Scalable Adaptive Networking for the Internet of Underwater Things

Nils Morozs, *Member, IEEE*, Paul D. Mitchell, *Senior Member, IEEE*, and Roee Diamant, *Senior Member, IEEE*

***Abstract*—Internet-of-Underwater-Things (IoUT) systems comprising tens or hundreds of underwater acoustic communication nodes will become feasible in the near future. The development of scalable networking protocols is a key enabling technology for such IoUT systems, but this task is challenging due to the fundamental limitations of the underwater acoustic communication channel: extremely slow propagation and limited bandwidth. The aim of this article is to propose the JOIN protocol to enable the integration of new nodes into an existing IoUT network without the control overhead of typical state-of-the-art solutions. The proposed solution is based on the capability of a joining node to incorporate local topology and schedule information into a probabilistic model that allows it to choose when to join the network to minimize the expected number of collisions. The proposed approach is tested in numerical simulations and validated in two sea trials. The simulations show that the JOIN protocol achieves fast convergence to a collision-free solution, fast network adaptation to new nodes, and negligible network disruption due to collisions caused by a joining node. The sea trials demonstrate the practical feasibility of this protocol in real underwater acoustic network deployments and provide valuable insight for future work on the tradeoff between control overhead and reliability of the JOIN protocol in a harsh acoustic communication environment.**

***Index Terms*—Adaptive network, Internet of Things (IoT), medium access control (MAC), node integration, underwater acoustic network (UAN).**

I. Introduction

MODERN developments in underwater acoustic modem capabilities will make large-scale underwater acoustic networks (UANs) of the order of tens to hundreds of nodes feasible in the near future. Such large-scale UAN deployments will enable the extension of Internet-of-Things (IoT) technologies to the ocean environment. This new emerging IoT application is referred to as the Internet of Underwater Things (IoUT) [1]. The IoUT will have a wide range of applications, e.g., water quality monitoring [2], seismic monitoring [3], marine animal tracking [4], off-shore oil and gas asset monitoring [5], and ocean exploration using autonomous underwater vehicles (AUVs) [6]. Fig. 1 depicts several examples of UANs used for these IoUT applications, all of which will require networks comprising multiple underwater acoustic nodes communicating among themselves, e.g., to deliver the data to a surface buoy with a radio link to the control center onshore. In this article, the focus is on the adaptive formation of such IoUT networks, where nodes may independently join or leave the network.

To allow efficient integration of new nodes into an operating IoT network, typical solutions proposed for terrestrial radio involve a proportion of the communication resources reserved for control signaling, e.g., periodic beacons and association request/response packets [7], [8] or regular broadcast messages for neighbor discovery [9]–[11]. Such protocols are not feasible for implementation in UANs due to the extremely long propagation delays of acoustic waves (the sound speed is approximately 1500 m/s) and low available bandwidth (typically in the order of several kilohertz [12]). More specifically, the signaling overhead of these protocols would consume most of the network throughput capacity by introducing long idle times, while nodes wait for the roundtrip delay of the control packet exchanges [13]. Therefore, transferring such terrestrial network solutions to the underwater environment may provide scalability, but at the expense of significantly reducing the network capacity.

A class of medium access control (MAC) protocols most suitable for providing high network capacity via efficient channel utilization is based on packet scheduling, e.g., time-division multiple access (TDMA) [13], where the nodes are scheduled to transmit their data packets at particular times such that the packets arrive at the intended receivers without collisions. Such contention-free MAC protocols do not involve control signaling in order to establish collision-free links, e.g., request to send (RTS) and clear to send (CTS). Therefore, they are capable of achieving high throughput by timing the transmissions in a way that results in a stream of data packets separated by guard intervals at the intended receivers. However, the typical scheduling protocols that focus on providing high capacity in UANs are limited to fixed network sizes or particular connectivity patterns, allowing these protocols to exploit the knowledge of long propagation delays and topology sparsity to increase the throughput, e.g., [14]–[17].

Manuscript received November 10, 2019; revised February 5, 2020; accepted April 12, 2020. Date of publication April 17, 2020; date of current version October 9, 2020. This work was supported in part by the U.K. Engineering and Physical Sciences Research Council through the USMART under Project EP/P017975/1 and Full-Duplex project (EP/R003297/1), in part by the Royal Society International Exchanges Programme under Grant IES\R2\181125, in part by the Center for Cyber Law and Policy at the University of Haifa in conjunction with the Israel National Cyber Directorate in the Prime Minister's Office, and in part by the Israeli Ministry of Science under Grant 3-16728 and Grant 3-16573. *(Corresponding author: Roee Diamant.)*

Nils Morozs and Paul D. Mitchell are with the Department of Electronic Engineering, University of York, York YO10 5DD, U.K. (e-mail: nils.morozs@york.ac.uk; paul.mitchell@york.ac.uk).

Roee Diamant is with the Department of Marine Technologies, University of Haifa, Haifa 3498838, Israel (e-mail: roee.d@univ.haifa.ac.il).

Digital Object Identifier 10.1109/JIOT.2020.2988621

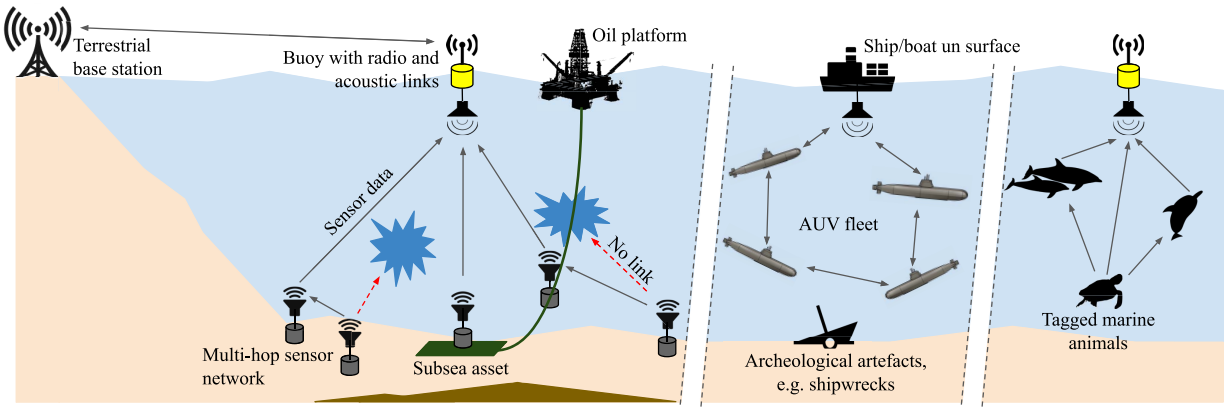


Fig. 1. Possible applications of the IoUT. The left panel shows fixed sensors reporting information to a common sink; the middle panel illustrates a fleet of AUVs communicating during a joint survey; and the right panel shows sensor acoustic tags attached to marine animals for migration and behavioral exploration.

Therefore, despite reducing the signaling overhead and providing higher capacity, these solutions are not sufficiently scalable to large-scale IoUT applications, where nodes may independently join or leave the network.

The main challenge of this article is to develop a MAC protocol that provides scalability without trading it off with the network throughput. In particular, the focus is on the problem of integrating new nodes into an already operating network, taking into account the limited communication resources and long propagation delays in IoUT, while minimizing network disruption and the delay the new node will experience. Since large-scale IoUT applications are targeted in this article, a hard constraint placed in this article is that only the core communication resources are used, i.e., the new node cannot rely on reserved time slots, frequencies or spreading codes to join the network. Therefore, the approach proposed in this article is scalable and applicable to any network topology, as the new node does not require any prior knowledge of the network size, topology, or packet schedule. Instead, the network setup assumes only knowledge of the basic communication properties, e.g., the ability to receive and decode the packets sent by other nodes in the network. As a test case, communication using TDMA is considered, which is widely used in UANs [13]. However, the approach here can also be extended to other types of protocols. In particular, the only limitation of the protocol proposed in this article is that the basic structure of the medium access protocol should be known to the new node joining the network. This does not refer to the per-node resource allocation, but rather to the structure of the communication frame. That is, the proposed approach fits the family of contention-free MAC protocols.

To join an IoUT network with minimal disruption, the new node starts with a promiscuous listening phase during which it learns who its one hop neighbor nodes are, and, by the content of their packet headers, who their neighbors are. Using this information, the new node builds a local connectivity map and, by solving an optimization problem, determines the best time to transmit a packet to minimize the likelihood of causing a collision. Then, relying on the notifications from its immediate neighbors that may detect and report collisions as part of the core MAC protocol, the new node iteratively acquires more information about the network until it converges to the best solution. The proposed protocol is taken one step further, and offers a fast means for the network to distribute the information about the new node, by detecting cases where the new node will not find any solution and proactively triggering a network update.

