapping of packets at the receiver. In addition, the RIPT protocol can adapt the frame size of data transmission period according to the number of data packets from the previous handshaking round, it does not take into account of the current traffic demands. To overcome the aforementioned weaknesses in RIPT, we propose an enhanced RIPT protocol (E-RIPT) by using the slotted random access protocol at the reservation time to decrease the requirement of knowing all propagation delays of every inter-node. Also, we set the frame size during DATA packet transmission time after received traffic demand from neighboring nodes to improve the RIPT performance.

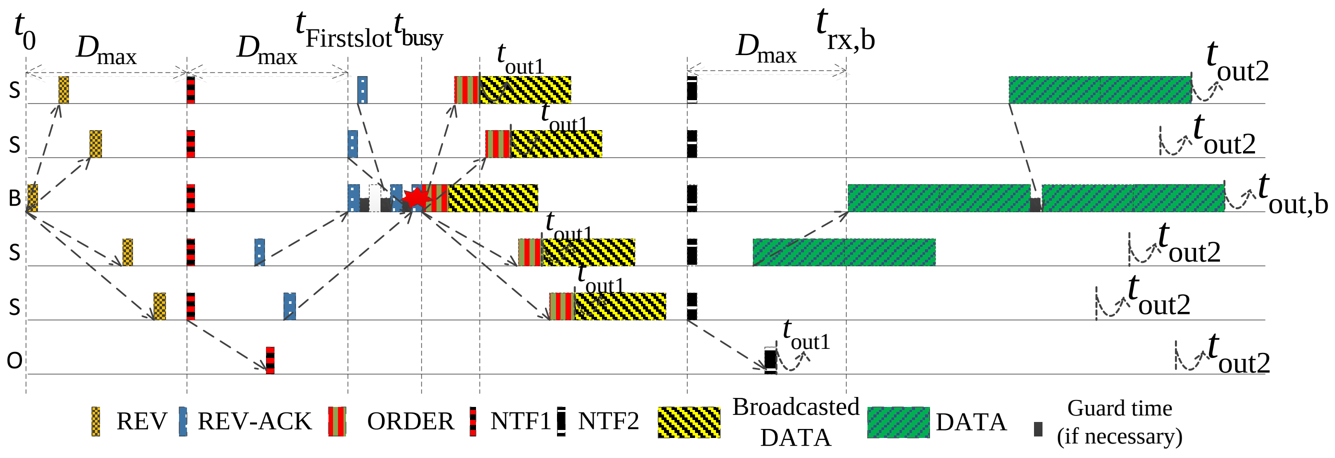


Fig. 1. Timing diagram of E-RIPT. Nodes "B", "S" and "O" refer to Beacon node, Slave node and The second-hop neighbors from beacon.

The rest of this paper is organized as follows. In Section 2, we describe the E-RIPT protocol in detail. Next, we compare the performance of E-RIPT with the original RIPT in Section 3. Finally, in Section 4, we give our conclusions and further work

II. E-RIPT PROTOCOL DESIGN

The timing diagram of E-RIPT is shown in Fig. 1 Details of the proposed algorithm is described in the following 5 steps.

1) Protocol initiation by REV packet: when an idle node wishes to become a receiver, it initiates a handshake by broadcasting an REV packet. (In order to avoid any confusion, we shall clarify that the terms "Beacon" refers to the receiver and "Slave" refers to its immediate neighbors that have packets to send to Beacon). The REV packet contains.

- The initiating beacon's node ID.
- The number of reservation slots, S_{all} which each slave can select to send reservation packet called REV-ACK.
- Time-stamp to calculate propagation delay from local time synchronization, if necessary.

2) Notification REV packet (NTF1) and slot reservation using the REV-ACK packet: when a neighboring node hears the REV packet, it will change its state to slave node. After becoming slave node, it will broadcast NTF1 packet to second-hop neighbors from beacon. They will be silent till the timeout. The broadcast time (NTF1) of all slaves will be the same which is the maximum propagation delay, $D_{\max}$ from beacon.

Next, each slave will check its own buffer. If it has packet that wish to send to beacon, it will respond to the beacon with a REV-ACK packet at the time slot which will be selected at random from S_{all}.

The slave(s) and beacon compute the busy duration, t_{busy} from S_{all} locally as follows:

$$t_{\text{FirstSlot}} = t_0 + T_{\text{REV}} + 2D_{\max} + T_{\text{NTF1}} + T_{\text{guard}} \tag{1}$$

$$t_{\text{busy}} = t_{\text{FirstSlot}} + S_{\text{all}} \times (T_{\text{rev-ack}} + T_{\text{guard}}) \tag{2}$$

where $t_{\text{FirstSlot}}$ is the first time slot that slave node can choose to respond against REV-ACK packet. T_x is transmission time of each fixed-length control packet of type x, where $x \in \{\text{REV,REV-ACK,ORDER,NTF1,NTF2}\}$. T_{guard} is a small guard time that can be inserted to protect against any estimation error in the inter-node propagation delay which is very small compared to DATA packet's transmission time.

