Scheduling Real-Time Traffic in Underwater Acoustic Wireless Sensor Networks

Rodrigo Santos[1], Javier Orozco[1], Matías Micheletto[1], Sergio F. Ochoa[2], Roc Meseguer[3], Pere Millan[4], Carlos Molina[4]

[1] Dep. Electrical Engineering and Computers, IIIE, UNS-CONICET
[2] Computer Science Department, Universidad de Chile
[3] Department of Computer Architecture, Universidad Politécnica de Catalunya
[4] Department of Computer Engineering, Universitat Rovira i Virgili

Abstract. Underwater sensor networks are an important field of research. Several applications require the use of this kind of networks like tsunami or oil spill alerts. The underwater medium is very harsh and only acoustic signals can be used for the transmission of information. The use of this kind of networks is still in a developing state far from reaching standard consensus on basic aspects like carrier frequency or modulation techniques. The use of these networks for real-time applications has not been analyzed previously. In this paper we present two solutions for the scheduling of real-time messages and we provide a time constraint analysis of the performance of the network.

Keywords: Underwater Sensor Networks, Acoustic Sensor Networks, Environmental Monitoring

1 Introduction

Underwater acoustic wireless sensor networks are becoming a hot research topic as they have turned into the primary tool to monitor and act upon the well-being of marine environments [1, 2]. Radio frequency electromagnetic signals do not propagate well underwater. Huge amount of power is required to transmit messages even for short distances. The presence of particles and moving obstacles such as fishes or plants prevents the use of optical carriers. For underwater transmissions, the best option is to use acoustic carriers. While wireless sensor networks based on radio frequency transmissions have been studied and several protocols have been proposed like ZigBee, Bluetooth or even Wi-Fi, the solutions achieved for them are not useful for acoustic underwater networks since propagation delay is usually larger than transmission time. A message may be received well after its transmission has finished in the source node.

Tsunamis are generated by earthquakes in the ocean. Not every earthquake in the ocean produces a tsunami but the existence of an earthquake may end in a tragic tsunami like the ones in Japan 2011 or Indonesia 2004.

Detecting a tsunami is a hard work. Seismic sensors may be deployed in the area in which the earthquake may take place (geologic fault) and if this is detected, depending on the intensity a tsunami alert may be issued. The time available between the earthquake and the arrival of the wave to the beach depends on the distance to the earthquake epicenter. However, it is clear that there is a hard real-time restriction as the alert should be issued with enough time for people to move into a safe place.

The system may have some buoys anchored along the fault and linked to the seismic sensors so once the earthquake is detected, the buoy connects through a satellite network to a management disaster office reporting the event, intensity and tsunami probability. However, buoys are vandalized by pirates or even fishermen jeopardizing the network operation. To avoid this, an underwater acoustic sensor network is proposed operating in real-time. The network deployment, nodes distribution and number of hops discussion is out of the scope of this paper. However, the real-time analysis and network performance modeling proposed here can be used to set-up the appropriate network.

Real-time communications require not only that messages are transmitted properly but also before a particular instant named deadline. If the deadline is missed the message is not valid and may have serious consequences [3]. A feasible real-time schedule is one in which all messages comply with their deadlines. Real-time message scheduling in multi-hop networks is a complex problem that requires the use of routing and queueing techniques. If all the nodes in the network have a direct link to the rest of the nodes the problem may be solved using an integer linear programming approach as presented later in the paper. However, when a message should go through intermediate nodes, is not only a question of when a node should transmit (MAC problem) but also of selecting the appropriate path. In this case, the shortest path is not always the best one as a per node scheduling should be incorporated in the analysis. In fact, a node holding more than one message has to schedule their transmission introducing additional delays.

Recently, we have proposed a simple distributed medium access control protocol (MAC) for the case of underwater wireless sensors networks [4]. The network was modeled as a tree with a sonobuoy as root and sensors as leafs. The information flow was from the leafs to the root using intermediate nodes for aggregating the information collected in the previous layer. The synchronization process was made in a hierarchical way from the root to the leafs. At each layer a synchronizing node was selected following certain rules and these nodes were in charged of aggregating the messages during the data transmission stage. The proposed algorithm considered the possibility of re-configuring the tree periodically. However, it was not designed to operate under real-time constraints, power considerations were not incorporated in the analysis and the possibility of transferring data between any pair of nodes was not evaluated. In this paper, we use the initialization phase in which the nodes are synchronized and the network topology is discovered. After this, messages may be sent between any pair of nodes.

