UD-TDMA: A Distributed TDMA Protocol for Underwater Acoustic Sensor Network

Zhengbao Li   Zhongwen Guo   Haipeng Qu   Feng Hong   Pengpeng Chen   Meng Yang
Department of Computer Science
College of Information Science and Engineering Ocean University of China
Qingdao, China
{lizhengbao, guozhw, quhaipeng, hongfeng, chenpengpeng}@ouc.edu.cn   yangmengzf@yahoo.com

***Abstract*—This paper presents a distributed and robust time slot scheduling algorithm, which is suitable for underwater acoustic sensor network (UASN). The information of nodes' 2-hop neighbors is needed to be collected and then be used to calculate nodes' initial time slot by a distributed algorithm. A maximal independent set is formed by the nodes which were assigned with the same initial slot. Some theorems were proved to reveal that in an interference graph the size of this maximal independent set is at least $\frac{1}{224}$ of the size of the maximum independent set of the nodes. The simulation compares UD-TDMA with other three MAC protocols . The results show that the proposed protocol is effective in the UASN with random deployment, especially in high-density underwater acoustic sensor network.**

I. Introduction

Many challenging technical issues are coexist in underwater acoustic sensor network (UASN) [1], such as high latency, low bandwidth and changeful environment and ubiquitous noise. One of the key methods used to improve the performance of UASN is to design medium access control (MAC) protocols, which call for approaches that are very different from the terrestrial counterparts [2]. The acoustic channel, however, is characterized by long propagation delays, large delay spreads, and frequency dependent fading. The design of a suitable MAC with high throughput as well as desired energy efficiency is still a work-in-progress. In past acoustic network deployments, FDMA can't be applied to the UASN because the band of acoustic used in the underwater environment is very narrow, moreover the existence of attenuation and multipath (e.g. 1998-1999 SeaWeb [3]). The problem with the CSMA/CA solution is that it exacerbates the end-to-end delays that are incurred, especially in underwater acoustic sensor network with a large number of nodes [4]. In addition, the protocol [5] used the RTC/CTS to avoid message conflict, which increases the energy consumption and improves end to end latency greatly.

Time Division Multiple Access (TDMA) is a good access technique for UASN. TDMA can deliver very good performance especially under high contention. In underwater acoustic sensor network, a packet's propagation delay generally exceeds its transmission delay, often by an order of magnitude. This is different in terrestrial networks. The time synchronization is the first problem to be resolved in the process of designing TDMA protocol. Because of the usefulness of time synchronization, numerous synchronization algorithms have been proposed. [6], [7], [8] Recently, MU-Sync [8] estimats both the clock skew and offset to solve the time synchronization problem in the underwater enviroment. So in this paper, we neglect the time synchronization problem. As opposed to synchronizing some nodes to operate during the same time period, there are two main challenges in design a TDMA protocol in UASN:(i)the number of necessary synchronized nodes and (ii) the time interval between slots assigned to a node. In this multi-hop network, each node has a neighborhood of varying size and the hidden terminal problem must be addressed. A subset of nodes are selected to transmit simultaneously without interference. Meanwhile, the time interval between slots assigned to a node should be found to improve the network throughput.

In this paper, we present a distributed TDMA scheduling for UASN, called UD-TDMA. In the process of slot assignment, a maximal independent set of the underwater sensor nodes is achieved by a distributed algorithm in each turn of slot assignment. UD-TDMA runs in $O(n)$, which n is the maximum size of a two-hop neighborhood in the UASN.

The rest of the paper is organized as follows. In section II, we discuss the network model and some assumptions. In section III, we give the details of our algorithm, present the new distributed TDMA algorithm. In section IV, we show some logical analysis results. We introduce the simulations of the algorithm and the simulation results in section V. The proposed protocol is available in underwater environment and performs better than other protocols, especially in high traffic load case. Finally, we make conclusion and future work in section VI.

II. Network Model

A. The Interference Graph

We study the problem of designing an energy efficiency MAC protocol in UASN. In this paper, an underwater acoustic sensor network is represented by a graph $G = (V, E)$, where V is the set of nodes, and E is the set of edges. In addition, we assume that the nodes are static or can be viewed as static during a reasonable period of time. An edge $e = (u, v)$ exists if and only if u and v are in V, meanwhile the two nodes can transmit to each other successfully. In the underwater acoustic sensor network, each node has a unique ID.

