MAC Scheduling for High Throughput Underwater Acoustic Networks

Yang Guan  Chien-Chung Shen
Department of Computer and Information Sciences
University of Delaware, Newark, DE, USA
{yguan,cshen}@cis.udel.edu

Abstract—Underwater acoustic networks (UWANs) have emerged as the primary tool to monitor and act upon the well-being of marine environment. However, the significantly slower propagation speed of acoustic signals, in contrast to RF signals, introduces the spatio-temporal uncertainty, which makes existing medium access control (MAC) solutions for terrestrial RF wireless networks not suitable for UWANs. In this paper, we investigate transmission scheduling for time-based MAC protocols and design scheduling algorithms that take advantage of the long propagation delay of acoustic signals to facilitate concurrent transmission and reception of acoustic communications. Specifically, we specify the constraints that MAC protocols need to satisfy to avoid collisions, and model these constraints into a conflict graph. We show that the scheduling problem is equivalent to the Traveling Salesman Problem (TSP) over the conflict graph, and demonstrate that simple heuristics can provide significant network throughput improvement.

I. INTRODUCTION

More than 70% of the Earth surface is covered by water, and underwater acoustic networks (UWANs) have emerged as the primary tool to monitor and act upon the well-being of marine environment for operations such as environmental monitoring, resources exploration, and early disaster warning [1], [2]. Since radio frequency (RF) electromagnetic signals do not propagate well in the underwater environment, UWANs use acoustic wave as the communication carrier [3], which differs fundamentally from RF signals in several ways. Foremost, the propagation speed of acoustic signals (1500 m/s) is roughly five orders of magnitude slower than that of RF signals. In addition, an effective transmission range of acoustic signals can reach several kilometers, which is much longer than that of typical Wi-Fi networks. However, in the state-of-the-art, the bandwidth of acoustic communications is relatively low, only accommodating transmission rate of less than 50 kbps. These characteristics challenge the networking protocols of UWANs.

The major function of a Medium Access Control (MAC) protocol over a shared channel is to coordinate the transmissions from different nodes to reduce the chance of collisions. In RF wireless networks, due to the negligible propagation delay, concurrent transmissions by two (or more) nearby nodes on the same channel nearly always collide. Therefore, classical MAC protocols mitigate collisions by enforcing no more than one transmitter within a receiver’s carrier sense range to transmit at any time.

In contrast, the propagation delay of acoustic signals in UWANs becomes much more significant, which can be equal to or even exceed the transmission time of data frames. When coordinating the transmissions of MAC-layer frames, the distance between different transmitters and the receiver (or propagation delay) should be taken into account so that signals arriving from different transmitters will not overlap at the receiver around the same time. Such issue is termed the Spatio-Temporal Uncertainty [4]. Fig. 1(a) depicts a 3-node UWAN topology, where the integer next to each link denotes its propagation delay in unit of time. Figs. 1(b) and 1(c) depict two different schedules of two transmissions A→B and C→B, each with a transmission time of one unit time. The schedule depicted in Fig. 1(b), although valid for terrestrial RF networks as nodes A and C do not transmit simultaneously (i.e., exclusive access), results in collision at the receiver node B in the context of UWANs as the two acoustic signals arrive at the receiver node B at the same time. In contrast, in Fig. 1(c) where nodes A and C transmit simultaneously, no collision occurs as the propagation delay interleaves two arriving signals at receiver node B.

Due to the non-negligible propagation delay incurred in UWANs, exclusive access is actually not necessary for collisions avoidance. Instead, the successful reception of a transmission by the intended receiver must only be separated in time from the reception of any interfering signals by that
receiver. This enables concurrent transmissions from nodes in the same carrier sensing range of the receiver as long as the different propagation delays can result in the different arrival times of the different signals at the receiver. Concurrent transmissions may reduce the length of idle periods of both transmitters and receivers, and hence boosts network throughput and channel utilization. Consider again the topology in Fig. 1(a). The transmission schedule depicted in Fig. 1(d), where 10 frames are successfully sent and received within a schedule of 9 units of time, results in no collisions. Such performance is attributed to the fact that interference can be “stacked;” for instance where node B, at time unit 5, simultaneously receives interference from frames \#5 (sent by node C) and \#6 (sent by node A).