Like in standard wireless sensors networks (WSN), the most common MAC protocols can be grouped in two classes: Carrier Sense Multiple Access (CSMA) and Time Division Multiple Access (TDMA) oriented. In the first one, nodes transmit whenever they are able to lock the shared channel. For real-time messages this approach is not useful as message delays may be unbounded. TDMA may introduce an important latency but the worst case delay may be computed and the timed behavior of the network is predictable. Moreover, the TDMA approach can be divided in two modes. The first one considers that each node sends a broadcast to every node within transmission range. In this case, if a node has several messages to transmit to different destinations, it has to wait for equal number of frames. In the second approach, a per message TDMA is computed in such a way that the slot for sending a message from node a to node b is defined but also the moment at which node b receives the message from node a. In this case, each message has a particular slot to be sent and received at destination and the nodes may wait for the proper instants to become active.

Underwater sensors network operate with acoustic carriers. Two basic approaches are followed in the medium access protocol. Those based in CSMA [5–7] and those based in TDMA [8–10]. The first group can not guarantee a real-time performance so we are not going to comment on them. The second group proposes different algorithms to allocate the nodes within the frame. In [8] the authors proposed an heuristic approach, however they are not considering real-time constraints. In [9] tha authors propose a dynamic slot allocation. This approach provides an important aspect as the network may vary its topology so new allocations are necessary. However, the approach is not considering the routing and real-time aspects involve in the node to node transmissions. Finally in [10], the authors put the focus on energy aspects and the transmission power use. To this end they optimize the node allocation mechanism. Like the previous cases, no real-time analysis is performed.

Contribution: In this paper we extend the proposed algorithm presented in [11] to include real-time constraints and message transfers between any pair of nodes in the system. A TDMA access protocol is proposed with an off-line allocation and scheduling algorithm. Feasibility conditions are given for the system to operate with hard real-time constraints.

2 System Model

2.1 Physical Model

For the sake of simplicity, we assume the propagation delay between two nodes within transmission range is equal in both directions. That is, if a message originating in node a requires four time slots to arrive to node b, then a message originating in b requires four time slots to arrive to a, too.

In Figure 1, lets suppose that both nodes begin their transmission simultaneously. Both messages propagate at the same speed and have the same duration, one slot time. Under the previous assumption, a collision occurs exactly at two time slots. However, as these are longitudinal waves, there is a transitory composition of both waves at the point of collision but both of them continue their propagation. If the transmission power is enough, both nodes receive each other message after four time slots. It is important to note that local collisions do not propagate and that nodes only detect a local collision.

2.2 Real-Time Message Model

In this paper we assume that any node may transmit a message to any other node in the network if there is a valid path between both of them. We denote a message from node

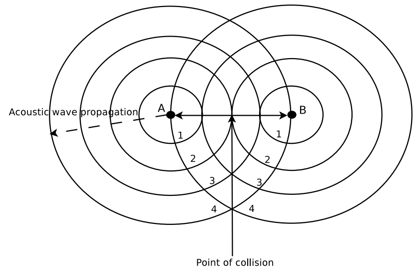

Fig. 1. Acoustic waves propagation and interference

a to node b as m_{ab}. We also assume all messages require one time slot to be transmitted and that they are sent periodically. Additionally, all messages should be received before the associated deadline. P_{ab} and D_{ab} represent the period and deadline respectively. In general we define the set of messages as $Z = \{m_{ij}(P_{ij}, D_{ij})\}$.

2.3 Network Model

The network can be modeled as a directed graph $G = (V, E)$, in which V is the set of nodes in the network and E the set of edges. If two nodes u and v are within transmission range, there is an edge connecting them, $e = (u, v)$. Each edge has a label that represents the transmission delay between the nodes measured in *time slots*, τ_{uv}. The set of nodes which have a direct link with $i \in V$ is the neighbor set and is notated $N(i)$. As collisions are important only if they are produced at the node, there are four different scenarios as stated in [8]. The first scenario is when two messages arrive simultaneously to a node, this case is named the Rx-Rx case. The second scenario is produced when two messages tried to be transmitted simultaneously in a node, Tx-Tx case. The third secenario is when a message is transmitted at the time another one is being received, this is the Tx-Rx case. The last one is named the Rx-Interference case and it is arisen when a message interferes another one in a node. The interfering message has a different destination node. This last case is similar to the Rx-Rx case.