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

We define the transmission radius of a node as the radius of the sphere representing its transmission region. Similarly, we define the interference radius as the radius of the sphere representing the interference region of this node. Each node $v \in V$ has a transmission radius t_v and an interference radius r_v, where $0 < t_v \leqslant r_v$. In this paper, we define $t_v = r_v$.

The model of this paper is an interference graph. We assume an interference occurs if the transmission region of one node intersects with the interference region of another node. Let $D(v, r)$ denotes the sphere centered at v with radius r. Each node then defines two spheres: the transmission sphere $D(v, t_v)$ and the interference sphere $D(v, r_v)$. The set of nodes V defines two sets of spheres $T = \{D(v, t_v)|v \in V\}$ and $T = \{D(v, r_v)|v \in V\}$ in the three dimensional plane. Given two nodes u and v, define their intersection region $I(u, v)$ as follows:

$$I(u, v) = (D(u, t_u) \bigcap D(v, r_v)) \bigcup (D(u, r_u) \bigcap D(v, t_v))$$

See Figure 1 for an illustration of intersection regions.

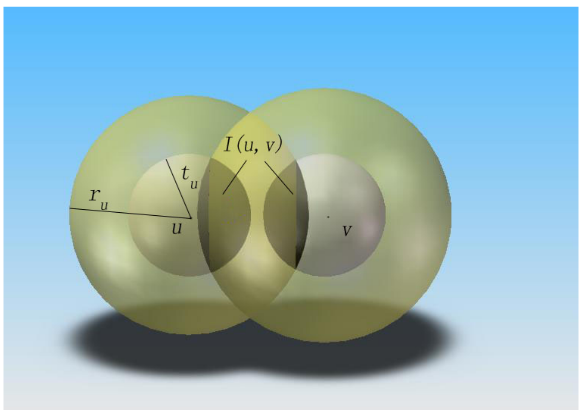


Fig. 1. The black region is the intersection region $I(u, v)$. The inner spheres are the transmission sphere and the outer spheres are the interference spheres.

In this paper, we assume that each node should collect the information of its 2-hop neighbors. For each node $u \in V$, Let $N_2(u)$ denote the set of its 2-hop neighbors. $N_2(u) = \{v|(u, v) \in V or\ ((u, r) \in V and\ (r, v) \in V)\}$.

B. Graph Problem and the Time Interval

A subset of vertices in graph G is an independent set (IS) if for any pair of vertices, there is no edge between them. It is a maximal independent set if no more vertices can be added to it to form an independent set. It's a maximum independent set(MIS) if no other independent set has more vertices [9].Note that the TDMA slot assignment problem is a direct extension of the graph coloring problem, where the goal is to color the vertices of a graph with minimum number of colors. A vertex coloring is valid if two adjacent vertices are assigned different colors. The graph coloring problem is known to be NP-Hard [10], so as the TDMA assignment. Hence, heuristic solutions often report the maximal independent set of vertices required to obtain a coloring assignment. In our TDMA slot assignment algorithm we call the number of the MIS the TDMA interval.

III. UD-TDMA ALGORITHM

In this section, we explain our distributed TDMA algorithm (UD-TDMA) for details. The distributed TDMA algorithm is based on the distributed maximal independent set algorithm. We can get the maximal number of nodes which can send message at the same slot time by a distributed maximal independent set algorithm. In the description, we assume the broadcast mode. At first we describe the initial TDMA slot assignment algorithm. Then, we present an algorithm used to obtain the time interval in subsequent slot assignment.

A. The Initial TDMA Slot Assignment Scheme

In our model, each node has an information record, which contains some information of itself and its 2-hop neighbors. The node u uses $R(u)$ to denote the recode and $R(u) = \{(ID_v, DG_v, t_v)|v \in N_2(u)\}$. ID_u is the ID of node u , $DG_u(=\| N_2(u) \|)$ is the degree of $N_2(u)$ and t_u is the initial time slot assigned to node u , $t_u = 0$, initially.