By observing Fig. 1(d), in UWANs, it can sometimes be possible for (1) two nodes to transmit concurrently (e.g. frames \#1 and \#2), (2) a node to transmit while receiving interference from other frames intended for other nodes (e.g. frame \#6 is sent while node A receives interference from frame \#2), and (3) different nodes to receive different frames at the same time (e.g. frame \#4 received by node A and frame \#3 received by node B), all without any collisions [5].

Note that such concurrent actions are not possible as a general rule in UWANs, but only in specific situations such as Figs. 1(c) and (d), where the propagation delays allows nodes to receive the transmitted signals at non-conflicting times. In this paper, we investigate transmission scheduling for time-based MAC protocols and design scheduling algorithms that take advantage of the long propagation delay of acoustic signals to facilitate concurrent transmission and reception of acoustic communications in order to improve network throughput. In particular, we specify the constraints that MAC protocols need to satisfy to avoid collisions, and model these constraints into a conflict graph. We show that the scheduling problem is equivalent to the Traveling Salesman Problem (TSP) over the conflict graph, and demonstrate that simple heuristics can significantly improve network throughput.

In the remainder of the paper, we first review related work in Section II, and then formulate the MAC scheduling problem for UWANs in Section III. We reduce the formulated scheduling problem into the Traveling Salesman Problem (TSP) [6], and prove its NP-completeness in Section IV. We develop heuristics to solve the formulated scheduling problem, and evaluate their performance in Section V. Section VI concludes the paper.

II. RELATED WORK

UW-FLASHR [5] is a distributed, time-based MAC protocol supporting isochronous traffic, where data from each user is of a constant size generated at a constant interval, and each user may select a distinct size and interval. UW-FLASHR does not require tight clock synchronization nor accurate propagation delay estimation. As a time-based protocol, UW-FLASHR operates over cycles of time, where each cycle is divided into an experimental portion and an established portion. To send data, a node requests a new time slot by sending a data frame at a random time in the experimental portion of each of several consecutive cycles. However, as each node contends to allocate a time slot by randomly choosing a transmitting time and checking to see whether such a transmission incurs any collision, UW-FLASHR gradually constructs a loose transmission schedule in a distributed manner so that time gaps may exist between transmissions (time slots).

ST-MAC [7] models the problem of transmission scheduling in UWANs using a weighted, directed conflict graph. ST-MAC schedules transmissions by assigning a color (an integer) to each edge in the conflict graph. Unlike the traditional vertex-coloring problem, the difference of colors between adjacent vertices must be larger than the weight of edge between the corresponding vertices in the conflict graph. Therefore, existing heuristics cannot be easily applied to this formulation. Moreover, in order to make vertex-coloring algorithms work, they have to divide transmissions into multiple unit size slots, and force the weight of edges of a conflict graph to be integer.

In our work, we consider MAC scheduling in one-hop acoustic networks where one node may directly communicate with another node. However, the proposed algorithm can be easily extended to multi-hop scenarios by properly constructing the conflict graph. We reduce the scheduling problem to the standard TSP so that existing heuristics are can be directly applied. Unlike TDMA, we do not divide a frame into multiple equal-size time slots so that scheduled transmissions can start at any time (instead of only on the boundary of slots), which further improves channel utilization as well as network throughput.

III. FORMULATION OF UWAN MAC SCHEDULING

In this section, we first define the UWAN MAC scheduling problem. We then present the constraints for resolving transmission conflicts in UWANs. Due to the spatial-temporal uncertainty, the constraints required to resolve transmission conflicts in UWANs differ from those in terrestrial RF networks. We formulate the UWAN MAC scheduling problem into a Mixed Integer Linear Programming (MILP) model which can provide us with optimal solutions to achieve the highest network throughput. Since the scheduling problem is NP-complete, solving it via MILP is not practical for large scale topology. However, we run MILP solver for small networks to compare with the performance of heuristic solutions we employ.

A. Problem Definition

To define the MAC scheduling problem for UWANs, we assume the existence of a base station which collects information about the transmission tasks between nodes and the propagation delays between nodes, and computes the transmission schedules for each node. We model an underwater acoustic network as a directed graph \( G = (V, E) \), where each vertex \( v \in V \) represents a node in the network capable of both transmission and reception of acoustic signals, and each edge \( e \in E \) represents an acoustic link between two different nodes. Since each node can hear the signal from any other node, \( G \) is fully
connected. A weight function $W$ maps each edge in $E$ into a real number, representing the propagation delay, which is calculated by dividing the distance between the corresponding transmitter and receiver by the propagation speed. However, a path along which acoustic signals propagate can be affected by factors such as salinity and temperature of ocean water, and the time an acoustic signal travels from one node to another might not necessarily be equal to the time that the acoustic signal travels in the reverse direction. We overcome this problem by adding guard time between two adjacent transmissions.

