DMC-MAC: DYNAMIC MULTI-CHANNEL MAC IN UNDERWATER ACOUSTIC NETWORKS

Hamid Ramezani, and Geert Leus

Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology
{h.mashhadiramezani, g.j.t.leus}@tudelft.nl.

ABSTRACT

In this article, we focus on the broadcasting task in an underwater acoustic sensor network when a few sensor nodes want to transmit their packets to the nodes within their communication range. Here, we utilize the relative position information of the transmitting nodes to adaptively determine the best channel allocation (multichannel transmission) and packet transmission scheduling that minimizes the collision-free broadcasting duration. Analytical results and examples show that adaptive multichannel packet transmission scheduling greatly reduces the broadcasting duration, and hence improves network efficiency.

***Index Terms*— Underwater acoustic, MAC, Dynamic multi-channel allocation, scheduling.**

1. INTRODUCTION

Although a considerable amount of research has been conducted in medium access control (MAC) protocol design for wireless sensor networks (WSNs) [1], the unique characteristics of underwater acoustic communications, such as a very low and distance dependent bandwidth [2], high power consumption in transmit and receive mode, and long propagation delay, make them inefficient and sometimes inapplicable for underwater acoustic sensor networks (UASNs). This inspired researchers either to adapt the existing WSN MAC protocols to UASNs or picture new ones. For instance, in a time division multiple access (TDMA) system, in order to decrease the collision probability, the slotted floor acquisition multiple access (FAMA) [3] sets the time slot duration equal to the packet length plus the maximum network propagation delay. In another work, with the knowledge of mutual distances among the sensor nodes, a transmitting node adjusts the guard time of its TDMA slot according to its distance to other nodes [4]. Using information of the relative distances between the sensor nodes provides a considerable gain in the network throughput. Based on the packet length, [5] demonstrates that this gain can be as large as 60 to 100%.

Another factor that augments network efficiency is the use of several independent channels for transmitting data [6]. According to [7], multi-channel MAC protocols help to improve network efficiency. Zhou et al. have analytically studied the idea of multi-channel MAC protocols [8]. They have shown that their theoretical analysis can closely estimate the system performance which is better than the one related to single-channel MAC protocols. Basically, multi-channel techniques reduce the probability of collisions in a network, but it is not well known how it acts in collision-free MAC protocols such as ordered carrier sense multiple access (OCSMA) [9] or [10].

In this article, we consider an underwater collision-free broadcasting MAC protocol which can adaptively determine the number of data-channels based on the nodes' relative propagation delays (based on relative position information), and their maximum communication range. Although here we focus on a broadcast scenario, the concept of this paper can be extended to point-to-point MAC protocols as well.

Two scenarios will be considered here. In the first scenario, the transmitted packet from a transmitting node has to be received by all the other sensor nodes within the communication range of the transmitter, (e.g., broadcasting locally gathered information), which we call B-MAC. In the second scenario, it has to be received by all the sensor nodes in the communication range of the transmitter excluding the nodes which have packets to send, (e.g., broadcasting localization packets from anchors), which we call L-MAC [11].

The rest of the paper is organized as follows. In Section 2, we explain the network and physical model. We formulate and analyze the problem of single and multichannel broadcasting in Sections 3 and 4, respectively. Afterwards, we propose a suboptimum algorithm to tackle the problem in Section 5, evaluate its performance in Section 6, and finally conclude the paper in Section 7.

2. SYSTEM MODEL

2.1. Network model

We consider a UASN with N sensor nodes where K of them have packets of $b_k (k = 1, \ldots, K)$ information bits, and they want to broadcast their packets to the sensor nodes which are located in their communication range as shown in Fig. 1. Moreover, it is assumed that the indices of these K nodes are stored in Ω, their maximum transmission range is D meters, their positions are known at a fusion center, and there is no information about the position of the other nodes (silent nodes) in the network (they can be anywhere in the operating area). For instance, it can be imagined that these K nodes succeeded in channel acquisition in a contention period, or

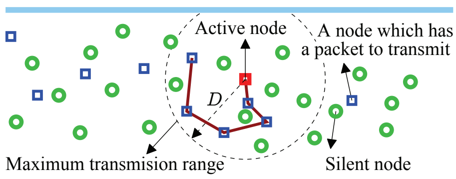

Fig. 1. Network model, vertical cross-section view. The solid line represents the Hamiltonian path between the transmitting nodes which are located within the communication range of each other (the assumption of a fully connected network).