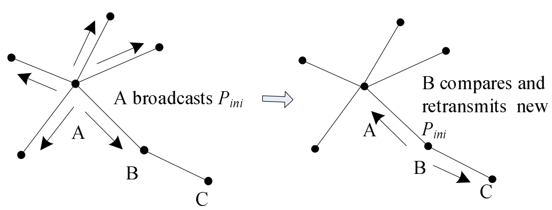
At first each node broadcasts a package (P_{ini}) including its own ID to its 1-hop neighbors either retransmits a P_{ini} received from its neighbors in order to collect the information of its 2-hop neighbors, and then works out the DG. For example, when a node A receives a P_{ini} from its neighbor B, it calculates the number of ID included in the P_{ini} . If there is only one ID(ID_1) in P_{ini} and the ID_1 isn't in $N_2(A)$, A updates its record and adds ID_1 into $N_2(A)$ and a vector $(ID_1, 0, 0)$ into $R(A)$. Then A adds its ID_A (denoted by ID_2) into the P_{ini} after ID_1, finally A retransmits the new P_{ini} to its neighbors. When there are two IDs denoted by ID_1, ID_2 included in the P_{ini}, A should consider whether ID_1 and ID_2 are in the list of its 2-hop neighborhood. The dealing of the absence of ID_1 has been mentioned above, a detail on the process of the absence of ID_2 is discussed in later. Similar to the process of ID_1, A still adds ID_2 into $N_2(A)$ and $R(A)$, and then A uses ID_2, ID_A to replace ID_1 and ID_2, respectively. A retransmits the new package to its 1-hop neighbors at last. In other cases, A drops the package P_{ini}.

If A does not receive any P_{ini} from its 1-hop neighbors within some time d_A, then it comes into the second step of the slot assignment. In this step, each node accounts its degree (DG) of N_2 and broadcasts a package N_{fon} including its DG and ID to its one-hop neighbors aiming to exchange the degree of 2-hop nodes. When A receives a N_{fon}, it uses a similar approach to deal with the package. At first, A calculates the number of DG involved in N_{fon}. While the N_{fon} has one DG (denoted by DG_1) and the DG_1 isn't in $R(A)$, A uses a vector $(ID_1, DG_1, 0)$ to replace the old one in $R(A)$, and then attaches its DG_A (denoted by DG_2) and ID_A (denoted by ID_2) into N_{fon} after DG_1. If there are two DG_2 in N_{fon} and DG_1 ist't in $R(A)$, then A adds it into $R(A)$. If DG_2 isn't in $R(A)$, A adds DG_2 into $R(A)$ and uses (DG_2, ID_2) to replace (DG_1, ID_1), and (DG_A, ID_A)

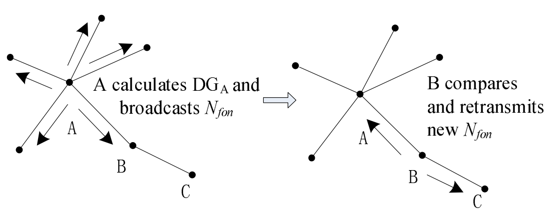
to replace (DG_2, ID_2),respectively. At last A retransmits the new N_{fon} . A discards the useless N_{fon} in other cases.

Each node compares its DG with the DGs of its 2-hop neighbors saved in its information record to calculate its initial time slot in step three. When A finds that all of its 2-hop neighbors have sent N_{fon}, it compares its DG_A with the DGs which is in $R(A)$ and decides the initial time slot assigned to itself. If DG_A is larger than any DGs in $R(A)$, that is to say A can send a message at slot $t_A = 0$, so a package (=$Slot_{ini}$) containing the initial time slot assigned to it is broadcasted to the neighbors of A. In other cases, A records the $Slot_{ini}$ sent by its neighbors, whose DGs are larger than DG_A. When those of the 2-hop nodes have sent their slots, we assume the max slot in $R(A)$ is t_{max}, the initial time slot of A is $t_A = t_{max} + 1$. In the status that some nodes have the same DG with A, they follow the decision sequence desided by their ID. The node with the largest ID can calculate its slot first. At last A broadcasts its slot to its 2-hop neighbors.