The REV-ACK packet serves two purposes. The first one is to inform the beacon about the number of relay and new DATA packets to transmit. The second one is to silent the second-hop neighbors from beacon with the same as NTF1 packet does.

3) Transmission order broadcast through the ORDER packet: after reservation time, the beacon will allocate its DATA slots using a simple strategy as described in RIPT protocol [4]. After the Beacon processes REV-ACK packets, it will broadcast an ORDER packet. The ORDER packet contains the time slots for each slave to send its DATA packet and the total number of data packet(s) which is allowed to send to beacon. The total numbers of DATA packet which beacon can allocate to all slave(s) have a limit to avoid long utilization of channel by a specific node. Immediately after transmitting the ORDER packet, beacon will transmit its BROADCAST packet, if any.

The time that Beacon will start receiving packets from slave(s), $t_{\text{rx,b}}$ is

$$t_{\text{rx,b}} = t_{\text{busy}} + 2D_{\max} + T_{\text{ORDER}} + T_{\text{NTF2}} + T_{\text{BROADCAST}} \tag{3}$$

4) DATA Train transmission and timeout round: when slave receives the ORDER packet, it will set new timeout (t_{out2}). Furthermore NTF2 packet is broadcast at the same time to set the new timeout (t_{out2}) of second-hop neighbors from beacon. If a slave has been allocated at least one DATA slot, it must compute the time at which it will start its DATA transmission which given by ORDER packet.

The slave that fails to send REV-ACK (due to collision at beacon, as shown in Fig. 1 in slot 4) will not be allowed to send DATA packet(s) to beacon.

The timeout of beacon, $t_{\text{out,b}}$ will occur when all DATA packets are received. Slave timeout doesn't need to be as long as beacon timeout but long enough to allow the beacon to finish receiving all DATA packets in the current handshaking loop. Timeout considering the propagation delay is given by:

$$t_{out2,x} = t_{out,b} - D_x \tag{4}$$

where $t_{out2,x}$ is t_{out2} of node x and D_x is the propagation delay between beacon and node x.

Second-hop neighbor nodes from beacon must avoid initiating an RTR handshake until a certain timeout, and that timeout is:

$$t_{out2,y} = t_{out2,x} + D_{x,y} \tag{5}$$

Where $D_{x,y}$ is the propagation delay between node x and node y.

5) When to initiate an REV packet? we simply pick the exponential distribution for the time between RTR-initiations with an exponential mean of T_{avg} for node to initiate an REV packet. After the timeout, all nodes will be in idle state. In order to avoid the same node to successively become a beacon node, it has to wait for t_{limit} (where t_{limit} corresponds to time after release from handshaking or has served as slave in any subsequent handshaking loop before it can initiate a new REV packet).

III. SIMULATION AND RESULTS

A. Simulation Model

For simulation, we used open source simulator NS-3 [8], with UAN module. Our simulation model is the same as RIPT simulation model [4]. The transmission data rate is 2400 bps. The acoustic propagation speed is assumed to be 1500 m/s. The grid spacing is 700m. The channel is also assumed to be error-free, thus all packets loss are caused by packet collisions. We also do not implement ACK for any of the schemes thus there is no retransmission for lost packets. The length of the REV, REV-ACK, NTF1, NTF2, ORDER packets is 48, 72, 56, 56, 184 bits respectively while DATA packets are 2400-bit long. Maximum number of all allowable DATA packets which neighboring nodes will send to the receiver per handshaking loop is 10 packets to avoid any node capturing the channel for too long. The buffer size for both new packets and relayed packets are set to 100 each. We choose to benchmark our protocol with RIPT [4] and MACA-U [5]. We set the control packet length of RTR, SIZE, ORDER to 40, 56, 184 bits for RIPT and RTS, CTS to 40 bits for MACA-U while keeping all other parameters the same.