The research leading to these results has received funding from the European Commission FP7-ICT Cognitive Systems, Interaction, and Robotics under the contract #270180 (NOPTILUS).

they are anchors which transmit localization packets periodically in a network. Therefore, it is reasonable to assume that the information about the positions of the transmitting nodes is available.

The goal of the network is to minimize the time duration of the broadcasting (or localization) task which is defined as the time of collision-free transmission of all packets from the transmitting nodes. If we assign a waiting time to each transmitting node which can be interpreted as the time that a node has to wait to transmit its packet, the problem can be formulated as,

$$\begin{aligned} \min \max_{k \in \{1,2,\dots,K\}} & w_k + \tfrac{b_k}{R_k}, \\ \text{s.t. } w_k \geq 0, & \text{ and collision-free transmission,} \end{aligned} \quad (1)$$

where w_k is the waiting time of node k, and R_k is the data rate at which node k transmits its packet. It is important to note that the waiting times are the outputs of any algorithm that tackles this problem.

2.2. Physical layer model

According to the developments in software-defined radio systems it is quite acceptable to assume that the sensor nodes are capable of adaptively adjusting their transmission data rate, bandwidth, frequency of operation and so on. Based on this capability, node k is able to transmit its data using the full system bandwidth (or other system resources), with data rate R, i.e., $R_k = R$. Alternatively, the full bandwidth can be splitted into M subbands where the m-th channel ($m \in \{1, 2, ..., M\}$) has data-rate

$$R^{(m)} = \beta^{(m)}(1 - \alpha^{(1:M)})R, \text{ s.t. } \sum_{m=1}^{M} \beta^{(m)} = 1, \quad (2)$$

where M is number of generated channels, $\beta^{(m)}$ is the percentage of the available data rate that a node uses to transmit its packet, $R^{(1:M)} = (1 - \alpha^{(1:M)})R$ is the sum of the data rates of all generated channels after channel splitting, and $\alpha^{(1:M)}$ is the percentage of throughput reduction due to the increase in the number of channels from 1 to M. Using this channel splitting, node k can then decide to choose the m-th channel and transmit at data rate $R^{(m)}$, i.e., $R_k = R^{(m)}$. In addition, at the receiver side, each node is able to detect the transmitted data regardless of how many channels are generated after channel splitting and which channels the data are transmitted in. Furthermore, we assume a sensor node can be either in the receiving mode or transmitting mode, and it can not simultaneously transmit and receive.

2.3. Collision-free transmission using one channel

Since the silent nodes can be located anywhere in the operating area, collisions may occur if the transmitted packets from two or more nodes sweep a point in this area at the same time. Imagine that two transmitting nodes, namely i and j, with respective waiting times w_i and w_j are going to transmit their packets to the nodes in their communication range, and have to satisfy the following conditions

$$w_j > w_{j,\min},\ w_i > w_{i,\min},\ w_{j,\min} > w_{i,\min}, \quad (3)$$

where $w_i > w_{i,\min}$ indicates that node i has to start its packet transmission not sooner than $w_{i,\min}$. Clearly, when the distance between these two nodes is greater than $2D$, their packets never collide with each other. However, when the distance between the two nodes is smaller than $2D$ (defined as connected or collision-risk nodes), the possibility of a collision exists, but we can eliminate this possibility by adjusting the waiting times of the nodes according to the following formulas [11]:

$$w_j > w_i + \frac{b_i}{R_i} + \frac{\tilde{d_{ij}}}{c}, \text{ if } d_{ij} < 2D. \quad (4)$$

$$\tilde{d_{ij}} = \min(d_{ij}, 2D - d_{ij}), \quad (5)$$

where c is the sound speed, and $\tilde{d_{ij}}$ can be considered as a modified distance between two collision-risk nodes (i.e., $d_{ij} < 2D$). Note that, for simplicity we assumed that there is no effect from a transmitted packet beyond its communication range. However, in reality the interference range has to be considered, and this can be included if a guard ring is added to the maximum transmission range.