We propose a slot allocation method to order the access of the nodes to the channel in such a way that each message originated in a node may reach its destination node without collisions. We begin considering that destination nodes are within transmission range of source/transmission node and later we extend the analysis for nodes at larger distances. Stated in this way, the slot assignment problem is an extension of the graph coloring problem [12]. The problem is similar to the L(2,1) labeling on graphs and the frequency assignment [13].

We present an integer linear programming (ILP) model, to minimize the frame length measured in slots. If we note $m_i \in N$ the slot in which node i transmits, we have the the following:

$$\texttt{Minimize} \sum_i m_i \tag{1}$$

Subject to:

$$
\begin{array}{llll}
m_i - m_j - M\delta_{ij} & \geq & \tau_{ij} + 1 - M & \forall i \in V, \forall j \in N(i) \\
m_i - m_j - M\delta_{ij} & \leq & \tau_{ij} - 1 & \forall i \in V, \forall j \in N(i) \\
m_j - m_k - M\omega_{ijk} & \geq & \tau_{ik} - \tau_{ij} + 1 - M & \forall i \in V \forall j, k \in N(i) \\
m_j - m_k - M\omega_{ijk} & \leq & \tau_{ik} - \tau_{ij} - 1 & \forall i \in V \forall j, k \in N(i) \\
\delta_{ij} & \in \{0,1\} & & \forall i \in V \forall j \in N(i) \\
\omega_{ijk} & \in \{0,1\} & & \forall i \in V \forall j, k \in N(i)
\end{array}
\tag{2}
$$

where M is a sufficiently large constant.

The model is significantly more complex if a per message slot allocation is performed. Further details can be found in [8].

3 Scheduling

Path discovery is a well known problem in networking. Several algorithms have been proposed to compute the best path for a message to reach destination from a source. The most common solutions are based on Dijsktra algorithm to determine the shortest path from any node in the network to any other node (SPF, shortest path first) or the Bellman-Ford distance vector algorithm. In the case of communication networks, the cost associated to the edges may be related to the actual delay between the nodes, an economical cost for using that link (paying service to a third party company) or the power required to use the link. For real-time messages, the total delay in the path should be less or equal to the deadline of the message. If this condition is not guaranteed, the message is not schedulable and the network does not comply the real-time requirements.

$$
\forall a, b \quad D_{ab} \geq \sum_{e \in \mathtt{path}(ab)} \tau_e \tag{3}
$$

Equation (3) sets the basic condition for the network to be schedulable. This requires that the sum of the transmission delays in each hop of the path plus the time needed in each intermediate node to gain access to the channel should be less or equal to the deadline of the message. Besides, the transmission to the next hop should be scheduled in such a way that no collisions at the destination or related nodes are produced.

3.1 Node scheduling policy - First-In First-Out

When a node has several messages ready to be transmitted, a scheduling policy should be implemented to order the transmission. In what follows we describe the First-in First-out policy (FIFO). Later a heuristic approach is discussed.

FIFO is easy to implement. However, messages do not have a priority associated and an urgent message may wait for several frames before being transmitted. As messages are periodic, the waiting time in the queue is computed from the worst case arrival of the message to the queue and the frame length. Depending on the kind of TDMA computed, node or message allocation, the node should have one or more queues depending on the amount of neighbors.

In either case, the frame size, T_f is used as the time based to determine the delay in the waiting queue W_i in the node.

$$W_i = (Q-1)T_f + m_i \tag{4}$$

where Q is the length of the queue at the moment in which a message arrives or is generated.

4 Example

In this section we present a simple example to show the different aspects discussed in the previous sections. Figure 2 shows a five node network with the labels in the edges representing the transmission delay between both nodes.