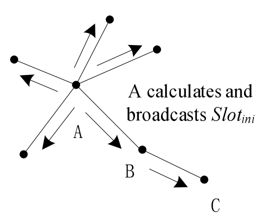
From the three steps above, each node is assigned an initial slot and has a table R recording its 2-hop neighbors' initial slot. Figure 2 illustrates a successful process of initial slot assignment.



(a) Step1



(b) Step2



(c) Step3

Fig. 2. The process of initial slot assignment scheme: (a) is step 1 of broadcasting P_{ini}, (b) is step 2 of boradcasting N_{fon} and (c) is step 3 of broadcasting $Slot_{ini}$

B. Time Interval

When each node knows its initial time slot, it should decide the next slot when it can send, that is the time interval. For example, if a sensor A has finished its $R(A)$, then it selects the maximal initial slot t_{max} in its $R(A)$, and let $T_{interval}$ denote the time interval between slots assigned to a sensor. The process is described as follows: First of all, $T_{interval}(A) = max(t_{max}, t_A) + 1$. A sends a package $Interval(A)$ with its $T_{interval}(A)$. When a node B receives the $Interval(A)$, it compares the $T_{intreval}(B)$ with the $T_{interval}(A)$ involved in $Interval(A)$, if $T_{interval}(B) < T_{interval}(A)$, B sets $T_{interval}(B) = T_{interval}(A)$ and retransmits the $Interval(A)$, else B drops the $Interval(A)$. When B never receive any $Interval$ package in a continuous time slots, that is to say that all the nodes in this network have the same time interval. B resets its clock to zero and decides to transmit message at slots assigned by the algorithm.

C. Energy Efficiency

Energy efficiency can be achieved in UD-TDMA as follows. A sensor remains in active mode only in its allotted time slots and in the allotted time slots of the sensors within its communication range. In the remaining slots, the sensor can save energy by turning off its radio and remaining in idle mode. Suppose the communication range of a sensor is 1, and a sensor have 6 neighbors. Let P be the period between successive time slots allotted to a sensor. It is clear that $P \geqslant 6$. A sensor will have to be in active mode in its allotted time slot and in the allotted time slot of its 6 neighbors, during every period.

IV. ALGORITHM PERFORMANCE ANALYSIS

In this section, we will show the performance of UD-TDMA algorithm. First we will prove that the sensors which have the same slot can't interfere with each other, and then we show the quality of the computed independent set.

Theorem 1. *Each node that assigned the same slot is independent with each other.*

Proof: In our network model, each sensor node has a transmission radius equaling to its interference radius. So if a sensor sends a message at slot t, all of its 2-hop neighbors shouldn't send any message simultaneously. In the UD-TDMA slot assignment algorithm, a node u has a record of its 2-hop neighbors' information. The slot assigned to u is different with any slots in its record $R(u)$. Meanwhile the initial slots in $R(u)$ are also different with each others. So we can easily conclude that in our initial slot assignment algorithm, each node assigned the same initial slot is independent. The nodes assigned the same initial slots form a maximal independent set. ■

Theorem 2. *The execution of UD-TDMA results in a conflict free TDMA schedule.*

Proof: We can see from theorem 1 that each round the nodes assigned the same slot compose a maximal independent set, meanwhile there is no two nodes within two hops of each other select the same time slot. The time interval algorithm ensures that all the nodes send successfully before sending the next round. This property ensures that the UD-TDMA results in a conflict free TDMA schedule. ■

Theorem 3. *The computed independent set in each round has size at least $\frac{1}{224}$of the MIS.*

Proof: We prove this using a volume argument. Consider any node u assigned in initial slot t. Let be the k nodes from an optimum solution that are 2-hop neighbors of u and excluded being assigned the initial slot t. The sphere D_i centered at v_i cannot contain any node $v_j(i = 1, 2, ..., k, j \neq i)$ inside because all spheres centered at $i = 1, 2, ..., k$ are independent in the interference graph model.