A network in which the distances between the connected nodes (collision-risk nodes) are modified (see (5)) is called the *modified network*. With this definition, (4) indicates that to eliminate the possibility of a collision between two collision-risk nodes in a modified network, one node ought to wait until the complete reception of the transmitted packet from the other node before it can start its transmission. The next section briefly reviews how the minimum broadcasting time using a single channel in a network can be obtained [11].

3. SINGLE CHANNEL BROADCASTING

Here, we claim that the optimal solution, which is not unique, belongs to at most $K!$ permutations of the transmitting nodes' indices. In each sequence (or permutation), the node whose index appears earlier has to transmit its packet sooner than the ones which appear later. Conditioned on a given sequence, the minimum duration of the broadcasting task, $T_{\text{broadcast}}$, can simply be computed based on (4). In this procedure, the first transmitting node is assigned to transmit its packet first, then the limitations on the transmission time of the other nodes are computed. Next, based on the computed waiting time (which is greater than or equal to the transmission time of the previously scheduled nodes) the second node transmits its packet. For instance, suppose that in a given sequence the index of node j appears after the one of node i, and the mutual distance between them is $d_{ij} < 2D$. It is again assumed that the conditions in (3) hold here. Hence, node i transmits its packet with data rate $R_i = R$, and after the complete reception of the packet at node j, node j is allowed to broadcast its packet with data rate $R_j = R$ according to its waiting time limit (see upper part of Fig. 2). Under this condition the duration of the collision-free broadcasting from these two nodes can be computed as

$$t_{ij} = w_j + \frac{b_j}{R_j}, \text{ s.t. } w_j > \max(w_{j,\min}, w_i + \frac{d_{ij}}{c} + \frac{b_i}{R_i}), \quad (6)$$

where $\frac{b_i}{R_i}$ is the duration of the packet transmitted by node i. The minimum value of t_{ij} can be obtained if we set

$$w_i = w_{i,\min} \quad (7a)$$

$$w_j = \max(w_{j,\min}, w_{i,\min} + \frac{b_i}{R_i} + \frac{d_{ij}}{c}). \quad (7b)$$

It can be deduced that after scheduling a node (e.g., node i), the remaining nodes (e.g., node j) have to update their waiting times, $w_{j,\min} = w_j$, based on (7). Doing this for each node in the given sequence leads to the minimum value of broadcasting time. Therefore, by comparing the broadcasting time of all sequences ($K!$ possible sequences), the one which has the lowest value is the optimal solution.

In the special case where each node in Ω is a collision-risk neighbor of the other nodes in Ω (here referred to as a *fully-connected* network), the nodes have to transmit their packets one after the other, and the duration of the transmission task is minimized if the Hamiltonian path (based on the modified distances) is formed and the nodes transmit their packets according to this path as depicted in Fig. 1. Note that in this figure only the nodes that are located on the drawn Hamiltonian path are considered for transmission (fully-connected network). Finding the Hamiltonian path in a network is equivalent to the traveling salesman problem (TSP) which is NP-hard. However, it has been studied extensively [12], and there are many sub-optimum algorithms for that.

4. DYNAMIC MULTICHANNEL BROADCASTING

In this section, the concept of channel splitting between transmitting nodes is introduced. This concept is formulated in the L-MAC and B-MAC scenarios.