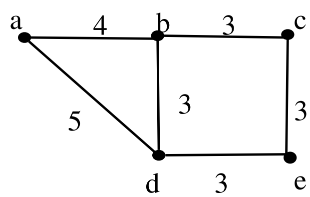

Fig. 2. 5 nodes network example

The ILP model for this network has twelve messages for allocating in the frame. Each node transmits a message to all its neighbors. The transmission and reception scheme is shown in Figure 3 where $T(ab)$ stands for a slot used by node a to send a message to node b and $R(ba)$ stands for a message received at node a from node b. The allocation was computed with GLPK 4.55 [14].

Node	Slots												
	1	2	3	4	5	6	7	8	9	10	11	12	13
a	$T(ab)$	$T(ad)$									$R(ba)$		$R(da)$
b	$T(bd)$		$T(bc)$		$R(ab)$		$T(ba)$			$R(cb)$		$R(db)$	
c			$T(ce)$			$R(bc)$	$T(cb)$				$R(ec)$		
d	$T(de)$			$R(bd)$	$R(ed)$		$R(ad)$	$T(da)$	$T(db)$				
c		$T(ed)$		$R(de)$		$R(ce)$		$T(ec)$					

Fig. 3. Message/Slot allocation for transmission and reception.

The schedule presented is minimum for transmitting a message from each node in the network to all its neighbors. The frame size has a duration of 13 slots. The empty slots consider the fact that messages should not interfere in the reception. Node a is active only in 4 out of 13 slots so it may save energy being kept shut down during the slots in which it has no activity.

The frame imposes a general upper bound for each hop. That is, in the worst case, for each hop in the path a whole time frame delay is introduced. Thus, transmitting a message from node a to node e, the SPF algorithm has a lower bound of 10 slots and an upper bound of 17 slots. In the analysis we are still not considering queueing problems in the nodes. This means that in node a there is just one message that has to be sent to node d and in this last one, only one message (the one received from a) has to be forwarded to node e. If this is not the case, a scheduling policy in each node should be selected and the transmission delay is incremented.

If a per node slot allocation scheme is used, the frame is reduced to just 6 slots but a node requires several frames to send consecutive messages to the neighbors and all the neighbors are listening while it is transmitting. Figure 4 shows the slot allocation in this case.

Node	Slots					
	1	2	3	4	5	6
a	$T(a)$				$R(b)$	$R(d)$
b	$T(b)$			$R(d)$	$R(a)$	$R(c)$
c			$T(c)$	$R(b)$	$R(e)$	
d	$T(d)$			$R(b)$	$R(e)$	$R(a)$
e		$T(e)$		$R(d)$		$R(c)$

Fig. 4. Node/Slot allocation for transmission of messages.

Like in the previous case node a is on in 3 out of 6 slots being off in the rest saving energy.

Let's consider a set of 8 messages.

$$Z = \{m_{ae}(20, 30), m_{ac}(20, 30), m_{ea}(20, 30), m_{ca}(20, 30), m_{de}(20, 30), \\ m_{db}(10, 10), m_{bc}(10, 10), m_{ae}(15, 20)\} \quad (5)$$

A message from a to e has a transmission delay of 8 units and goes through two nodes a and d before reaching destination. For the case of node allocation, and considering a queue with one place for each connected neighbor, in the worst case, the message has to wait for one frame ($Q = 2$) in node a to be transmitted in the following one, see (4). Once it has arrived to node d it may wait for two frames ($Q = 3$) before being transmitted to node e.

$$\tau_{ae} = W_a + W_d + \tau_{ad} + \tau_{de} = 6 + 1 + 5 + 12 + 1 + 3 = 26$$

In the described situation, considering the worst case, m_{ae} has an end-to-end delay of 26 slots. Computing in the same way for the rest of the messages we found the worst case delays for all messages as presented in Table 1 in the second column.

m_{ij}	m_{ae}	m_{ac}	m_{ea}	m_{ca}	m_{de}	m_{db}	m_{bc}	m_{be}
τ_{ij}	26	25	29	29	16	16	16	21

Table 1. Messages worst case delays

The network scheduling of the example proposed shows that the system is stable and that messages will go through the network within their deadlines.

5 Heuristic approach

In the proposed model, the variables that affect the communication speed and therefore the timing of the system are the frame duration, the order of transmissions and reception of messages and the routes that each message follows within the network. In this section an heuristic algorithm is presented to optimize the message/slot allocation to minimize the frame size and guarantee the deadlines.