To the best of our knowledge, the proposed protocol is the first to suggest a solution for scalable network adaptation in IoUT that does not require any reserved resources. The contribution of this article is twofold.

1) An optimization framework for new nodes to join existing IoUT networks based on the information they gather locally.
2) A network joining (JOIN) protocol that incorporates this optimization framework to enable efficient node integration and network adaptation.

The proposed approach is tested in numerical simulations and validated in two sea trials. The results show that the JOIN protocol achieves significantly faster and less disruptive node integration than random opportunistic access and that it is scalable with the network size. Successful sea trials also show that JOIN is a feasible solution in real-world UAN deployments with different network topologies.

The remainder of this article is organized as follows. Section II gives a brief overview of the state of the art in scalable network protocols. Section III describes the system model considered in this article. Section IV presents the JOIN node integration protocol. Sections V and VI describe the results of numerical simulations and sea trials, respectively. Finally, Section VII concludes this article.

II. State-of-the-Art Overview

There is a large body of literature on protocols that enable the integration of new nodes into terrestrial radio networks. As resources in terrestrial networks are much less constrained than in UANs, most available solutions rely on dedicated beacons to provide potential new nodes with the information required to join the network. For example, in 4G LTE [7] networks, the base stations transmit regular beacons, including the information necessary for any potential new

node to synchronize to the network and use the appropriate time–frequency resources to send its association request. Similarly, in IEEE 802.11 Wi-Fi [18] networks, the access points transmit regular beacons that allow the new nodes to discover them and use appropriate transmission parameters to send their association request frame. There are also many beacon-based solutions proposed for *ad hoc* networks. For example, Ronai and Kail [9] proposed the SND protocol for Bluetooth *ad hoc* networks, where the nodes send beacon messages in pseudorandom time slots to enable neighbor discovery. The S-MAC [10] and PMAC [11] protocols proposed for wireless sensor networks involve all nodes sending periodic beacons to announce their sleep/wake-up timing, thus enabling both network discovery and adaptive scheduling. Similarly, the PISTONSv2 [19] protocol was designed to adjust the interval between the neighbor discovery probes to conserve energy; however, it still relies on the use of dedicated probes to achieve network scalability.

An alternative approach that eliminates the need for broadcasting beacons is reserving a part of the communication resources that are kept unused to allow any new nodes to join the network. Such is the adaptive TDMA approach, where the frame length is varied to accommodate changes in the network size, e.g., [20] and [21]. For example, the E-ASAP protocol [20] adaptively varies the number of slots per frame, while the first slot in each frame is kept free to enable any new node to join the network. Similarly, in the adaptive frame structure of the A-ADHOC protocol [21], some slots in a frame are marked as "available" to allow a new node to contend for access. The drawback of using reserved time slots for new node integration is the underutizalition of the channel airtime since most of the time these slots remain unused. Furthermore, the significant limitation of distributed frame adaptation at each node, such as that used in E-ASAP and A-ADHOC protocols, is that only mutually compatible frame lengths can be used, with the options typically restricted to powers of two ($2, 4, 8, 16\ldots$ slot frames). In the context of UANs, this approach is highly inefficient, since, due to the long propagation delays, the typical slot duration in UANs is significantly longer than the packet duration. Therefore, any redundancy in the frame structure, i.e., unused time slots, has a large negative impact on the network throughput.

Another adaptive scheduling solution is based on the hybrid TDMA/CSMA principle [22], e.g., the TDMA-ASAP protocol [23], whereby unused slots are detected by other nodes and opportunistically "stolen" for their own transmissions if needed. This approach is not feasible in UANs due to the carrier sensing delay caused by the slow propagation of acoustic waves. It is highly unlikely that a free slot can be detected and simultaneously reused unless the slot duration is artificially extended to allow this, which would significantly reduce the network throughput, similar to the adaptive TDMA protocols described above.

Little research on scalable networking protocols with node discovery and integration has been done specifically in the UAN domain. Most proposed protocols require a separate network discovery stage, where the nodes exchange broadcast messages to discover their neighbors and adapt to topology changes, e.g., [17] and [24]–[26]. The capability to adapt to new node arrivals is either not addressed or is achieved by disrupting the network operation and repeating the neighbor discovery process. For example, in DQA-MAC [17], any significant changes in the network topology would require a partial or full repeat of the network setup stage that involves the measurement of propagation delays and the assignment of a new packet schedule. Similarly, in the DIVE protocol [26], if a "Hello" packet is received from a new node, the other nodes restart the neighbor discovery process to update the network topology. Guo *et al.* [27] proposed an adaptive routing protocol that circumvents this problem by "piggy backing" the acknowledgment (ACK) packets to send the "Hello" messages for neighbor discovery. However, this protocol involves the "epidemic ACK" feature, which significantly increases the number of ACK packets in the network, thus still introducing a large signaling overhead to enable network adaptability. Another alternative solution to a dedicated network discovery stage is UWAN-MAC [28], where every node listens for "Hello" messages from potentially incoming new nodes for the remainder of the slot after its own transmission. However, this feature significantly extends the slot duration to include an idle listening time, which dramatically limits the network throughput. Therefore, UWAN-MAC is only applicable to low-duty-cycle networks, e.g., it is evaluated in [28] for a 0.2%–0.6% duty cycle range.

To address the scalability and adaptability of future IoUT networks, a gap in the available literature is identified. In particular, since the communication resources in UANs are highly limited, the solutions that use dedicated frequency/time/code resources introduce an overhead that is not scalable for large networks, e.g., they typically involve long idle times on the channel to allow the new nodes to transmit their "Hello" packets without collisions. Furthermore, the protocols that involve a separate network discovery phase to enable network adaptability, where nodes broadcast beacons in order to rediscover the topology, are too disruptive to normal network operation. A new protocol that enables network adaptability using only the core communication resources and without disrupting the normal network operation is therefore required.

III. System Model

A decentralized UAN that is already up and running is considered, where every node is aware of the current network topology and transmits its packets according to a mutually agreed schedule. As mentioned above, the general case of a TDMA schedule is presented, where each node is assigned at least one-time slot to transmit, and it may or may not reuse time slots with other, spatially separated nodes. In the illustrative example given in Fig. 2, there are six nodes running a spatial-reuse TDMA (SR-TDMA) protocol [29], such that in a 4-slot frame, N1 and N5 use slot 1, N2 and N6 use slot 2, N3 uses slot 3, and N4 uses slot 4. The new node, also referred to as the *joining node*, is connected to N1 and N2, and is not aware of the above schedule, nor of the existence of other nodes in the network. The frame size is not revealed to this node, and only a common time reference and the method to decode the packets are globally known. Similarly, the current

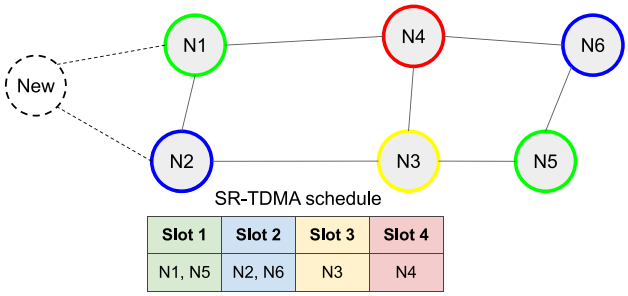


Fig. 2. Example of a network using the SR-TDMA protocol from [29]. Some nodes (N1 and N5 and N2 and N6) can reuse the same slot without interference. The problem investigated in this article is how to integrate new nodes into the network with minimal overhead and disruption.

network nodes, referred to as the *incumbent nodes*, are not aware of the existence of the new node. In the proposed setup, no dedicated time slots or control packets exchanges to accommodate joining node access requests are allowed. This constraint is to avoid the idle time on the channel, e.g., due to unused time slots or round trip delays of control packet exchanges, the duration of which would be significantly extended by the long propagation delays of acoustic signals. To further reduce the overheads, the schedule information is not broadcasted at any stage. Instead, every node, including the new nodes that may be deployed in the future, has an identical copy of a deterministic computer function that generates a packet schedule given a particular network topology, e.g., [30]. In this way, if the nodes keep track of the network topology updates locally, they are able to update their schedules independently without any communication overhead.