4.1. Channel splitting in the L-MAC scenario

Assume that given a sequence, a few unscheduled nodes can split the channel between themselves and transmit their packets simultaneously as shown in the lower part of Fig. 2 for two transmitting nodes[1]. For example, suppose that node i and j decide to split the whole channel into channels 1 and 2 with respective data rates $R^{(1)} = \beta^{(1)}(1-\alpha^{(1:2)})R$, and $R^{(2)} = \beta^{(2)}(1-\alpha^{(1:2)})R$, and suppose node i will use channel 1, i.e., $R_i = R^{(1)}$, and node j channel 2, i.e., $R_j = R^{(2)}$. The minimum broadcasting duration can then be formulated as

$$t_{ij,\text{split}} = \min_{w_i,w_j} \max_{\beta^{(1)},\beta^{(2)}} \tag{8}$$
$$(w_i + \frac{b_i}{\beta^{(1)}(1-\alpha^{(1:2)})R}, w_j + \frac{b_j}{\beta^{(2)}(1-\alpha^{(1:2)})R}),$$
$$\text{s.t. } w_{j,\text{min}} < w_j, \quad w_{i,\text{min}} < w_i,$$
$$\beta^{(1)} + \beta^{(2)} = 1, \text{ and } \beta^{(1)}, \beta^{(2)} > 0.$$

Since one of the arguments of the $\max$ operator in (8) is monotonically increasing and the other is monotonically decreasing, the minimum value of this operator, i.e., $t(w_i, w_j)$, will be obtained when the arguments of (8) are equal which leads to

$$\beta_{1,\text{opt}} = \frac{b_i}{R^{(1:2)}} \frac{1}{t(w_i, w_j) - w_i}, \tag{9a}$$

$$\beta_{2,\text{opt}} = \frac{b_j}{R^{(1:2)}} \frac{1}{t(w_i, w_j) - w_j}. \tag{9b}$$

Using $\beta^{(1)} + \beta^{(2)} = 1$, the minimum broadcasting time can then be obtained as

$$t_{ij,\text{split}} = \min_{w_i,w_j} t(w_i, w_j) =$$
$$\min_{w_i,w_j} \frac{1}{2}\left(w_i + w_j + \frac{b_i + b_j}{R^{(1:2)}}\right) +$$
$$\frac{1}{2}\left[\left(w_i - w_j - \frac{b_i - b_j}{R^{(1:2)}}\right)^2 + \frac{4 b_i b_j}{[R^{(1:2)}]^2}\right]^{\frac{1}{2}}. \tag{10}$$

[1]Here, we limit our analysis for two unscheduled nodes, but this analysis can be extended easily to more number of nodes.

It can be shown that the argument of the $\min$ operator in (10) is a monotonically increasing function of w_i and w_j, and therefore the minimum value for broadcasting will be obtained if $w_i = w_{i,\text{min}}$ and $w_j = w_{j,\text{min}}$. After scheduling these two nodes, the waiting times of the unscheduled nodes in a sequence have to be updated according to their distance to these two nodes (see (4)). However, this time the updated waiting time is not for the whole bandwidth, but depends on which bandwidth the scheduled nodes transmit in. In other words, the updated waiting times become a piecewise constant function of the frequency.

In a more general setting, in the beginning of the channel splitting between the nodes, different waiting times for different parts of the network resources (for instance different sub-bands in frequency division multiple access) may exist which should be considered in the optimization problem of (8). We care about this issue in the proposed algorithm.

4.2. Channel splitting in the B-MAC scenario

In the B-MAC scenario, the transmitted packet by a transmitting node has to be received by all the silent nodes as well as other transmitters which are located within its communication range. This imposes other conditions on the channel splitting between two transmitting nodes located $d_{ij} < D$ meters away from each other formulated as

$$\frac{b_i}{\beta^{(1)}(1-\alpha^{(1:2)})R} < w_j - w_i + \frac{d_{ij}}{c}, \tag{11a}$$

$$\frac{b_j}{\beta^{(2)}(1-\alpha^{(1:2)})R} < w_i - w_j + \frac{d_{ij}}{c}, \tag{11b}$$

which means that a transmitting node has to finish its packet transmission before the packets from other transmitters reach it. It can be shown that channel-splitting for B-MAC is possible if

$$\max\left(\frac{b_i}{\beta^{(1)}(1-\alpha^{(1:2)})R}, \frac{b_j}{\beta^{(2)}(1-\alpha^{(1:2)})R}\right) \leq \frac{d_{ij}}{c}, \tag{12}$$

which means the propagation delay between two nodes has to be greater than the packet length after channel splitting. Otherwise, channel-splitting is not possible for the B-MAC.