The minimum length frame is not necessarily the optimal to meet the system time requirements and this impedes uncoupling the calculation of the frame with respect to the calculation of routes. The heuristic presented in this section generates a fixed length frame and optimize the paths of the messages to meet all system deadlines.

Let H denote the least common multiple between all message periods. If the frame is a time window of fixed length H, there is a number $H/P_{i,j}$ instances of the message $m_{i,j}$, so $m_{i,j,k}$ denotes the k-th instance of the message $m_{i,j}$.

Each instance $m_{i,j,k}$ has an associated path which is chosen from all possible paths between nodes i and j. The path is noted $path_{i,j,k}$ and contains the nodes of the network which the message goes through. Optimal path for each instance must be calculated to obtain the best solution, as will be explained later.

As messages require communication between nodes to propagate through the network, node allocation in the frame is performed following the corresponding route. Each node must be allocated to the minimum slot to ensure meeting all deadlines.

Each node is associated to a binary interference matrix which simplifies the collision avoidance scheduling. This matrix has dimensions n by τ_{max} where n is the number of nodes and τ_{max} is the delay between the transmitting node and its farthest neighbor. For the i-th node, the element $(i, 1)$ of the interference matrix is always 1, and the element (j, s) is 1 if and only if j-th node has a communication delay of s slots from i-th node. To perform allocability test of a transmitter node to a desired slot, interference matrix of the transmitter node is overlapped on the frame so that the first column of the interference matrix match the column of the frame correspondig to the allocation slot and finally an element by element product is calculated. If the result is a null matrix, then the allocability test is positive, which means transmitter node is allocable to that slot. In Section 5.1 we present the allocation procedure for the example of Section 4.

Once all the nodes of the path are allocated, the message has arrived to destination, and time remaining until deadline is calculated and denoted by $margin_{i,j,k}$. The pseudocode 1 shows the scheduling procedure required to generate the communication frame.

5.1 Node allocation example

Based on example of Section 4, Figure 5 shows a portion of the frame where there are free and taken slots. It is desired to know if node d is allocable to slot 9. As it can be seen, if node d transmits at slot 9, there is a Rx-Rx collision on node b at slot 12 and a Tx-Rx collision on node e at slot 12. Following the same procedure, node d is not allocable to slot 10 either. Finally, Figure 7 shows that node d is allocable to slot 11.

```
frame:=zeros(N,H) // Frame matrix initialization.
forall the m_{i,j,k} sorted by increasing order of periods do
    slot:=(k − 1) · P_{i,j} // First slot of period.
    transmitter:=path_{i,j,k} (first) // First node of path.
    receiver:=path_{i,j,k} (second) // Second node of path.
    destination:=path_{i,j,k} (last) // Second node of path.
    while True do
        if allocable(frame, transmitter, slot) then
            allocate(frame, transmitter, slot)
            slot:=slot+τ_{transmitter,receiver}
            if receiver==destination then
                break
            end
            else
                transmitter:=receiver
                receiver:=path_{i,j,k} (next)
            end
        end
        else
            slot:=slot+1
            if slot==H then
                return 'Non schedulable system with chosen paths'
            end
        end
    end
    margin_{i,j}:=j ∗ P_i + D_i - slot
end
return margin_{i,j}, frame
```

Algorithm 1: Scheduling algorithm pseudocode.

5.2 Optimization algorithm

Towards obtaining optimal path for each message instance, different optimization algorithms can be implemented to solve the problem. Figure 8 shows a simplistic representation of the optimization model that iterates over a loop in order to improve the solution's quality. As it is desired to find a frame where communication between nodes allows messages to reach their destination in the minimum number of slots as possible, it is considered the summation of all delay margins between the arrival time of messages and their deadlines as the quality of the solution. In the context of the mathematical optimization, the objective function takes as input the set of paths for each message instance and returns a scalar quantity as output which represents the quality of the proposed solution. The objective function calculates the output following three ba-

		8	9	10	11	12	13	14	15	16	
a	...	T(a)	R(b)		T(a)	R(d)			R(b)		...
b				R(d)	T(b)	R(a)	R(c)		R(a)		
c		R(b)		T(c)		R(e)		R(b)	R(e)		
d		R(b)				R(e)	R(a)	R(b)	R(e)	R(a)	
e			T(e)	R(d)		T(e)	R(c)				

Fig. 5. Section of the communication frame.