Nodes $v_i, i = 1, 2, ..., k$ are excluded in the initial slot t implies that D_i intersects with sphere D_u. Let V_u be the sphere centered at u with radius $2r_u$ and $M_u = V_u - D_u$. So every sphere D_i will intersect with M_u. It is not difficult to show that $M_u \bigcap D_i$ achieve the smallest volume when v_i is on the boundary of V_u. See Figure 3, Notice that the smallest volume is at least $\frac{13}{24}\pi$ and the volume of $M_u \bigcap D_i$ is $\frac{28}{3}\pi$.

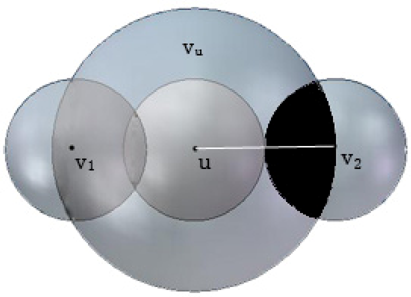


Fig. 3. The intersection graph: the black region is $M_u \bigcap D_{v_2}$

Notice that the volume $M_u \bigcap D_i$ and $M_u \bigcap D_j$ for $1 \leqslant i, j \leqslant k$ may overlap. However, we will show that every point a is covered by at most 13 spheres from $D_i, i = 1, 2, ..., k$. See Figure4. Assume node o is covered by sphere D_a, D_b and D_c. The Euclidean distance $OC = OA = OB = AB = BC = AC$. Thus, some nodes like A, B, C form a set UNI. All the nodes in the UNI link up to form a convex polyhedron. Next, we will find the number of the convex polyhedron vertexes. A set $IND$is formed by the nodes which are independent and their interference sphere D can cover o. The foundation of IND is smaller than UNI. We can see from Figure 5, the area embraced by A, B, C in the sphere is a spherical triangle. The area of the spherical triangle is $3arccos\frac{1}{3} - \pi$, the surface area of the sphere D_o is 4π. With the Euler's Theorem, we can find the relation between faces (f), vertices (v), and edges (e):

$$2e = 3f$$

$$v = 2 + \frac{f}{2}$$

while

$$f = 4\pi/(3arccos\frac{1}{3} - \pi)$$

so

$$v = 2 + 2\pi/(3arccos\frac{1}{3} - \pi) \doteq 13.4$$

.

Thus o is covered by at most 13 independent spheres. Therefore, by a volume argument, we have

$$k \cdot \frac{13}{24}\pi \leqslant 13 \cdot \frac{28}{3}\pi$$

Thus,

$$k \leqslant 224$$

Consequently, there are at most 224 independent nodes are removed when we remove all nodes adjacent to a node u selected by our algorithm. ■

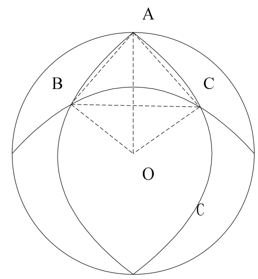


Fig. 4. The nodes A, B and C cover the node O.The angle between arcs AB and AC is the spherical angle $\widehat{ABC}$.

V. SIMULATION

So far our aim has been illustrated for the design process of the MAC layer by example. In this section we show the simulation and analysis.

A. Simulation Enviroment

We wrote our simulation program in Java and ran it on a computer with Ubuntu operation system. Another part of our simulation is run on NS2. We considered a random deployment underwater acoustic sensor network modeled with the size of $30 \times 30 \times 30$. We varied the size of sensor nodes to get different density. The number of nodes, n, was varied from 50 to 100, in increment of 10. The transmission range is set to 1 unit, so as the interference range. The S-Aloha [11] protocol was used in our simulations and the fractions=1. The USS-TDMA [12] is deployed in the random UASN.

B. Impact of Varying Size of Neighbors

We now look at the impact of varying average size of neighborhood. As can be shown in Figure 5, the size of neighborhood and the maximal initial slot assigned to nodes have a direct relationship. When a node has much more neighbor nodes, more initial slot need to be assigned to those nodes. The upward trend is in accordance with the actual, but the relationship didn't set up in some cases. The exact cause of this behavior is currently being investigated.

C. Comparison with Existing Algorithm

We now compare UD-TDMA with CSMA, S-Aloha and USS-TDMA.