In order to minimize the broadcasting time in the B-MAC scenario, the conditions in (11) have to be added to (8) which turns the optimization problem into a non-convex one. However, due to the large propagation delays, it is quiet common that $t(w_i, w_j) < \min(w_i, w_j) + \frac{d_{ij}}{c}$, and as a result most of the time (where few nodes in a network have packets for transmission) the optimum value of (10) also holds for the B-MAC scenario.

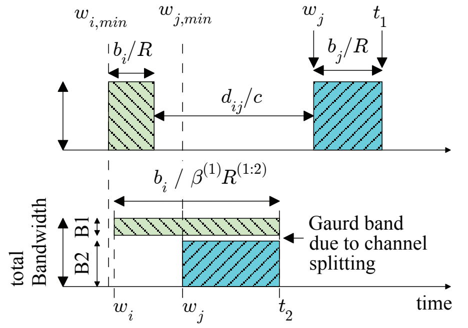

Fig. 2. Comparison of channel splitting technique with single channel transmission between two nodes.

5. PROPOSED SUBOPTIMAL ALGORITHMS

As for single channel scheduling, the problem of minimizing the broadcasting time based on dynamic channel allocation is NP hard, and therefore not tractable. In order to find the optimum value, all the possible solutions (permutations and combinations of all transmitting nodes' indices) have to be analyzed. However, this exhaustive search has a huge computational complexity especially when the number of transmitting nodes is considerable. Thus, in this section, a greedy dynamic multichannel allocation algorithm for a system that uses at most M_{max} orthogonal channels with equal bandwidth, and equal packet length is proposed. The algorithm contains three subalgorithms which are explained below.

The steps of the first subalgorithm are listed in Algorithm 1 which schedules the packet transmission given a fixed number of orthogonal channels, M. Based on the application (B-MAC, L-MAC, TDMA), it adjusts the waiting times of the transmitting nodes for collision-free transmission. In the initial phase the waiting times of the transmitting nodes are set to zero, and a buffer of size $K \times M$ is defined to store the transmission waiting time in each channel for each node. The algorithm starts from the I-th node, random starter (RS), and assigns the first channel ($m = 1$) to it, and updates the waiting times of the neighboring nodes based on (4). Note that for B-MAC, this limit is an additionally limit that is put on all channels if the distance of a neighbor is less than D, because a node has to receive the packet from the already assigned node, and cannot transmit while it is receiving a packet. In a TDMA system, updating the waiting-times (or putting limits on them) is done based on the duration of the time-slot [3] which is defined as $t_{\text{s}} = t_M^{\text{p}} + \frac{D}{c}$ where $t_M^{\text{p}} = \frac{Mb}{R^{(1:M)}}$. Then, a node which has the minimum waiting time (in all the channels) is selected for the next round, and this procedure continues until all nodes are scheduled. Eventually, the waiting time of each node is assigned as the minimum one it has in different channels, and it is allowed to transmit in that channel. In the second subalgorithm, namely the best starter (BS), we run the first subalgorithm K times starting with a different node each time. Then, we compare the broadcasting time of all these K answers and select the one which has the lowest value.