The MAC scheduling problem concerns with a set of transmission tasks $\Delta$, where each element $\delta$ in $\Delta$ is fully specified by its source ($\delta.src$), destination ($\delta.dst$), transmission duration ($\delta.duration$), and start transmitting time ($\delta.start$). In the defined MAC scheduling problem, the first three items of each transmission task are given as input parameters, and the paper presents effective methods to compute a proper start transmitting time for each transmission task so as to maximize network throughput.

The base station collects the source, destination and duration information of each transmission task. The base station takes $G$, $W$ and $\Delta$ as input to the scheduling algorithm which computes the time slot (start transmitting time) to each transmission task. The objective of such a scheduling algorithm is to avoid conflicts between transmission tasks as well as to maximize the network throughput. The schedules assigned are not preemptive, so that the scheduling algorithms do not split transmission tasks into pieces and assign them different slots. We also do not make any assumption the transmission duration be a multiple of unit time slots.

### B. Conflict-free constraints

The goal of MAC scheduling is to coordinate how nodes access the physical channel to avoid transmission conflicts, while maximizing throughput. As discussed previously, unlike terrestrial RF networks, in UWANs, two nodes within the carrier sensing range of receivers can transmit at the same time without causing collision as long as the arriving signals do not overlap at the receiver. We aim to design algorithms to schedule transmitting times so that all the conflicts at the receivers are avoided and we call such a schedule valid.

In the remainder of this subsection, we list the conflict-free constraints that a valid schedule needs to satisfy, in he context of spatio-temporal uncertainty.

We discuss how two transmission tasks (denoted as $\delta_i$ and $\delta_j$) may conflict with each other. We define the nodes where conflicts occur as conflicting nodes. Based on whether a conflicting node is a transmission source or a transmission destination, we divide conflicts into four categories: (1) TX-TX conflict (Fig. 2(a)); (2) TX-RX conflict (Fig. 2(b)); (3) RX-RX conflict (Fig. 2(c)) and (4) RX-Interface conflict (Fig. 2(d)).

It is worth pointing out that interference signals do not cause conflicts when they arrive at transmitting nodes. We first give constraints for each conflict case, and then show that all the cases can be treated in an unified way and give two inequality equations to resolve all possible conflicts.

- **TX-TX conflict**: This case occurs when two transmissions share the same source ($\delta_i.src = \delta_j.src$) but with different destinations (common source and common destination transmissions can be aggregated and considered as a single transmission). Since we assume each node is equipped with one physical interface, a node can not transmit more than one packet simultaneously. For two packets with a common source, they must be interleaved and either Eq. (1) or Eq. (2) must be satisfied. Eq. (1) considers the case that $\delta_i$ is transmitted first and Eq. (2) takes care of the other case.

  $$\delta_i.start + \delta_i.duration \leq \delta_j.start$$  \hspace{1cm} (1)
  $$\delta_i.start \geq \delta_j.start + \delta_j.duration$$  \hspace{1cm} (2)

- **TX-RX conflict**: A node cannot receive signals destined to itself while it is transmitting another signal. A valid schedule must guarantee that incoming signals arrive at a conflicting node after the conflicting node finishes transmitting (Eq. (3)) or ends before the conflicting node starts transmitting (Eq. (4)).

  $$\delta_i.start + \delta_i.duration + W(<\delta_i.src, \delta_i.dst>) \leq \delta_j.start$$  \hspace{1cm} (3)
  $$\delta_i.start + W(<\delta_i.src, \delta_i.dst>) \geq \delta_j.start + \delta_j.duration$$  \hspace{1cm} (4)

- **RX-RX conflict**: In this type of conflict, two transmissions share a common node as the destination, which is the conflict node. Different from TX-TX conflict, a valid schedule needs to take the propagation delay into consideration. For example, in Fig. 2(c), both $\delta_i$ and $\delta_j$ have data for node $A$. When scheduling the transmitting times, we need to consider the propagation delay from $B$ to $A$ and the propagation delay from $C$ to $A$. In general, to resolve RX-RX conflicts, one of the following two inequality equations must be satisfied:

  $$\delta_i.start + \delta_i.duration + W(<\delta_i.src, \delta_i.dst>) \leq \delta_j.start + W(<\delta_j.src, \delta_j.dst>)$$  \hspace{1cm} (5)
\[ \delta_i \text{.start} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) \geq \delta_j \text{.start} \]
\[ + \delta_i \text{.duration} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) \] (6)