		8	9	10	11	12	13	14	15	16	
a	...	T(a)	R(b)		T(a)	R(d)			R(b)		...
b				R(d)	T(b)	R(a)	R(c)		R(a)		
c		R(b)		T(c)		R(e)		R(b)	R(e)		
d		R(b)				R(e)	R(a)	R(b)	R(e)	R(a)	
e			T(e)	R(d)		T(e)	R(c)				

Fig. 6. Allocation test of node d to slot 9. Collision slots highlighted.

		8	9	10	11	12	13	14	15	16	
a	...	T(a)	R(b)		T(a)	R(d)			R(b)		...
b				R(d)	T(b)	R(a)	R(c)		R(a)		
c		R(b)		T(c)		R(e)		R(b)	R(e)		
d		R(b)				R(e)	R(a)	R(b)	R(e)	R(a)	
e			T(e)	R(d)		T(e)	R(c)				

Fig. 7. Allocation test of node d to slot 11.

sic steps, first, generates the communication frame following the Algorithm 1, second, determines the delay margin for each instance and finally computes the solution quality with the summation of all delay margins.

5.3 Analysis of results

The proposed heuristic is better suited for complex problems where exists multiple paths for different messages. However, example of Section 4 is considered for better understanding of the situation.

In order to begin the example with a simple case, the objective function was tested using shortest path for each message. Table 2 shows the delay margins of each message instance. This solution is feasible, has no message collisions, and no deadline is missed.

In order to improve the previous result, a genetic algorithm was used to find an optimal solution. Different instances of the optimization finds the same solution which has a slight difference with the previous one. As it can be seen from the results listed in Table 2, message 8 alternates its path in the first and last instance, and this relieves the load of node d, so other messages can arrive to their destination with greater delay margin.

6 Conclusions and Future Work

This paper presented a real-time analysis for an underwater acoustic wireless network. We have introduced two approaches. In the first one, the network is analyzed with in-

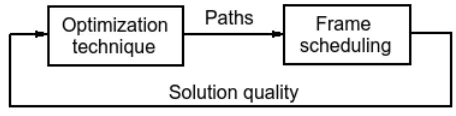

Fig. 8. Optimization model.

$m_{i,j,k}$	SFP		GA	
	$path_{i,j,k}$	delay margin	$path_{i,j,k}$	delay margin
$m_{a,e,1}$	a-d-e	21	a-d-e	21
$m_{a,e,2}$	a-d-e	22	a-d-e	22
$m_{a,e,3}$	a-d-e	22	a-d-e	22
$m_{a,c,1}$	a-b-c	19	a-b-c	21
$m_{a,c,2}$	a-b-c	22	a-b-c	22
$m_{a,c,3}$	a-b-c	19	a-b-c	19
$m_{e,a,1}$	e-d-a	21	e-d-a	21
$m_{e,a,2}$	e-d-a	19	e-d-a	21
$m_{e,a,3}$	e-d-a	19	e-d-a	21
$m_{c,a,1}$	c-b-a	17	c-b-a	18
$m_{c,a,2}$	c-b-a	18	c-b-a	18
$m_{c,a,3}$	c-b-a	14	c-b-a	17
$m_{d,e,1}$	d-e	16	d-e	19
$m_{d,e,2}$	d-e	19	d-e	19
$m_{d,e,3}$	d-e	15	d-e	16
$m_{d,b,1}$	d-b	7	d-b	7
$m_{d,b,2}$	d-b	7	d-b	7
$m_{d,b,3}$	d-b	7	d-b	7
$m_{d,b,4}$	d-b	7	d-b	7
$m_{d,b,5}$	d-b	7	d-b	7
$m_{d,b,6}$	d-b	7	d-b	7
$m_{b,c,1}$	b-c	7	b-c	7
$m_{b,c,2}$	b-c	7	b-c	7
$m_{b,c,3}$	b-c	7	b-c	7
$m_{b,c,4}$	b-c	7	b-c	7
$m_{b,c,5}$	b-c	7	b-c	7
$m_{b,c,6}$	b-c	7	b-c	7
$m_{b,e,1}$	b-d-e	12	b-c-e	12
$m_{b,e,2}$	b-d-e	14	b-d-e	12
$m_{b,e,3}$	b-d-e	12	b-d-e	12
$m_{b,e,4}$	b-d-e	14	b-c-e	12