The third subalgorithm, called multi-channel (MC), uses Algorithm 1 (or the second subalgorithm) for different numbers of channels ($M = 1, ..., M_{\text{max}}$). Among all these numbers of channels, it selects the one which leads to the fastest response. The value of M_{max} depends on $\alpha^{(1:M)}$ and the maximum vertex degree in the modified network, and is defined as

$$M_{\text{max}} = \arg\max_{M} \alpha^{(1:M)} \text{ s.t. } \alpha^{(1:M)} < 1. \qquad (13)$$

Because of the combinatorial nature of the minimization problem, other metaheuristic algorithms such as the greedy randomized adaptive search procedure (GRASP), simulated annealing, and genetic algorithms can also be employed here. Nevertheless, due to the space limitations we do not examine them here.

In the next section, we evaluate the performance of the proposed greedy algorithm and compare it with the performance of the other algorithms that do not utilize a channel splitting technique.

Algorithm 1 BL-MAC-IS: Start from the I-th anchor

Input: distances between collision-risk transmitting nodes, d_{ij},
maximum transmission range, D,
multi-channel data-rate deficiency, $\alpha^{(1:M)}$,
Maximum data-rate in single channel, R,
Packet size, b for $k = 1, 2, ..., K$,
Number of channel-splitting, M.

Output: waiting times before packet transmission, w_k for $k = 1, 2, ..., K$,
channel in which each node has to transmit its packet, m_k,
Task duration, $T_{\text{broadcast}}$.

Set all the waiting times to zero: $w_k = 0$, for $k = 1, 2, ..., K$,
Set all entries of $\mathbf{W}_{K \times M}$ to zero.
Set $m = 1$.
Set the packet lengths, $t_M^{\text{p}} = \frac{Mb}{R^{(1:M)}}$
Set $\Omega = \{1, 2, ..., K\}$.
Start with the pre-defined anchor index I, $j = I$,
for $k = 2$ to $K - 1$ **do**
Remove j-th anchor from the network: $\Omega = \Omega - \{j\}$
Find the collision-risk neighbors of the j-th anchor, and modify their waiting time to eliminate possible collisions:
for $i \in \Omega$ **do**
if $d_{ij} \leq 2D$ **then**
if L-MAC **then**
if TDMA-based **then**
$[\mathbf{W}]_{i,m} = \max([\mathbf{W}]_{j,m} + t_M^{\text{p}} + \frac{D}{c}, [\mathbf{W}]_{i,m})$
else
$[\mathbf{W}]_{i,m} = \max([\mathbf{W}]_{j,m} + t_M^{\text{p}} + \frac{\tilde{d}_{ij}}{c}, [\mathbf{W}]_{i,m})$
end if
else if B-MAC **then**
if $d_{ij} \leq D$ and $\frac{d_{ij}}{c} - |[\mathbf{W}]_{j,m} - [\mathbf{W}]_{i,m}| < t_M^{\text{p}}$ **then**
for $p = 1$ to M **do**
$[\mathbf{W}]_{i,p} = \max([\mathbf{W}]_{j,p} + t_M^{\text{p}} + \frac{\tilde{d}_{ij}}{c}, [\mathbf{W}]_{i,p})$
end for
else
$[\mathbf{W}]_{i,m} = \max([\mathbf{W}]_{j,m} + t_M^{\text{p}} + \frac{\tilde{d}_{ij}}{c}, [\mathbf{W}]_{i,m})$
end if
end if
end if
end for
Select the anchor with the minimum waiting time:
$[j, m] = \arg \min_{i \in \Omega,\ m \in \{1 \text{ to } M\}} [\mathbf{W}]_{i,m}$
end for
Compute the waiting times of each anchor and its channel
for $k = 1$ to K **do**
$w_k = \min_{m=1 \text{ to } M} [\mathbf{W}]_{k,m}$,
$m_k = \arg \min_{m=1 \text{ to } M} [\mathbf{W}]_{k,m}$.
end for
Compute the broadcasting task duration: $T_{\text{broadcast}} = \max_{k=1 \text{to } K} w_i + t_M^{\text{p}}$