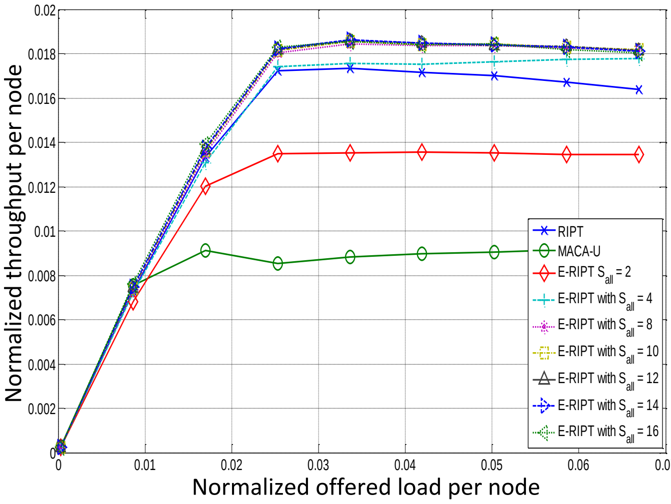


Fig. 2. Comparing Throughput of RIPT and E-RIPT with difference number reservation slots S_{all}

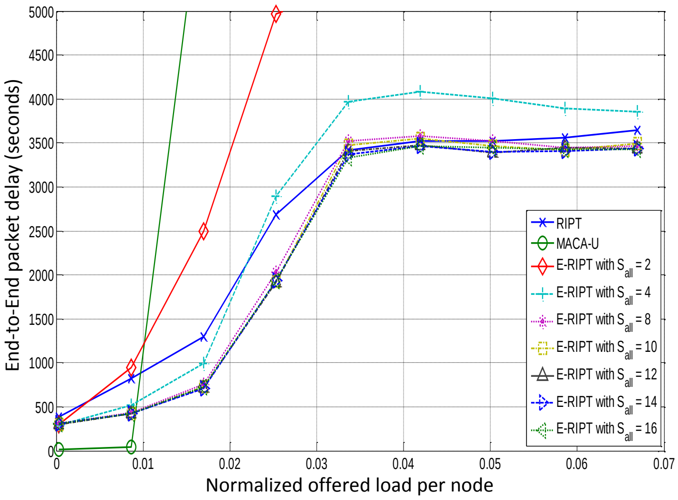


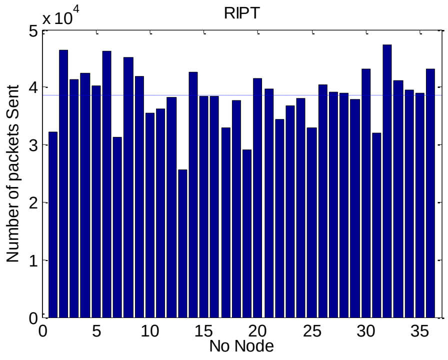
Fig. 3. Comparing End-to-End packet delay of RIPT and E-RIPT with difference number reservation slots S_{all}

B. Simulation Results

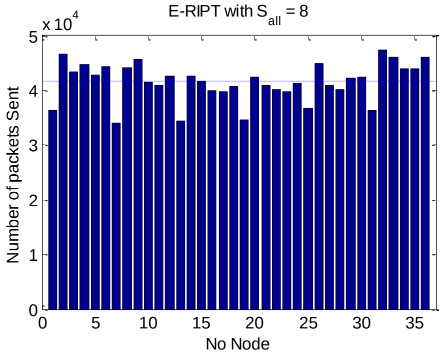
The simulation duration for each data point was $1\text{x}10^6$s. The simulation results are collected after $2\text{x}10^5$ seconds to avoid transient effect. Here we define "throughput per node" as the average throughput of 36 nodes as follows

$$\text{Throughput per node} = \frac{1}{36}\left[\frac{\text{No. of Packets Received/Simulation Time}}{\text{Data Rate/Packet Length}}\right]$$

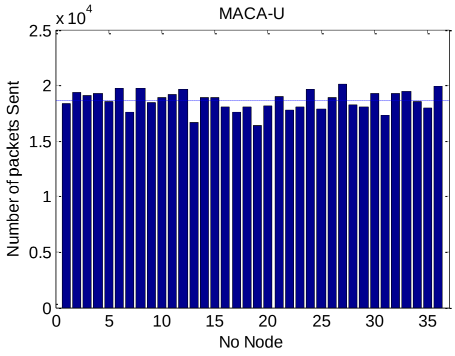
As shown in Fig. 2, throughput of each protocol become stable when load is greater than 0.34 which implies that the channel is saturated. The performance of the E-RIPT outperforms MACA-U in-terms of throughput. If E-RIPT has more than 4 reservation slots, throughput will be better than RIPT. Throughput of MACA-U protocol is quite low when compared with E-RIPT and RIPT protocols because MACA-U transmits a single data packet per round of handshake, so the throughput will suffer from under-utilization of channel due to long propagation delay in Underwater Acoustic Channel. From



(a)



(b)



(c)

Fig. 4a. Number of packets sent of RIPT protocol, normailzed offerload per node = 0.503, Mean = 38528.1389, SD = 4922.9368