- **RX-Interference** conflict: in this situation, signal from one transmission interferes with signal of the other transmission at destinations. Different from previous situations where transmissions share nodes as either source or destination, here two transmissions are disjoint and both destinations are conflicting nodes. In terrestrial networks where FDMA or CDMA is available, nodes are able to differentiate signals destined to them from interfering signals. However, currently UWANs do not generally support FDMA (due to limited bandwidth) and CDMA (due to hardware complexity) and we must schedule the transmitting times such that interference signals do not arrive at the conflicting nodes when they are receiving. Either Eq. (7) or Eq. (8) must be satisfied to resolve conflict at \( \delta_j \text{.dst} \). Constraints to resolve conflict at \( \delta_j \text{.dst} \) are similar but not presented due to the space limitation.

\[ \delta_i \text{.start} + \delta_i \text{.duration} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) \leq \delta_j \text{.start} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) \] (7)

\[ \delta_i \text{.start} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) \geq \delta_j \text{.start} + \delta_j \text{.duration} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) \] (8)

**C. Mixed Integer Linear Programming Model**

Examining Eqs.(1–8), we can find that Eq. (7) is actually a generalized form of Eqs. (1), (3), and (5) and Eq. (8) is a generalized form of Eqs. (2), (4), and (6) if we define the propagation delay from one node to itself as zero, i.e. \( \forall X, W(< X, X >) = 0 \). This provides us an easy way to formulate the scheduling problem into an MILP problem. We also notice that popular MILP solvers such as GLPK [9] do not support disjunction form of constraints as Eq. (7) and Eq. (8). We overcome this issue by introducing a new binary parameter \( b_{ij} \) and rewrite the constraints as:

\[ \delta_i \text{.start} + \delta_i \text{.duration} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) < \delta_j \text{.start} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) + b_{ij} * C \] (9)

\[ \delta_i \text{.start} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) > \delta_j \text{.start} + \delta_j \text{.duration} \]
\[ + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) + (b_{ij} - 1) * C \] (10)

where \( C \) is a constant that is large enough to guarantee when \( b_{ij} = 0 \), Eq. (9) is automatically satisfied and when \( b_{ij} = 1 \), Eq. (10) is true all the time. \( b_{ij} \) actually defines the order among transmissions. For example, when \( b_{ij} = 1 \), the conflicting node will serve \( \delta_i \) before \( \delta_j \).

We also need to redefine the frame size. In terrestrial RF networks, frame length is usually set to the number of transmission slots in one epoch. In UWANs, due to the large propagation delay, a new frame can not begin immediately after the last transmission finishes otherwise inter-frame conflicts may happen. One method to avoid inter-frame conflicts is to append large guard time at the end of each frame to make sure the channel become clean when the next frame starts. This can greatly reduce network throughput since the guard period needs to be as long as the maximum propagation delay. We define the frame size as the minimum delay between two successive transmissions to avoid inter-frame conflicts and try to minimize the frame size since it is inversely proportional to network throughput. The constraints to resolve inter-frame conflicts are:

\[ \delta_j \text{.start} + \delta_j \text{.dur} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) < \delta_i \text{.start} \]
\[ + \text{FrameSize} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) \] (11)

We formulate the scheduling problem into MILP as follows:

**Parameters:**
- \( \delta \text{.src} \in V, \delta \text{.dst} \in V, \delta \text{.dur} \in R^+, \forall \delta \in \Delta; \)
- \( W(< l, m >) \in R^+, \forall l, m \in E; \)

**Variables:**
- \( \text{FrameSize}; \)
- \( \delta \text{.start} \in R^+, \forall \delta \in \Delta; \)
- \( b_{ij}, \delta_i, \delta_j \in \Delta; \)