6. NUMERICAL RESULTS

In Fig. 3, four transmitting nodes are going to broadcast their packets in the network. The packet length with full data rate, $t_1^{\text{p}} = \frac{b}{R}$, is the same for all these nodes, and their distances to each other are depicted in the figure. Based on their mutual distances and the packet length, it is shown which nodes can split the channels among themselves in a B-MAC scenario (see (11)). For instance, for two-channel splitting, nodes (1,2), (1,3), (1,4), (2,3),(2,4), and (3,4) can be considered as candidates. The number of possible solutions (combinations and permutations) for two-channel splitting is $6 \times 3!$ (two single channel transmissions and one double-channel transmission) plus $3 \times 2!$ (two subsequent double channel transmissions).

The optimum solution can be obtained by evaluating all possible solutions for all possible number of channels. We have done this for the setup illustrated in Fig. 3, and compared the optimum solution with other existing algorithms in Fig. 4. It can be seen that the gap between the proposed greedy algorithm and the optimum solution is not large, and the performance of the greedy algorithm is better than the ones which

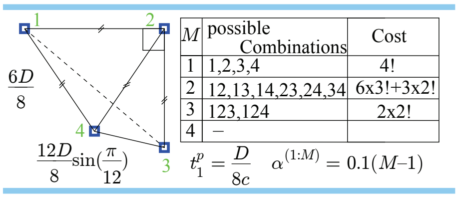

Fig. 3. An example of a fully connected network. Here, a solid edge connecting two nodes indicates that they are within the communication range of each other, and the dashed one indicates they are collision-risk but are not within the communication range of each other.

use a single channel approach.

In another simulation, we consider the parameters of an actual acoustic modem (S2CR-18/34) from Evologics [13]. The simulation set-up is as follows. For the computation of each point in the following figures, we average the solution over 10^3 independent Monte Carlo runs. The full data rate is $R = 13.9$kbps, the number of bits in each packet is $b = 2085$ bits ($t_1^{\mathrm{p}} = 150$ms), the total available bandwidth is $B = 16$KHz, the guard band between two adjacent channels in channel splitting is 1.6KHz, and therefore $\alpha^{(1:M)} = 0.1(M-1)$. The maximum transmission range of each anchor is assumed to be $2c$ meters. There are $N = 300$ nodes in the network, and $K = 16$ of them are going to broadcast their packets. The positions of the nodes are assumed to be uniformly distributed over the operating area with dimensions $d_x = d_y = 5c$ and $d_z = c$.

In Fig. 5, the performances of the proposed algorithms are compared with OCSMA, and the traditional TDMA-based system [14]. In OCSMA, the nodes transmit their packets one after another, and it does not support simultaneous transmission. On the other hand, for TDMA, simultaneous transmission between the nodes that are not in the collision range of each other is possible. In this simulation, the performances of the algorithms are evaluated over different values for the maximum transmission range. As the maximum transmission range increases, two phenomena adversely affect the performance of the TDMA system; first, more transmitting nodes become collision-risk, and second the time-slot duration gets larger. The time-slot duration does not have any impact on the distance-aware algorithms (OCSMA and the proposed ones), but the average number of collision-risk neighbors does. Moreover, it can be seen that as the network becomes more connected, the performance of the single-channel MAC algorithms approach the performance of OCSMA, and when the network if fully connected they perform the same. Still, it can be seen that the performances of the multi-channel algorithms are always better than their single channel opponents.

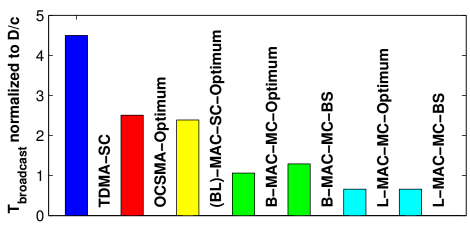

Fig. 4. Comparison of the single and multichannel algorithms in the network setup of Fig. 3.