The solution proposed here is set by learning the one- and two-hop connections of the joining node. Therefore, as in any contention-free protocol with topology dependent resource allocation, the underlying assumption is that the communication links between the network nodes do not change during the joining procedure. Since the link connectivity in underwater networks is usually robust to small node displacements (other than in extreme cases, e.g., within harbors), slow mobility of the incumbent network nodes and the joining node is allowed. However, since the joining node is assumed to be aware of the structure of the resource allocation protocol, such mobility does restrict the use of range-dependent resources. For example, while spatial reuse of the TDMA slots is allowed, the duration of the time slots should remain fixed.

The aim of JOIN is to enable a new node to join this network either by finding a collision-free slot in the current schedule, or, if no such slot is available, by triggering a network update, allowing it to join without causing interference.

The following performance measures are considered.

1) *Number of collisions*, which quantifies the degree of disruption caused by the new node joining the network.
2) *Time until convergence*, which is measured as the number of slots until the joining node converges to a collision-free solution. This metric indicates how quickly the new node is able to become integrated into the network.
3) *Network adaptation time*, measured as the number of slots from the initial transmission by the joining node until every node in the network knows of its existence. This metric indicates how quickly the network can grow.

In this article, the focus is on the network growth and not on the scenario of nodes leaving the network. This is because the solution to the latter problem is much simpler than integrating new nodes. Specifically, it would involve the same network adaptation protocol as that proposed in Section IV-H, but instead of accommodating a new node into an updated schedule, the absence of a node is detected by its immediate neighbors and reported via the "network update" packets in the same way as the detected presence of the new node is reported.

IV. JOIN Protocol

In this section, the details of the JOIN protocol designed for integrating new nodes into an active UAN with an unknown topology and packet schedule are described. The protocol is described using the top-down approach: starting by formulating the optimization problem solved by the joining node to choose a slot for its transmission; followed by the description of the proposed iterative algorithm to solve this problem, taking into account the limited computational capacity of the joining node; followed by a description of the protocol implementation at the joining node and the incumbent nodes; and ending with a discussion of the practical considerations of the JOIN protocol.

A. Slot Selection Problem

The joining node chooses a slot for transmission by minimizing the *expected number of collisions*. This optimization problem is formulated as follows:

$$\arg \min_{s\in\{1\ldots N_{\text{slots}}\}} \quad \mathbb{E}[n_c[s]] \tag{1a}$$

$$\text{s.t.} \quad p_{\text{free}}[s] > 0 \tag{1b}$$

where $\mathbb{E}[n_c[s]]$ is the expected number of collisions the joining node will cause if it transmits in slot s; N_{slots} is the number of slots in the TDMA frame; and $p_{\text{free}}[s]$ is the probability that slot s is free, i.e., if the joining node transmits in it, it would not cause any collisions. Note that the term *probability* in this section is used to describe the probabilities *estimated by the joining node* based on its incomplete information rather the true probabilities of events according to the global system model. This is a global nonconvex optimization problem that can be solved using the branch-and-bound method since $\mathbb{E}[n_c[s]]$ is a nonlinear discontinuous function with the values varying between 0 and $N_{\max}$ depending on the global network topology unknown to the joining node. $N_{\max}$ is the maximum possible number of collisions that is equal to the number of the joining node's 1-hop neighbors $N_{1-\text{hop}}$. Assuming that every node is assigned a finite number of slots in the TDMA frame, the worst case time complexity of iterating over all elements in $\mathbb{E}[\boldsymbol{n_c}]$ and choosing the optimal slot is $\mathcal{O}(k_{\max} \times n) \approx \mathcal{O}(n)$, i.e., *linear time*, where $k_{\max}$ is the maximum number of slots that can be assigned to a single node. However, the linear time complexity of solving the optimization problem (1a), given the knowledge of $\mathbb{E}[\boldsymbol{n_c}]$ and $\boldsymbol{p_{\text{free}}}$, is negligible compared with the time complexity of

computing the values in $\mathbb{E}[\boldsymbol{n_c}]$ and $\boldsymbol{p_{free}}$ in the first place, as described in the rest of this section.

The expected number of collisions for a given slot s can be calculated as a weighted sum of all possible numbers of collisions that could occur

$$\mathbb{E}[n_c[s]] = \sum_{n=0}^{N_{\max}} (p(n \text{ collisions}) \times n) \tag{2}$$

where $p(n$ collisions$)$ is the probability of causing exactly n collisions by transmitting in the given slot s. The expected number of collisions in every slot can be calculated using a single matrix calculation as follows:

$$\mathbb{E}[\boldsymbol{n_c}] = \big(0, 1 \ldots N_{1-\text{hop}}\big)\boldsymbol{P_{nc}} \tag{3}$$

where $\mathbb{E}[\boldsymbol{n_c}]$ is a vector stating the expected number of collisions in every slot; $(0, 1, \ldots, N_{1-\text{hop}})$ is a horizontal vector of possible nubers of collisions (ranging from zero to $N_{1-\text{hop}}$); and $\boldsymbol{P_{nc}}$ is a matrix where every element $P_{nc}[n, s]$ is defined as the probability of causing exactly $n-1$ collisions in slot s. Furthermore, the probability of causing zero collisions in a given slot is obtained by extracting the first row of $\boldsymbol{P_{nc}}$

$$\forall s \in \{1 \ldots N_{\text{slots}}\},\ p_{\text{free}}[s] = P_{nc}[1, s]. \tag{4}$$

Therefore, to solve the optimization problem (1a), the new node needs to estimate $\boldsymbol{P_{nc}}$ based on limited information it can gather about the network topology and schedule. The proposed method to estimate $\boldsymbol{P_{nc}}$ is described as follows.

B. Estimating the Number of Collisions

In order to compute the elements of the $\boldsymbol{P_{nc}}$ matrix to enable the joining node to solve the optimization problem (1a), it needs to obtain some information about the network topology and schedule.

To incorporate the uncertainty caused by the likely incomplete knowledge available to the joining node, the slot usage probability matrix $\boldsymbol{P_{su}}$ is defined, where $P_{su}[i, s] \in [0, 1]$ is the probability of the ith node using slot s in the schedule. The dimensions of $\boldsymbol{P_{su}}$ are $N_{\text{nodes}} \times N_{\text{slots}}$, where N_{nodes} is the number of nodes detectable by the joining node. Assume that the joining node is able to detect at most its own immediate 1-hop neighbors and, by decoding the destination addresses of their packets, their 1-hop neighbors, hereafter referred to as the joining node's *2-hop neighborhood*. For example, in the scenario in Fig. 2 $N_{\text{nodes}} = 4$, i.e., the joining node can detect at most nodes 1, 2, 3, and 4. Note that the incumbent nodes are indexed by the joining node as $\{1 \cdot N_{\text{nodes}}\}$ and not by their global network addresses. In the illustrative example in Fig. 2, nodes N1, N2, N3, and N4 are detected by the joining node, therefore, their indices from the viewpoint of the joining node, $i \in \{1, 2, 3, 4\}$, coincide with their global addresses $\{N1, N2, N3, N4\}$. However, if the joining node had detected a different subset of the incumbent nodes, e.g., N3, N4, N5, and N6, then these nodes would still be indexed by the joining node as $i \in \{1, 2, 3, 4\}$, respectively.

To keep track of its local topology, i.e., which nodes are connected within its 2-hop neighborhood, a connectivity matrix $\boldsymbol{C}$ of dimensions $N_{\text{nodes}} \times N_{\text{nodes}}$ is maintained by the joining node. $\boldsymbol{C}$ is a binary matrix whose elements are defined as $C[i, j] = 1$, if there is a link between nodes i and j, and $C[i, j] = 0$, otherwise. For example, if the joining node correctly estimates its local topology in Fig. 2, the connectivity matrix is

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \tag{5}$$

where the rows and columns of $\boldsymbol{C}$ correspond to nodes 1, 2, 3, and 4, respectively, and the diagonal "self-connectivity" elements are defined as 1. Let $M_{1-\text{hop}}$ be the set of nodes within the connection range of the joining node, represented by their corresponding indices in $\boldsymbol{C}$. In the example in Fig. 2, the joining node is connected to nodes 1 and 2, therefore, $M_{1-\text{hop}} = \{1, 2\}$.

Given $\boldsymbol{P_{su}}$ and $\boldsymbol{C}$, the joining node can estimate $P_{\text{col}}[n, s]$—the probability of collision with a particular 1-hop neighbor n in slot s

$$\forall n \in M_{1-\text{hop}} \quad \forall s \in \{1 \ldots N_{\text{slots}}\}$$
$$P_{\text{col}}[n, s] = 1 - \prod_{i=1}^{N_{\text{nodes}}} (1 - C[i, n]\, P_{su}[i, s]) \tag{6}$$

i.e., one minus *the probability of no transmissions* in slot s, the latter calculated as the joint probability of node n, and all of its neighbors *not* transmitting in slot s.