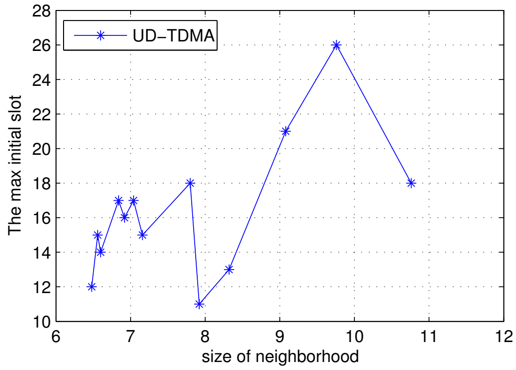


Fig. 5. the relationship between the sizes of neighborhood and the maximal initial slot

1) Throughput: We can see from Figure 6, when the offered load is below 1.3, CSMA has the best performance; S-Aloha has the second wonderful throughput. As the offered load increases, CSMA and S-Aloha have a decreased tendency. When the offered load is up to 1.5, UD-TDMA and USS-TDMA have a better performance than the other two protocols. Because the channel interference and conflict impact CSMA and S-Aloha more obviously than the two TDMA protocols. Meanwhile, we can see that UD-TDMA has a 25% higher throughput than USS-TDMA in random underwater acoustic sensor network. This is due to nodes with USS-TDMA protocols in random UASN encounter much more conflict than that in cube-based UASN. After the offered load is larger than 3, the UD-TDMA can have a stationary throughput

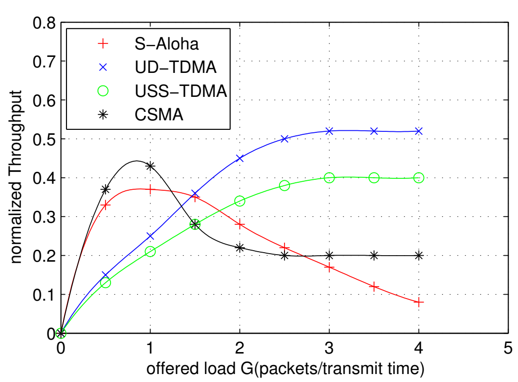
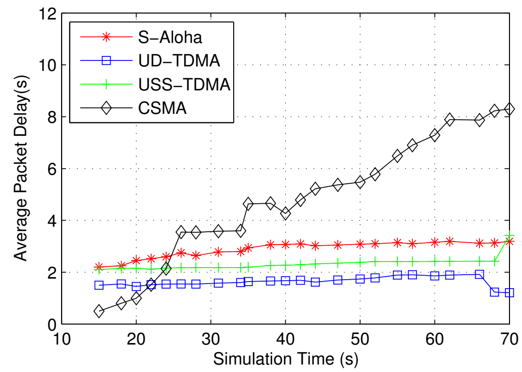
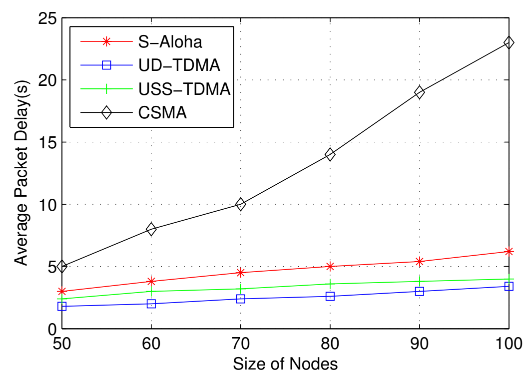


Fig. 6. the throughput of the four MAC protocols

2) Delay: Figure 7(a-b) depict the average packet delay in the simulation transient state when sensors are randomly deployed. The proposed UD-TDMA protocol version outperform the other MAC schemes in terms of both delay, especially the S-Aloha protocol (one order of magnitude), although the extremely harsh scenario leads to delays in the order of seconds for all the MAC schemes. Figure 7(b) shows the overall performance of the four MAC protocols when the number of deployed sensors increases. We can conclude that both UD-TDMA and USS-TDMA have a much smaller average packet delay than the other two schemes. The excellent performance of our scheme in the underwater scenario mainly because of the higher channel reuse achieved. When the number of sensors increases, the sensors with routing algorithm have a higher flexibility in the choice of data paths, which rely more on multi-hop communications, thus increasing their average number of hops. While at the routing layer this decreases the expected end-to-end energy to forward packets, higher interference is generated at the MAC layer.