**Minimize:** \( \text{FrameSize}; \)

**Subject to:** \( \forall \delta_i, \delta_j \in \Delta, \)

\[ \delta_i \text{.start} + \delta_i \text{.duration} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) < \delta_j \text{.start} + \delta_j \text{.duration} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) + b_{ij} * C \] (12)

\[ \delta_i \text{.start} + \delta_i \text{.duration} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) > \delta_j \text{.start} + \delta_j \text{.duration} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) + (b_{ij} - 1) * C \] (13)

\[ \delta_i \text{.start} + W(< \delta_i \text{.src}, \delta_i \text{.dst}>) > \delta_j \text{.start} + \delta_j \text{.duration} + W(< \delta_j \text{.src}, \delta_j \text{.dst}>) + (b_{ij} - 1) * C \] (14)

IV. TSP PROBLEM CONSTRUCTION

In this section, we reduce the UWAN scheduling problem into the Traveling Salesman Problem (TSP) [6]. Doing so not only proves that the UWAN scheduling problem is NP-complete, but also facilitates the application of existing heuristic TSP solvers to solve the UWAN scheduling problem.

We first define the conflict free delay \( (C \text{FD}) \) of transmission \( \delta_j \) after transmission \( \delta_i \) (denoted as \( C \text{FD}_{\delta_i \rightarrow \delta_j} \)) as the minimum time \( \delta_j \) must wait after \( \delta_i \) starts transmitting. If \( \delta_j \)
starts transmitting earlier than this delay, conflict will occur. From the discussion above, $CFD_{\delta_i,\delta_j}$ is defined as follows.

$$CFD_{\delta_i,\delta_j} = \max\{\delta_i.dur + W(<\delta_i.src,\delta_i.des>),\delta_i.dur + W(<\delta_j.src,\delta_j.des>), -W(<\delta_j.src,\delta_i.des>), -W(<\delta_j.src,\delta_j.des>)\}$$

(18)

In order to reduce the UWAN scheduling into TSP, we first construct a conflict graph $CG(V', E')$ based on the network topology $G(V, E)$, propagation delay matrix $W$ and transmission pairs $\Delta$. For each $\delta \in \Delta$, there is a vertex $v' \in V'$, i.e., we map each transmission $\delta$ into a vertex in $V'$. For the ease of discussion, we define this one-to-one mapping as $M$ so that $v'_i = M(\delta_i)$. The edge between vertices in $CG$ is direct and meant to describe the conflict relation between two transmissions, so that $E'$ is defined that $\forall v'_i, v'_j \in V'$, there is an edge $<v'_i, v'_j> \in E'$. We also define a weight function $W'$ on each $e' \in E'$ such that $W'(<v'_i, v'_j>) = CFD_{\delta_i,\delta_j}$.

This ends our construction of a conflict graph. In the remainder of this section, we describe the properties of CG, and explain how CG would assist us to solve the scheduling problem.

- Each conflict graph is a complete graph. We assume that all the nodes of a network are within a single carrier sense range so that every node can receive signal from every other node. Also each transmission can potentially conflict with any other transmissions. Thus, CG must be fully connected to express all the conflict relations between transmissions.

- Each edge in CG is asymmetric, i.e. $W'(<v'_i, v'_j>)$ is not necessarily equal to $W'(<v'_j, v'_i>)$. This can be proved by using the definition of CFD and Eq. (18).

- The triangle inequality is not satisfied here. We prove this by giving an example. Fig. 3(a) gives a network with 4 nodes and 3 transmissions: $\delta_1 : A \rightarrow D$, $\delta_2 : B \rightarrow C$ and $\delta_3 : C \rightarrow D$. Fig. 3(b) shows a partial conflict graph built from the network topology in Fig. 3(a) where only a subset of edges are drawn. The weight of each edge in the conflict graph can be computed via Eq. (18) and we can show

$$W'(<\delta_3, \delta_1>) + W'(<\delta_1, \delta_2>) > W'(<\delta_3, \delta_2>)$$

(19)

which indicates the triangle inequality does not hold in this conflict graph.

After building the conflict graph, we can schedule all the transmissions by visiting all the nodes sequentially. This forms a tour in the conflict graph and the length of the tour is equal to the final frame size. To increase the network throughput, we want to reduce the frame size as much as possible, which means we need to find a tour with the minimum length. This is equivalent to TSP. As proved above, triangle inequality does not hold in the constructed conflict graphs, so that it is unlikely to develop approximation algorithms with constant approximation ratio. However, we can apply the existing heuristics to TSP to efficiently solve the scheduling problems in UWANs. We show the performance of these heuristic solutions in the next section.