Finally, having established $\boldsymbol{P_{col}}$, the joining node can calculate $\boldsymbol{P_{nc}}$. First, the probability of causing zero collisions (i.e., the first row of $\boldsymbol{P_{nc}}$) is calculated as the joint probability of *not colliding with any* 1-hop neighbors

$$\forall s \in \{1 \ldots N_{\text{slots}}\},\ P_{nc}[1, s] = \prod_{n=1}^{N_{1-\text{hop}}} (1 - P_{\text{col}}[n, s]). \tag{7}$$

Next, the probability of causing the maximum number of collisions $N_{\max}$, i.e., the last row of $\boldsymbol{P_{nc}}$, is calculated as the joint probability of *colliding with all* 1-hop neighbors

$$\forall s \in \{1 \ldots N_{\text{slots}}\},\ P_{nc}\big[N_{1-\text{hop}} + 1, s\big] = \prod_{n=1}^{N_{1-\text{hop}}} P_{\text{col}}[n, s]. \tag{8}$$

To calculate the intermediate rows of $\boldsymbol{P_{nc}}$, the probabilities of colliding with every subset of the joining node's 1-hop neighbor set $M_{1-\text{hop}}$ needs to be considered.

A set of all possible k-subsets of $M_{1-\text{hop}}$ is defined as follows:

$$[M]^k \coloneqq \big\{M \big| M \subset M_{1-\text{hop}},\ |M| = k\big\} \tag{9}$$

where $|\cdot|$ is the cardinality of a set.

The probability of collision with exactly k neighbors in slot s, i.e., $(k+1)$th row of $\boldsymbol{P_{nc}}$, can be calculated as the sum of probabilities of collision with every possible k-subset of $M_{1-\text{hop}}$

$$\forall k \in \{1 \ldots N_{1-\text{hop}} - 1\} \quad \forall s \in \{1 \ldots N_{\text{slots}}\}$$
$$P_{nc}[k+1, s] = \sum_{M \in [M]^k} \left(\prod_{n \in M} P_{\text{col}}[n, s] \prod_{n \in \overline{M}} (1 - P_{\text{col}}[n, s]) \right) \tag{10}$$

where $\overline{M} = \{M_{1-\text{hop}} - M\}$ is the set of nodes excluded from the subset M.

Note that the calculation of $\boldsymbol{P}_{\mathbf{nc}}$ matrix defined in (10) involves iterating over all k-subsets of $M_{1-\text{hop}}$, thus giving the proposed algorithm exponential time complexity. Since the set of all subsets, i.e., the powerset of $M_{1-\text{hop}}$ has $2^{N_{1-\text{hop}}}$ elements [31], the number of k-subsets of $M_{1-\text{hop}}$ for $k \in \{1 \dots N_{1-\text{hop}} - 1\}$ grows exponentially with the number of 1-hop neighbors of the joining node. Furthermore, the computation of $\boldsymbol{P}_{\mathbf{nc}}$ in (10) also involves nested iteration over all time slots in the frame and all nodes within a k-subset of $M_{1-\text{hop}}$. Therefore, given that both $N_{1-\text{hop}}$ and N_{slots} are bounded by the total number of nodes in the network, the worst case time complexity of the proposed optimization algorithm can be described as $\mathcal{O}(2^n n^2)$. However, for realistic scenarios with a limited number of 1-hop neighbors and a limited number of slots in a frame, this exponential time complexity has negligible effect on the computational load of the joining node. Furthermore, considering that in realistic sparsely connected *ad hoc* networks, the number of 1-hop neighbors of the joining node is likely to be limited (e.g., in the order of 5–10 nodes), regardless how big the rest of the network is, the time complexity reduces to linear time: $\mathcal{O}(2^{N_{1-\text{hop}}} \times n \times N_{1-\text{hop}}) \approx \mathcal{O}(n)$ (assuming that $N_{1-\text{hop}}$ is a fixed number), showing that the proposed algorithm can also be highly computationally efficient, depending on the number of 1-hop neighbors within communication range of the new node joining the network.

C. Estimating the Slot Usage Probabilities

The key prerequisite for the joining node to calculate $\boldsymbol{P}_{\mathbf{nc}}$, as described in the above section, is its ability to estimate the slot usage probability matrix $\boldsymbol{P}_{\mathbf{su}}$ given incomplete information. Assume that the joining node is only able to observe the activity of its immediate 1-hop neighbors. For example, in the scenario in Fig. 2, the joining node can only interact with nodes 1 and 2 and has no way of directly obtaining any slot usage information from nodes 3 and 4. Therefore, the slot usage probability matrix estimated by the joining node in Fig. 2 could have the following structure:

$$\boldsymbol{P}_{\mathbf{su}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p_{3,3} & p_{3,4} \\ 0 & 0 & p_{4,3} & p_{4,4} \end{bmatrix} \tag{11}$$

where it was able to detect that node 1 is using slot 1, node 2 is using slot 2, and given the topology constraints, nodes 3 and 4 cannot reuse slot 1 or 2 without collisions. However, this leaves $\boldsymbol{P}_{\mathbf{su}}$ with unknown elements caused by the partial observability of the network. The job of the joining node is to assign values to the unknown slot usage probabilities denoted by $p_{n,s}$.

The simplest approach is to assign uniform probabilities to the unknown elements of $\boldsymbol{P}_{\mathbf{su}}$, e.g., replacing the unknown elements in (11) with 0.5, meaning that the joining node has no additional information that could inform the unknown slot usage probability distribution

$$\forall n \in M_{2-\text{hop}} \quad \forall s \in S^n_{\text{avail}}, \; P_{su}[n, s] = \frac{1}{|S^n_{\text{avail}}|} \tag{12}$$

where $M_{2-\text{hop}} = \{\{1 \dots N_{\text{nodes}}\} - M_{1-\text{hop}}\}$ is the set of indices of the joining node's 2nd hop neighbors, e.g., $M_{2-\text{hop}} = \{3, 4\}$ in the scenario in Fig. 2; S^n_{avail} is the set of slots available to node n, i.e., the slots with unknown usage probabilities in the nth row of $\boldsymbol{P}_{\mathbf{su}}$.

D. Biased Slot Usage Probabilities

To leverage the joining node's knowledge of *how* TDMA schedules are generated, a method of introducing a bias into the slot usage probabilities is proposed here. A copy of a deterministic computer function that generates TDMA schedules is stored at every incumbent node and at any new node joining the network, such that all nodes can independently generate new schedules if needed, as explained in Section III. The input to this function is a complete network topology represented by the connectivity matrix $\boldsymbol{C}_{\mathbf{full}}$ as an input; and the output is a conflict-free slot assignment for every node that minimizes the number of slots per frame.

The nodes corresponding to the rows and columns of $\boldsymbol{C}_{\mathbf{full}}$ are ordered by their numerical addresses from lowest to highest. For example, in the network from Fig. 2, $\boldsymbol{C}_{\mathbf{full}}$ has six rows and six columns representing nodes 1, 2, 3, 4, 5, and 6, respectively. As a result, the slot assignment tends to be correlated with the node addresses. Fig. 2 gives an example of this behavior where nodes 1, 2, 3, and 4 are assigned slots 1, 2, 3, and 4, respectively, because the deterministic slot assignment function iterates over the nodes in the ascending order of their addresses.

The new node joining the network can use its knowledge of this deterministic slot assignment procedure to skew the slot usage probabilities, predicting the likely correlation between the node addresses and their assigned slots. For example, while deriving the unknown values of slot usage probabilities in (11), the joining node could predict that node 3 is more likely to use slot 3 and node 4 uses slot 4, than the other way around.