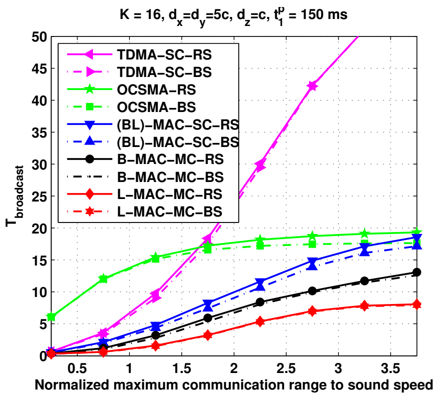

Fig. 5. Comparison of the single and multi channel algorithms with OCSMA and traditional ones.

7. CONCLUSION

In this article, we have introduced the state-of-the-art concept of dynamic channel-splitting in UASNs which utilizes only the position information of the transmitting nodes. It jointly determines the number of orthogonal channels, and schedules the packet transmission over the nodes and channels in a way that minimizes the total collision-free transmission time of all packets. We have shown that the problem of minimizing the duration of the packet transmission is NP-hard, and therefore not tractable. Afterwards, we have introduced a suboptimum algorithm for this problem, and have shown that it works much better than the scheduling algorithms which use a single channel for broadcasting. In the future, we are going to extend the proposed algorithm to a more elaborate one which could perform near optimal and be implemented in a distributed way.

REFERENCES

[1] P. Huang, L. Xiao, S. Soltani, M. Mutka, and N. Xi, "The evolution of MAC protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, to be published, Early Access.

[2] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 84–89, Jan. 2009.

[3] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC protocol for underwater acoustic networks," in *Proc. OCEANS 2006 - Asia Pacific*, 2006, pp. 1–7.

[4] Lu Hong, Feng Hong, Zhongwen Guo, and Xiaohui Yang, "A TDMA-based MAC protocol in underwater sensor networks," in *Proc. 4th Int. Conf. Wireless Communications, Networking and Mobile Computing WiCOM '08*, 2008, pp. 1–4.

[5] Yang Guan, Chien-Chung Shen, and J. Yackoski, "MAC scheduling for high throughput underwater acoustic networks," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, 2011, pp. 197–202.

[6] S. Shahabudeen, M. Chitre, and M. Motani, "A multi-channel MAC protocol for AUV networks," in *Proc. OCEANS 2007 - Europe*, 2007, pp. 1–6.

[7] Dario Pompili, Tommaso Melodia, and Ian F. Akyildiz, "A distributed CDMA medium access control for underwater acoustic sensor networks," *in Proc. Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Corfu, Greece*, 2007.

[8] Z. Zhou, Z. Peng, J.H. Cui, and Z. Shi, "Analyzing multi-channel MAC protocols for underwater acoustic sensor networks," Tech. Rep., UCONN CSE Technical Report: UbiNet-TR08-02, August 2008., 2009.

[9] Yin-Jun Chen and Hao-Li Wang, "Ordered CSMA: a collision-free MAC protocol for underwater acoustic networks," in *Proc. OCEANS 2007*, 2007, pp. 1–6.

[10] Wouter van Kleunen, Nirvana Meratnia, and Paul J. M. Havinga, "Mac scheduling in large-scale underwater acoustic networks," *Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS 2011), Sevilla, Spain*, pp. 27–34, July 2011.

[11] Hamid Ramezani and Geert Leus, "L-MAC: Localization packet scheduling for an underwater acoustic sensor network," *Accepted in Proc. of the International Conference on Communications, ICC2013*, 2013.

[12] Vasek Chvtal & William J. Cook David L. Applegate, Robert E. Bixby, *The Traveling Salesman Problem: A Computational Study*, Princeton University Press. ISBN 978-0-691-12993-8., 2006.

[13] *Evologics, Underwater acoustic modems, S2CR 18/34*.

[14] R. Ramaswami and K. K. Parhi, "Distributed scheduling of broadcasts in a radio network," in *Proc. Eighth Annual Joint Conf. of the IEEE Computer and Communications Societies. Technology: Emerging or Converging IEEE INFOCOM '89*, 1989, pp. 497–504.