Fig. 4b. Number of packets sent of E-RIPT protocol with S_{all} = 8, normailzed offerload per node = 0.503, Mean = 41608.8056, SD = 3470.5466

Fig. 4c. Number of packets sent of MACA-U protocol, normailzed offerload per node = 0.503, Mean = 18598.6944, SD = 906.4938

the study of original RIPT protocol, RIPT can transmit multi data packets per round of handshake by forming a packet train so it can improves channel utilization. The length of packet train in each round adapts by using knowledge from the previous round, so it does not take into account of the current traffic demands which throughput lower than E-RIPT. E-RIPT use the packet train size of the current demand by number of received packet(s) from slave(s) before selecting a packet train size. But if E-RIPT has low reservation slots, the REV-ACK packets will collide due to choice of the same reservation slot so that the beacon cannot receive the information from slave which will cause low throughput.

In term of delay, Fig. 3, shows that at very low load (below 0.01) MACA-U seems to have the best delay performance because the packet train tends to be very short when load is low and the sender-initiated handshake has overhead is less than receiver-initiated handshake. However, beyond a load 0.01, the packet train is long enough to overcome the overhead of receiver-initiated handshake, so RIPT and E-RIPT has better performance in term of delay than MACA-U.

From Fig. 3, it can also be observed that E-RIPT with more than 8 reservation slots will outperform RIPT in terms of delay before the channel has been saturated (load below 0.34). Due to E-RIPT chooses the packet train for current demand better than RIPT that adapt the packet train by using information from previous handshake. After the channel is saturated, delay of E-RIPT and RIPT become the same because the number of packet which are allowed to transmit will be at the limit and this two protocols. We set limit as the same number of these 2 protocols for this simulation.

In term of fairness when the channel is saturated, we found that E-RIPT with 8 reservation slots has more fairness than RIPT due to standard deviation of number of packet that each node sent by E-RIPT protocol is less than RIPT. Furthermore, the mean of transmitting packets from each node of E-RIPT is greater than RIPT. For E-RIPT, each node optimizes time in each handshake loop so it makes more opportunity to act as a beacon compare to the original RIPT. MACA-U has the best fairness because each node capture the channel with less times than E-RIPT and RIPT. Fig. 4 explains the above mentioned process.

IV. CONCLUSION

In this paper, we proposed and evaluated a MAC protocol for multi-hop underwater acoustic network which uses receiver-initiated handshaking with slotted random access at reservation time. Our proposed protocol decreases the time latency, improve throughput and fairness of original RIPT protocol. Our results suggest that if reservation slots are selected carefully, then our proposed E-RIPT protocol can outperform RIPT in terms of throughput, and delay. E-RIPT is more suitable for the case where it is difficult to predict the traffic load. Our further work will be focused on refining the optimal settings for E-RIPT protocols and the conditions conductive to their efficient operation.

REFERENCES

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," Ad Hoc Networks (Elsevier), 2005.

[2] Chen, K.; Ma, M.; Cheng, E.; Yuan, F.; Su, W., "A Survey on MAC Protocols for Underwater Wireless Sensor Networks," *Communications Surveys & Tutorials, IEEE* , vol.PP, no.99, pp.1-15,2014.

[3] Wen-Hwa Liao; Chih-Chien Huang, "SF-MAC: A Spatially Fair MAC Protocol for Underwater Acoustic Sensor Networks," *Sensors Journal, IEEE* , vol.12, no.6, pp.1686, 1694, June 2012.

[4] Chirdchoo, N.; Wee-Seng Soh; Kee Chua, "RIPT: A Receiver-Initiated Reservation-Based Protocol for Underwater Acoustic Networks," *Selected Areas in Communications, IEEE Journal on* , vol.26, no.9, pp.1744,1753, December 2008

[5] Hai-Heng Ng; Wee-Seng Soh; Motani, M., "MACA-U: A Media Access Protocol for Underwater Acoustic Networks," *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE* , vol., no., pp.1,5, Nov. 30 2008-Dec. 4 2008

[6] Yen-Da Chen; Shu-Sheng Liu; Chia-Ming Chang; Kuei-Ping Shih, "CS-MAC: A Channel Stealing MAC protocol for improving bandwidth utilization in underwater wireless acoustic networks," *OCEANS 2011* , vol., no., pp.1,5, 19-22 Sept. 2011

[7] Zhai, Weihong; Zhang, Jun; Gupta, Sindhu Hak, "Asynchronous reservation MAC protocol for underwater acoustic sensor network," *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on* , vol., no., pp.241,244, 20-21 Feb. 2014

[8] [Online]. Available: http://www.nsnam.org/