To predict the slot usage of its 2-hop neighbors, first, the joining node determines which slot is most likely used by each of them based on their relative numerical addresses

$$\forall n \in M_{2-\text{hop}}, \; \hat{s}_n = \max\left\{1, \left\lfloor \left|S^n_{\text{avail}}\right| \frac{n}{N_{\text{nodes}}} \right\rfloor\right\} \tag{13}$$

where $\hat{s}_n$ is the index of a slot in S^n_{avail} most likely used by node n, and $\lfloor \cdot \rfloor$ rounds down the number to the nearest integer. Next, the nonuniform slot usage probability distribution is chosen such that $\hat{s}_n$ is the *mode*, and the other slots are assigned decreasing probabilities of usage, the further away they are from $\hat{s}_n$. The normal probability distribution is adopted, although other unimodal distributions could also be used, e.g., triangular. The relative likelihood of node n using a particular slot is calculated as follows:

$$\forall s \in S^n_{\text{avail}}, \; p^*_n[i_s] = f_n(i_s|\mu_n, \sigma_n) \tag{14}$$

where i_s is the index of slot s in S^n_{avail}; $p^*_n[i_s]$ is the unnormalized probability of node n using slot s; $f_n(x|\mu, \sigma)$ is the

normal probability density function with the mean μ and standard deviation σ evaluated at x; $\mu_n = \hat{s}_n$ is the mean (and mode) of the distribution; and σ_n is the standard deviation calculated for the nth row of $\boldsymbol{P}_{\mathbf{su}}$ as follows:

$$\sigma_n = \frac{1}{2}\max\left\{1, \hat{s}_n - 1, \left|S^n_{\text{avail}}\right| - \hat{s}_n.\right\}. \tag{15}$$

This method was empirically validated to provide an appropriate spread of probability values between the most likely and the least likely slot used by a particular node.

Finally, the discrete probability distribution values are normalized and saved in the overall slot usage probability matrix

$$\forall n \in M_{2-\text{hop}} \quad \forall s \in S^n_{\text{avail}}$$
$$P_{su}[n, s] = \frac{p^*_n[i_s]}{\sum_j p^*_n[j]}. \tag{16}$$

E. Local Topology Uncertainty

The method of estimating the number of collisions proposed in Section IV-B assumes knowledge of the joining node's 2-hop neighborhood connectivity matrix $\boldsymbol{C}$ in (6). However, as established in the system model in Section III, the network will only be partially observable to the joining node via its direct 1-hop neighbors. Therefore, in many cases, the connectivity matrix estimated by the joining node will contain a mixture of known and unknown elements, similar to the uncertainty in the slot usage probabilities in (11). For example, by observing its immediate 1-hop neighbors in the scenario from Fig. 2, the joining node could estimate the following connectivity matrix:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & c_{3,4} \\ 1 & 0 & c_{4,3} & 1 \end{bmatrix} \tag{17}$$

where the link between nodes 3 and 4 is unknown.

To incorporate this topology uncertainty into the probabilistic analysis proposed in Section IV-B, we define $\boldsymbol{\mathcal{T}}$—the set of possible network topologies that assigns all combinations of binary values to the unknown links

$$\boldsymbol{\mathcal{T}} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_N\} \tag{18}$$

where N is the total number of possible network topologies which is a function of N_{ul}—the number of unknown links in the joining node's 2-hop neighborhood

$$N = 2^{N_{ul}}. \tag{19}$$

For example, the topology uncertainty in (17) only includes one unknown link; therefore, it would be covered by a set of two possible topologies $\boldsymbol{\mathcal{T}} = \{\boldsymbol{C}_1, \boldsymbol{C}_2\}$, one topology where nodes 3 and 4 connected and the other where there is no link between them.

In this way, the expected number of collisions can be estimated by averaging $\boldsymbol{P}_{\mathbf{nc}}$ over all possible network topologies

$$\forall k, s, P_{nc}[k, s] = \frac{1}{N}\sum_{\boldsymbol{C}\in\boldsymbol{\mathcal{T}}} P^{\boldsymbol{C}}_{nc}[k, s] \tag{20}$$

where $P^{\boldsymbol{C}}_{nc}[k, s]$ is the probability of causing $k-1$ collisions in slot s, given the network topology $\boldsymbol{C}$.

The complexity of this approach grows exponentially with the number of unknown links, as shown in (19), thus introducing another exponential term into the time complexity of the proposed optimization algorithm. Since the number of unknown links $N_{ul} = [(N_{2-\text{hop}}(N_{2-\text{hop}} - 1))/2]$ is proportional to $N^2_{2-\text{hop}}$, and the number of 2-hop neighbors $N_{2-\text{hop}}$ is bounded by the total number of nodes, the overall worst case complexity of the algorithm, taking into account the topology uncertainty, is $\mathcal{O}(2^{n^2}2^n n^2) = \mathcal{O}(2^{n^2+n}n^2)$, or $\mathcal{O}(2^{n^2}n)$ if the number of 1-hop neighbors is limited (as described in Section IV-B). This is significantly less computationally feasible for large networks. Therefore, to ensure low computation requirements at the joining node, an approximation is required, where a random subset of possible topologies $\boldsymbol{\mathcal{T}'} \subset \boldsymbol{\mathcal{T}}$ is used to estimate the number of collisions in (20), in cases where the number of unknown links is too large to iterate over all possible topologies in $\boldsymbol{\mathcal{T}}$. For example, in this article, the number of analyzed topologies is limited to $|\boldsymbol{\mathcal{T}'}| = 2^{12}$, without any visible effect on the joining node's performance. This reduces the time complexity of the algorithm for large network sizes back to $\mathcal{O}(2^{12}2^n n^2) \approx \mathcal{O}(2^n n^2)$, or $\mathcal{O}(2^{12}n) \approx \mathcal{O}(n)$ if the number of 1-hop neighbors is limited, and was empirically found to keep the computation time low for network sizes up to 100 nodes.

F. Initial Listening Phase

In order to produce initial estimates of $\boldsymbol{C}$ and $\boldsymbol{P}_{\mathbf{su}}$ when the new node joins the network, it starts with a promiscuous listening phase, observing other nodes' packets, and recording the following information.

1) $\boldsymbol{a}_{\mathbf{s}}$: Packet source addresses.
2) $\boldsymbol{a}_{\mathbf{d}}$: Packet destination addresses.
3) $\boldsymbol{s}_{\mathbf{rx}}$: Slot counter values at the time of reception, i.e., how many slots have passed between the start of recording and receiving each packet.

First, the joining node detects the set of its 1-hop neighbor addresses $A_{1-\text{hop}}$ by parsing the unique packet source addresses

$$A_{1-\text{hop}} = A_s, A_s = \{a_s[i]\}_{i\in\{1\ldots N_p\}} \tag{21}$$

where A_s is the set of unique source addresses in $\boldsymbol{a}_{\mathbf{s}}$ sorted from lowest to highest, $a_s[i]$ is the source address of the ith recorded packet, and N_p is the total number of recorded packets.

Next, it creates the set of addresses of all detected nodes, parsing the unique source and destination addresses from all recorded packets

$$A_{\text{full}} = A_s \cup A_d, A_d = \{a_d[i]\}_{i\in\{1\ldots N_p\}} \tag{22}$$

where A_{full} is the full set of detected node addresses sorted from lowest to highest, and A_d is the set of detected destination addresses.

Based on this information, the joining node initializes the topology-related variables required for the probabilistic slot selection approach proposed in this section.

1) N_{nodes}: Number of elements in A_{full}.
2) N_{1-hop}: Number of elements in $A_{1-\text{hop}}$.

3) M_{1-hop}: Set of indices stating which of the node addresses in A_{full} is its 1-hop neighbors.

It also creates the connectivity matrix $\boldsymbol{C}$ as follows.

1) Initialize all elements to 0.
2) Set all diagonal "self-connectivity" elements as 1.
3) For every recorded packet $i \in \{1 \dots N_p\}$, mark the links between the source $a_s[i]$ and destination $a_d[i]$ address and *vice versa* with 1.
4) Mark the links between every pair of detected nodes whose address is not in $A_{1-\text{hop}}$ (i.e., the 2nd hop neighbors) as *unknown*.

For example, following the above steps, the connectivity matrix initialized by the joining node in the scenario Fig. 2 is given in (17).

In addition to estimating the local topology from the recorded packets, the joining node can also extract valuable schedule information from them. First, it can establish the current frame length (number of slots per frame) by detecting the *minimum interval* between any two packets received from the same source address

$$N_{\text{slots}} = \min\Big\{ s_{rx}[i] - s_{rx}[j] \Big| i, j \in \{1 \dots N_p\} \quad i > j, a_s[i] = a_s[j] \Big\} \tag{23}$$

where $s_{rx}[i]$ is the value of the slot counter when the ith packet was received.

Having established the current frame length, the joining node can detect the slot usage pattern of its 1-hop neighbors

$$\forall i \in \{1 \dots N_p\}, s_{tx}[n_i] = ((s_{rx}[i] - 1) \bmod \mathrm{N_{slots}}) + 1 \tag{24}$$

where n_i is the index of the source address of packet i in A_{full}, equal to the source node's index in the $\boldsymbol{P_{su}}$ matrix, and $s_{tx}[n_i] \in \{1..N_{\text{slots}}\}$ is the slot in which this packet arrived, i.e., the slot used by node n_i. This knowledge can be used to initialize $\boldsymbol{P_{su}}$ as follows:

$$\forall n \in M_{1-\text{hop}} \quad \forall s \in \{1 \dots N_{\text{slots}}\}$$
$$P_{su}[n, s] = \begin{cases} 1, & s_{tx}[n] = s \\ 0, & s_{tx}[n] \neq s \end{cases} \tag{25}$$

i.e., using the direct observations from the listening phase to set the slot usage probabilities to 1 or 0 depending on whether the joining node received any transmissions from its 1-hop neighbors in a given slot.