Table 2. SPF and Genetic algorithm solutions evaluation

teger linear programming techniques. The shortest path first is used as routing policy combined with a message or node slot allocation procedure in a TDMA frame. Based on this, we presented the schedulability condition for the case in which messages are transmitted following a First-In First Out policy. This scheduling discipline is quite simple and requires little processing within the underwater nodes reducing the computing complexity and demand on the processors. However, better results may be obtained if some real-time priority policies are implemented like fixed priorities or earliest deadline first. The performance of the network with these is left for future work. The second solution is based on a heuristic approach. In this case, messages are scheduled following a per link approach and finding the route with lower delay. The solution obtained in this way improves the two step approach of finding the SPF in the first place for allocating the slots within the frame later. As the heuristic only considers the messages actually being transmitted, unnecessary restrictions are avoided. We have also presented a real application in which real-time transmissions are necessary. Tsunami early alert problem is a very important issue for countries in the Pacific and Indic oceans and the routing proposal here introduced may help satisfy the real-time requirements.

References

1. Açar, G., Adams, A.: Acmenet: an underwater acoustic sensor network protocol for real-time environmental monitoring in coastal areas. IEE Proceedings-Radar, Sonar and Navigation **153**(4) (2006) 365–380
2. Akyildiz, I., Pompili, D., Melodia, T.: Underwater acoustic sensor networks: research challenges. Ad hoc networks **3**(3) (2005) 257–279
3. Liu, C.L., Layland, J.W.: Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of the Association for Computing Machinery **20**(1) (jan 1973) 46–61
4. Santos, R., Orozco, J., Ochoa, S.F., Meseguer, R., Eggly, G., Pistonesi, M.F.: A mac protocol to support monitoring of underwater spaces. Sensors **16**(7) (2016) 984
5. Fang, D., Li, Y., Huang, H., Yin, L.: A csma/ca-based mac protocol for underwater acoustic networks. In: 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM). (Sept 2010) 1–4
6. Guo, X., Frater, M.R., Ryan, M.J.: A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks. In: OCEANS 2006 - Asia Pacific. (May 2006) 1–6
7. Pompili, D., Melodia, T., Akyildiz, I.: A cdma-based medium access control for underwater acoustic sensor networks. Wireless Communications, IEEE Transactions on **8**(4) (April 2009) 1899–1909
8. Guan, Y., Shen, C.C., Yackoski, J.: Mac scheduling for high throughput underwater acoustic networks. In: 2011 IEEE Wireless Communications and Networking Conference. (March 2011) 197–202
9. Chen, Y.D., Lien, C.Y., Chuang, S.W., Shih, K.P.: Dsss: A tdma-based mac protocol with dynamic slot scheduling strategy for underwater acoustic sensor networks. In: OCEANS 2011 IEEE - Spain. (June 2011) 1–6
10. Diamant, R., Lampe, L.: Spatial reuse time-division multiple access for broadcast ad hoc underwater acoustic communication networks. IEEE Journal of Oceanic Engineering **36**(2) (April 2011) 172–185
11. Santos, R., Orozco, J., Ochoa, S.F.and Meseguer, R., Eggly, G.: A mac protocol for underwater sensors networks. In García-Chamizo, J., Fortino, G., Ochoa, S., eds.: Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information. Volume 9454., Springer International Publishing 394–406
12. Lv, C., Wang, S., Tan, M., Chen, L.: Ua-mac: An underwater acoustic channel access method for dense mobile underwater sensor networks. International Journal of Distributed Sensor Networks **2014**
13. Shao, Z., Yeh, R.K., Zhang, D.: The l(2 , 1)-labeling on graphs and the frequency assignment problem. Applied Mathematics Letters **21**(1) (2008) 37 – 41
14. Makhorin, A.: Glpk (gnu) linear programming kit