V. PERFORMANCE EVALUATION

In this section, we first review three heuristics used to solve TSP. We then describe the simulation setup used to evaluate these heuristic solutions. We run simulations with different networking scenarios and compare the performance of different heuristics against the optimal scheduling.

A. Heuristics to TSP

Since the scheduling problem for UWANs has been reduced into the extensively studied TSP Problem, we will apply existing heuristics to solve the scheduling problem. In this paper, we choose to implement the following three heuristics:

- Nearest Neighbor (NN): NN starts from a node in the conflict graph and iteratively jumps to a next unvisited neighbor with the minimum distance, which eventually returns to the starting node. For better performance, the NN heuristic we used here finds multiple tours by setting each node to be the starting one and return the tour with the minimum length.

- Nearest Neighbor with Random Start (NNRS): NNRS is very similar to NN but only randomly chooses one node as the starting node. This can reduce the complexity by a factor of $O(|\Delta|)$.

- Minimum Spanning Tree (MST): As the weight matrix of a conflict graph is asymmetric, before applying the MST heuristic, we first transform the asymmetric graph into a symmetric graph by splitting each node into an incoming node and an outgoing node. Then a minimum spanning tree can be found by using either the Prim or the Kruskal algorithm. A tour can be formed by traversing over the spanning tree in preorder manner.

After a tour in the conflict graph is found, we assign the transmitting time of each transmission as the distance from the starting node to the corresponding node along the tour and the length of tour is used as the frame size.

As a baseline, we also show the performance of a naive scheduling which simply assigns transmission time in a conflict-free fashion, and does not change the order of transmissions to archive high channel utilization.
B. Simulation setup

We use QualNet 3.7 [10] to evaluate the performance of different heuristics. To simulate acoustic channels, we modified QualNet to include spherical path loss and Thorp attenuation. We use a terrain of 1000 meters × 1000 meters and randomly place 30 nodes (small networks) or 100 nodes (large networks). As mentioned above, the transmission range of acoustic signals can reach several kilometers, so in our simulated networks, all nodes can hear from one another. The propagation speed is 1500 m/s and the transmission rate is 15 kbps. We simulated both small networks and large networks. For small networks, we set up to have 10 transmissions with random source and destination pairs, which allows us to run GLPK [9] to solve MILP and obtain the optimal transmission schedule in reasonable time. For large networks, up to 30 transmissions are generated and transmission schedule is obtained from heuristics described above. The transmissions are generated as Constant Bit Rate (CBR) application with payload size of either 512 bytes or 1024 bytes. Each network configuration (number of nodes, number of transmissions and payload size) is simulated 40 times with different node placements and different source-destination pairs to calculate the average network throughput.

C. Performance Comparison and Analysis

Figs. 4(a)-(d) show the network throughput achieved by heuristics described above. We can see from Figs. 4(a) and 4(b) that the optimal scheduling from solving the MILP model archives the highest throughput. This is obvious because the MILP model selects the best schedule from |Δ|! possible schedules. The throughput of the optimal scheduling increases with the number of transmissions, due to the fact that when there are only a few transmissions, it is difficult to find transmitters to transmit concurrently while avoiding collisions, so that most nodes will remain idle when signals are propagating. However, the throughput of the optimal scheduling is still 32% higher than that of naive scheduling.

Comparing Fig. 4(a) and Fig. 4(b) tells us that transmissions with large payload (1024 bytes) shows better throughput than that with small payload (512 bytes). The reason is that the transmission time of larger frames is longer, while the propagation delay remains the same, so that the issue of spatio-temporal uncertainty becomes less significant. If we could transmit frames with megabyte payload, the propagation delay becomes tiny in comparison with transmission delay, and we come back to a situation that is similar to terrestrial networks. Figs. 4(c) and 4(d) confirm this point.

The NN heuristic archives similar throughput to the optimal scheduling. This makes NN an ideal candidate to schedule large scale UWANS and/or UWANs with heavy traffic, since the time complexity of NN is $O(|\Delta|^2)$ while that of the optimal scheduling is exponential due to its NP-completeness.

VI. CONCLUSION

This paper investigate the problem of transmission scheduling for time-based MAC protocols that take advantage of the spatio-temporal uncertainty to exploit concurrent, conflict-free transmission and reception of acoustic communications in the context of UWANs. A conflict graph is constructed to express constraints used to avoid transmission collisions. Transmissions are then scheduled by running TSP heuristics over the conflict graph. The simulation results show that these heuristics can greatly improve network throughput.

REFERENCES