Furthermore, the joining node can eliminate the possibility of other nodes reusing the slots of its 1-hop neighbors, if they are within 2-hop connectivity of each other

$$\forall n \in M_{2-\text{hop}} \quad \forall m \in \left\{ j \in M_{1-\text{hop}} \,\middle|\, C_{2-\text{hop}}[n, j] = 1 \right\}$$
$$P_{su}[n, s_{tx}[m]] = 0 \tag{26}$$

where $C_{2-\text{hop}}[n, j]$ is a binary flag indicating whether the hop distance between nodes n and j is ≤ 2, e.g., calculated using Dijsktra's algorithm [32].

The remaining slot usage probabilities are unknown and are estimated as explained in Section IV-D. For example, the slot usage probability matrix derived by the joining node after recording the packets transmitted by nodes 1 and 2 in Fig. 2 is given in (11).

G. Updates Based on Collision Notifications

After the joining node completes the initial listening phase and creates the local topology and schedule estimates as described in Section IV-F, it selects the time slot for its initial transmission by solving the optimization problem defined in (1a). It then waits until the chosen slot and broadcasts a predefined "new node" message that contains its *detected 1-hop neighbor list*. For example, in the scenario in Fig. 2, the joining node would broadcast a message stating that its 1-hop neighbors are nodes 1 and 2.

Afterward, the joining node waits for the next packet transmitted by each of its 1-hop neighbors, which is one of the following.

1) *Collision Notification:* The neighbor node detected a collision (or a scheduling clash) with another node, and broadcasted "Collision with Node X," where X is the address of the node scheduled to transmit in that slot.
2) *No Response:* The neighbor node continues its normal operation if the packet was successfully received and no collision or scheduling conflict was detected.

The information gained from observing the reaction of its neighbors to the joining node's transmission is incorporated into its local topology and schedule estimates, reducing the uncertainty in estimating the expected number of collisions for the next transmission. The new information is incorporated into $\boldsymbol{C}$ and $\boldsymbol{P_{su}}$ as follows.

Let M_{col} be the set of nodes with which a collision was detected based on the notifications sent by its 1-hop neighbors. The joining node can then update the slot usage probabilities for the nodes that collided with it (if any) as follows:

$$\forall m \in M_{\text{col}} \quad \forall s \in \{1 \dots N_{\text{slots}}\}, \; P_{su}[m, s] = \begin{cases} 1, & s = \hat{s} \\ 0, & s \neq \hat{s} \end{cases} \tag{27}$$

and for the remaining 2nd hop neighbors that did not clash with the joining node

$$\forall m \in \left\{ M_{2-\text{hop}} - M_{\text{col}} \right\}, \; P_{su}[m, \hat{s}] = 0 \tag{28}$$

where $\hat{s}$ is the slot used by the joining node to transmit its last packet.

These updates reduce the number of uncertain elements in $\boldsymbol{P_{su}}$, i.e., where $0 < P_{su} < 1$, which can be further reduced by eliminating the possibility of slot reuse between any 2-hop neighbors, similar to (26)

$$\forall n \in M_{2-\text{hop}} \quad \forall m \in \left\{ j \in M_{\text{certain}} \,\middle|\, C_{2-\text{hop}}[n, j] = 1 \right\}$$
$$P_{su}[n, s_{tx}[m]] = 0 \tag{29}$$

where

$$M_{\text{certain}} = \left\{ n \in \{1 \dots N_{\text{nodes}}\} \;\middle|\; \exists s, \; P_{su}[n, s] = 1 \right\} \tag{30}$$

i.e., M_{certain} is the set of nodes whose scheduled slot is known for certain. The remaining uncertain elements of $\boldsymbol{P_{su}}$ are assigned new values using the method described in Section IV-D.

In some cases, when a collision with more than one 2nd hop neighbor was detected, the joining node can also reduce uncertainty in its local topology estimate by eliminating the

possibility of a link between those nodes, since they would not be able to use the same slot without collisions if they were connected

$$\forall m \in M_{\text{col}} \quad \forall n \in \left\{ j \in M_{\text{col}} \,\middle|\, j \neq m \right\},\ C[m,n] = 0. \quad (31)$$

After updating $\boldsymbol{P}_{\mathbf{su}}$ and $\boldsymbol{C}$ as described above, the joining node repeats the slot selection procedure by solving the optimization problem in (1a) using the refined local information, if it has not found a collision free slot yet, i.e., if $M_{\text{col}} \neq \emptyset$.

Otherwise, if a transmission in slot $\hat{s}$ has not caused any collisions or scheduling conflicts, updating the expected number of collisions as described in Section IV-B will yield

$$P_{nc}[1,\hat{s}] = 1,\ \mathbb{E}\left[n_c[\hat{s}]\right] = 0 \quad (32)$$

which means that $\hat{s}$ is a collision-free slot and that the slot selection procedure at the joining node has converged.

H. Network Adaptation to New Node

When an incumbent node receives the "new node" packet containing its detected 1-hop neighbor list, it reconstructs the local topology of the joining node consisting of the $M_{1-\text{hop}}$ and $M_{2-\text{hop}}$ node sets. It then checks if it is possible for the joining node to find a collision-free slot or if there are no slots available to reuse (due to topology constraints). A collision-free solution is possible if the following condition is met:

$$\exists s \quad \forall n \in \left\{ M_{1-\text{hop}} \cup M_{2-\text{hop}} \right\},\ s^*_{tx}[n] \neq s \quad (33)$$

where $s^*_{tx}[n]$ is the slot allocated to node n according to the TDMA schedule (ground truth). If such a slot exists, the incumbent node proceeds with the normal operation, responding to the joining node's packet as described in the previous section.

If the incumbent node detects a collision and responds with the collision notification broadcast, the use of the *quiet frame feature* is proposed, whereby the incumbent nodes that received this collision notification *back off* for one frame. This ensures that the "new node" packet will be received in the next frame, thus reducing the probability of collision and speeding up the node integration process.

However, if there is no collision-free slot available according to (33), the network will have to adapt its schedule to accommodate the new node. In this case, the incumbent node waits until its scheduled slot and broadcasts a "network update" packet containing the address of the joining node and its 1-hop neighbor list, thus triggering a network-wide update. The other incumbent nodes receiving this packet, rebroadcast it to their neighbors until all nodes in the network have received the update.

Once the update is received, every incumbent node has the complete topology information, including the 1-hop neighbor list of the joining node. Therefore, it can track which nodes have already received the topology update and which nodes are yet to receive it. Furthermore, since every incumbent node knows the network schedule, it can accurately estimate when the update will propagate across the entire network, i.e., after how many slots it will reach the last incumbent node. Afterward, having access to an identical copy of the deterministic computer function that produces a packet schedule, every incumbent node generates it locally and switches to the new schedule after the calculated number of slots.

When the joining node receives the "network update" packet from one of its neighbors, it learns that there is no solution to the slot selection problem, stops its opportunistic transmissions, and waits for a packet from one of its neighbor nodes containing the full network topology and the slot in which the new schedule starts, thus completing the integration of the new node into the network.

I. Protocol Details

Fig. 3(a) summarizes the JOIN protocol proposed in this section. The key input/output variables and the subsections explaining the corresponding steps of the slot search procedure are annotated in the flowchart. Fig. 3(a) describes the key protocol steps at the incumbent nodes, including the collision feedback, quiet frame feature, and the network adaptation process proposed in Sections IV-G and IV-H.

J. Discussion

It was shown in Section IV-B that the time complexity of the optimization algorithm performed by the joining node varies between linear time and exponential time, depending on the local topology of the joining node. If the number of 1-hop neighbors of the joining node grew indefinitely with the network size, then the time complexity of the optimization algorithm would indeed be exponential, and further research on the JOIN protocol is required to reduce this time complexity to apply it to large densely connected UANs (e.g., hundreds of nodes covering a small geographical area). However, if sparsely connected *ad hoc* networks are considered, such as that shown in Fig. 2, where the size of *the joining node's local topology* remains relatively small (e.g., in the order of 5–10 direct connections), then the exponential term in the worst case time complexity of the algorithm is reduced to a constant, because the number of 1-hop neighbors remains limited to a realistic number, even for large network sizes. The simulations reported in this article have shown that the proposed protocol is computationally feasible for networks up to 100 nodes.