(a)



(b)

Fig. 7. the comparison of transmission delay. (a): average packet delay vs. simulation time(50 nodes). (b): average packet delay vs. size of nodes.

VI. Conclusion

We introduce UD-TDMA, a distributed and robust TDMA algorithm for underwater acoustic sensor network. UD-TDMA is ideal for UASN with limited mobility. Each node in UASN just needs to collect its local information and then calculate its initial slot with a distributed algorithm. The set of nodes that assigned with the same initial slot form a maximal independent set. This algorithm can improve the utilization of the bandwidth. Compared to existing schemes, we can see from our simulation results that UD-TDMA gives more efficient slot assignments which result in better channel utilization while the network has a high-density nodes deployment. The direction for future research in this area includes the development of a distributed MAC scheduling shceme that could adapt to long, unknown propagation delays in the medium and the sensor nodes have a random interference radius.

ACKNOWLEDGMENT

This work was supported by the Chinese National Science Foundation of China under Grant No.60873248 and the National Basic Research Program of China (973 Program) under grant No.2006CB303000.

REFERENCES

[1] L. Badia, M. Mastrogiovanni, C. Petrioli, S.Stefanakos, and M. Zorzi. An Optimization Framework for Joint Sensor Deployment, Link Scheduling and Routing in Underwater Sensor Networks. SIGNOBILE 2007, 11(4):44-56, Oct 2007.

[2] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. State-of-the-Art in Protocol Research for Underwater Acoustic Sensor Networks. In Proc. ACM WUWNet'06, pp. 7-16, Sep. 2006.

[3] Rice, B. Creber, C. Fletcher, P. Baxley, K. Rogers, K. McDonald,D. Rees, M. Wolf, S. Merriam, R. Mehio, J. Proakis, K. Scussel,D. Porta, J. Baker, J. Hardiman, and D. Green, "Evolution of seawebunderwater acoustic networking," in Proc. of OCEANS 2000 MTS/IEEE Conf. and Exhibition, vol. 3, Sept. 2000, pp. 2007-2017.

[4] Volkan Rodoplu, Min Kyoung Park,"An Energy-Efficient MAC Protocol for Underwater Wireless Acoustic Networks", in Proc. Of OCEANS 2005 MTS/IEEE.

[5] R. Creber, J. Rice, P. Baxley, and C. Fletcher, "Performance of undersea acoustic networking using RTS/CTS handshaking and ARQ retransmission," in Proc. of OCEANS 2001 MTS/IEEE Conf. and Exhibition, vol. 4, Nov. 2001, pp. 2083-2086.

[6] A. A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks",in Proc. INFOCOM 2006, April 2006, pp. 1C12.

[7] C Tian, W. Liu, J. Jin, J. W, and Y. Mo, Localization and Synchronization for 3D Underwater Acoustic Sensor Networks", Springer Berlin / Heidelberg , pp. 622C631, 2007.

[8] Nitthita Chirdchoo, Wee-Seng Soh, Kee Chaing Chua. "MU-Sync: A Time Synchronization Protocol for Underwater Mobile Networks". In Proc. WUWNET 2008.

[9] Xiang Yang Li, Yu Wangb, "Simple approximation algorithms and PTASs for various problems in wireless ad hoc networks", In Proc. Of Journal of Parallel and Distributed Computing Volume 66, Issue 4, April 2006, Pages 515-530.

[10] S. Ramanathan. "A unied framework and algorithms for (T/F/C)DMA channel assignment in wireless networks", In IEEE INFOCOM, pages 900-907, 1997.

[11] Affan Syed, Wei Ye, Bhaskar Krishnamachari, John Heidemann."Understanding Spatio -Temporal Uncertainty in Medium Access with ALOHA Protocols". In MobiCom 2007.

[12] Zhongwen Guo, Zhengbao Li, Feng Hong, "USS-TDMA: Self-stabilizing TDMA Algorithm for Underwater Wireless Sensor Network," iccet, vol. 1, pp.578-582, 2009 International Conference on Computer Engineering and Technology, 2009