Another important consideration of the JOIN protocol is that the joining node builds its knowledge based on data gathering from received packets, thus making JOIN sensitive to packet loss due to channel distortion or strong ambient noise. In such cases, the loss of packets at the joining node will generate false decisions leading to an increased delay until convergence. On the other hand, loss of packets at the incumbent nodes would yield an increase in the delay until the network adapts to the change. However, since the protocol followed by the incumbent nodes is assumed to be collision free, it is possible for the loss of packets due to collisions to actually speed up the convergence of the joining node by extending its local topology knowledge based on the extra collision notifications it receives. The effects of packet loss on the performance of the JOIN protocol in real underwater acoustic environments are

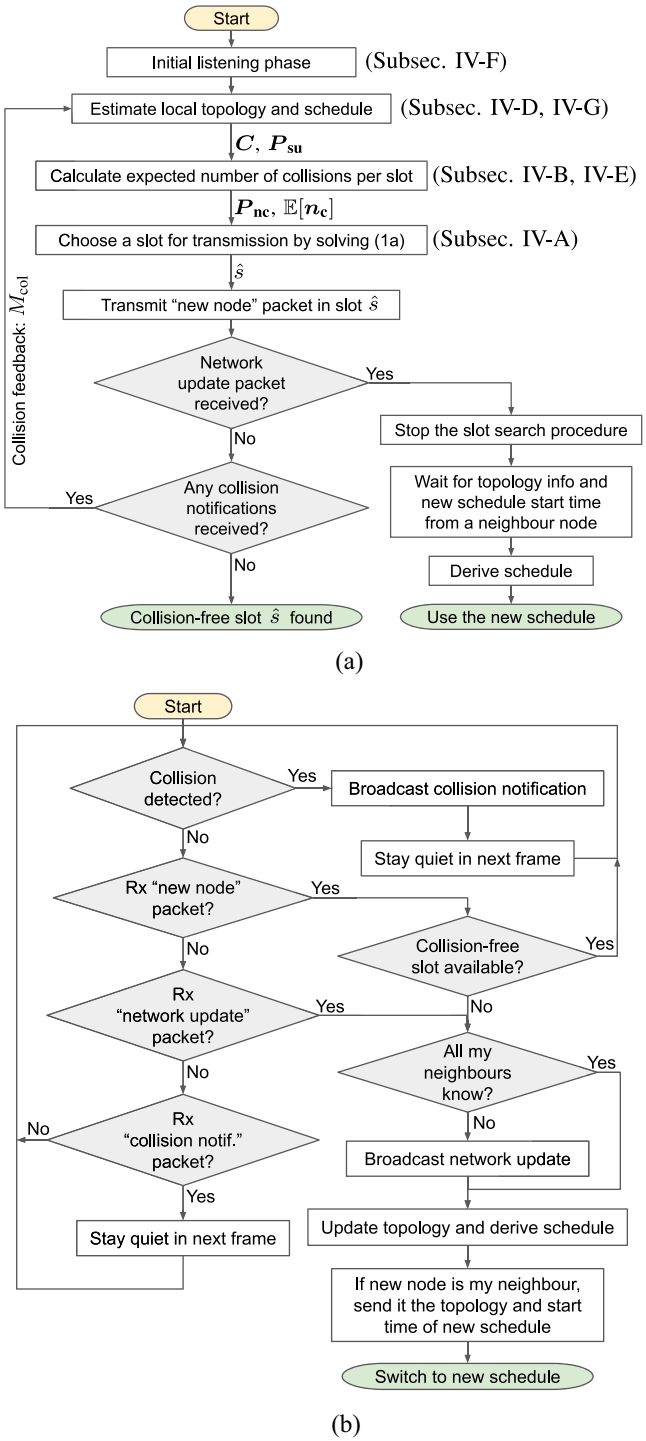


Fig. 3. JOIN protocol. (a) New node trying to join an active network. (b) Incumbent network nodes.

analyzed in Section VI where we test JOIN in two separate sea trials. Another source of load on the network is the overhead caused by the distribution of the topology update across the network, which is necessary in cases where the schedule adaptation is required. The information content of this update in most cases is small (a list of 1-hop neighbor addresses) and can be "piggybacked" onto the regular packets of the network nodes, thus significantly reducing the effect of this overhead on the system performance and making it comparable to any other form of topology change.

V. Simulation Results

In this section, the JOIN protocol is evaluated in a series of Monte Carlo simulations implemented in MATLAB and its performance is quantified using the metrics described in Section III: time until convergence of the joining node, network adaptation speed, and the number of collisions caused by the joining node.

A. Simulation Setup

Networks comprising between 10 and 100 nodes are considered, with 50 000 randomly generated topologies simulated for every network size, thus providing sufficient data to draw statistically valid conclusions about the network performance. To generate a random network topology, the nodes with a fixed 1-km connection range were randomly distributed within a square coverage area with the average node density $D = 2$ nodes per km^2. Without loss of generality, this approach was empirically tested to produce realistic, sparse network connectivity patterns with opportunities for spatial TDMA slot reuse, such as that shown in Fig. 2, regardless of the overall network size. The average node density D was achieved by setting the square coverage area to $A = N_{\text{total}}/D$ km^2, where N_{total} is the total number of nodes in the network. All simulated topologies were fully connected (no isolated nodes).

Since one of the key motivations of this article is to maintain high throughput capacity in UANs, JOIN is evaluated under the full buffer traffic load—the worst case scenario for the new node joining the network. The performance of JOIN is compared with "opportunistic random access," where the joining node gathers the same initial information about the slot usage as described in Section IV-F, i.e., the frame length and the slots used by its immediate neighbors, and chooses randomly among the unoccupied slots in the frame, similar to the random access cognitive radio approach in terrestrial networks [33]. This is a good benchmark for comparison because it isolates and quantifies the benefits of the intelligent probabilistic slot selection approach, compared with treating the uncertainty in the network as uniform random. Furthermore, the state-of-the-art solutions for node integration into existing networks, reviewed in Section II, do not meet the control signaling constraints of the assumed system model in Section III, and, therefore, could not be used for direct benchmark comparison.

B. Joining Node Convergence Time

Fig. 4 shows the plots of the joining node's convergence time, measured as the number of slots between the joining node's initial opportunistic transmission until it found a collision-free slot. Out of the 50 000 simulations for every network size, the scenarios without a collision-free slot available were filtered out, and the results of the remaining simulations were used for the plots, where every data point represents the mean, and the error bars show the 10th and 90th percentiles.

Fig. 4(a) demonstrates the scalability of JOIN, since the average time it takes the joining node to converge on a solution is relatively insensitive to the network size. This is

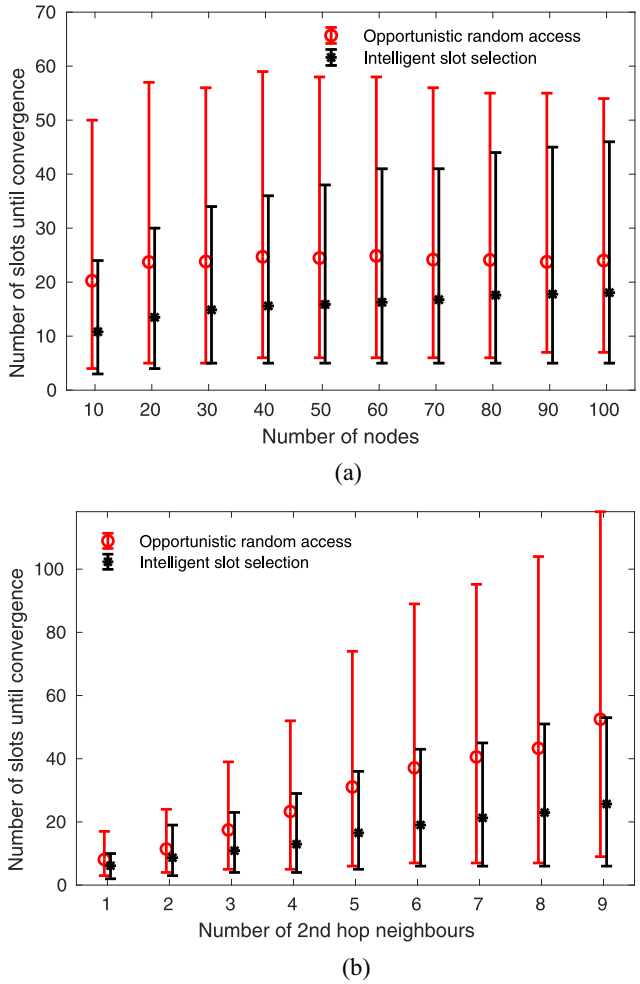


Fig. 4. Joining node's convergence speed. (a) Scalability with the network size. (b) Scalability with the number of 2nd hop neighbors in a 20 node network.

because any nodes outside of the joining node's 2-hop neighborhood have no effect on this process, thus reducing the *effective network size* to the joining node's local topology. The slight increase in the convergence time with the network size is due to the number of slots per frame being statistically higher for larger networks, resulting in longer gaps between the joining node's attempts to find a slot. The figure also shows that the intelligent slot selection approach in JOIN achieves significantly faster convergence, compared with random opportunistic access, aided by the probabilistic analysis of all available local topology and slot usage information.

Fig. 4(b) shows that the key parameter affecting the speed of convergence of the joining node to a solution is the number of 2nd hop neighbors. This is because the 2nd hop neighbors are within the joining node's local network topology, yet they cannot be directly observed by it. Therefore, they introduce uncertainty into the joining node's probabilistic analysis, increasing the complexity of the slot selection problem as described in Section IV-E. Nevertheless, Fig. 4(b) shows that JOIN scales significantly better than random opportunistic access, with an increased benefit gained from the proposed intelligent slot selection method as the number of 2nd hop neighbors increases.

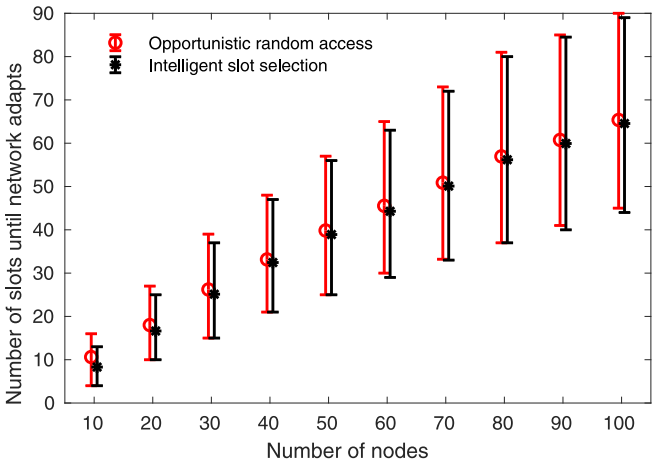


Fig. 5. Speed of network adaptation to a new node arrival.

C. Network Adaptation Speed

Fig. 5 shows how quickly the network distributes the topology update to all nodes in the scenarios without a collision-free solution, where the job of the joining node is to trigger a network-wide topology update such that the nodes can derive a new schedule and switch to it as soon as possible. This network adaptation speed is quantified as the number of slots from the joining node's initial transmission until the last node in the network receives an update about it. The plot shows that the proposed protocol achieves a fast network-wide distribution of the topology update, especially for smaller network sizes, increasing for larger networks due to the increase in the maximum number of hops, and, therefore, in the number of frames it takes to distribute the topology update to every node in the network. Nevertheless, the network adaptation process takes a comparable number of slots to that required by the joining node to iteratively learn a collision-free solution [see Fig. 4(a)]. In these scenarios, the benefit of intelligent slot selection is less pronounced, especially for larger network sizes, because the job of the joining node is reduced to delivering a single "new node" packet to at least one of its neighbors, with the rest of the network subsequently propagating this information to all other nodes.

D. Interference Caused by the Joining Node

Fig. 6 shows the number of collisions caused by the joining node in all simulations (both with and without a collision-free solution possible). First, the JOIN protocol consistently reduces the average number of collisions by approximately 50% compared with the opportunistic random access approach, which directly shows the effect of the proposed intelligent slot selection method based on minimizing the expected number of collisions, as formulated in Section IV-A. Second, the number of collisions caused by the joining node is on average between 1 and 2, and in the vast majority of cases less than 4. It is claimed that it constitutes a negligible level of network disruption, considering the signaling overhead savings provided by JOIN. Finally, Fig. 6 once again demonstrates the scalability of JOIN, since the number of collisions does not grow with

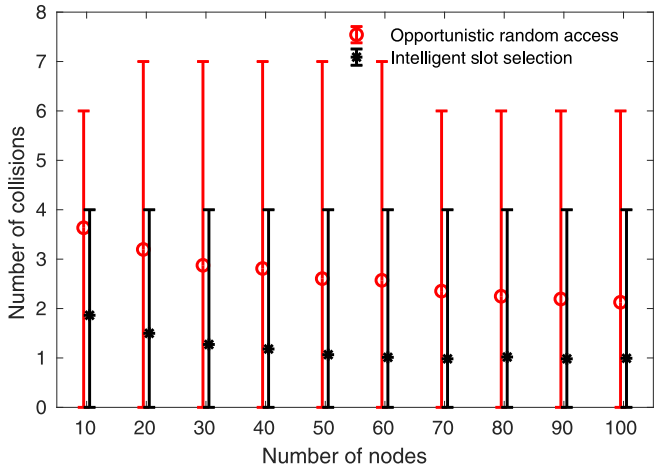


Fig. 6. Number of collisions caused by the joining node.

the number of nodes. In fact, in these simulations, the number of collisions slightly decreased with the network size. This is due to an increased likelihood of longer TDMA frames with potential collision-free slots in larger networks, thus making these scenarios slightly more favorable compared with smaller networks with fewer slots to choose from for the joining node.

VI. Sea Trials

In addition to evaluating JOIN in simulation, the protocol was implemented in hardware and tested in two separate sea trials with different network topologies and spatial reuse schedules. This demonstrated the successful operation of JOIN in a real UAN, thus validating its key assumptions, such as the ability of the joining node to detect its local 2-hop topology and establish the TDMA frame structure from listening and recording the packets sent by its neighbors, and the convergence of the iterative slot selection algorithm based on collision notifications.

A. Experimental Setup

Fig. 7 shows the pictures of the hardware used in the sea trials. Eight underwater acoustic nodes were built in total as follows.

1) *3 Tube Nodes:* Raspberry Pi 3B+ with a 5-V power supply inside a 3" BlueRobotics watertight enclosure, connected to an acoustic modem via a potted cable penetrator [Fig. 7(a)].
2) *3 Buoy Nodes:* Raspberry Pi 3B+ with a 5-V power supply inside a custom-built buoy connected to an acoustic modem via an 8-way Birns Aquamate connector [Fig. 7(b)].
3) *2 Laptop Nodes:* A laptop directly connected to an acoustic modem with a 10-m cable [Fig. 7(c)].

All nodes were equipped with the NMv3 acoustic modems built at Newcastle University, U.K. The NMv3 modems are the next generation of the modems used in [34], with 640-b/s raw data rate, 168-dB re μPa^{2}m^2 source level, 24–32-kHz frequency range, and 2–64-B data payloads. They were controlled using the serial interface via RS232-USB converters, and powered either from 5 V and GND pins on the Raspberry

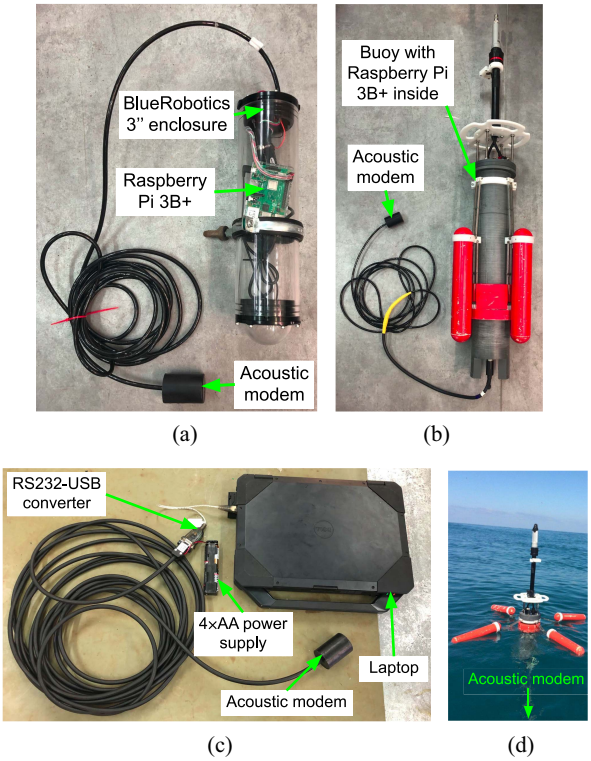
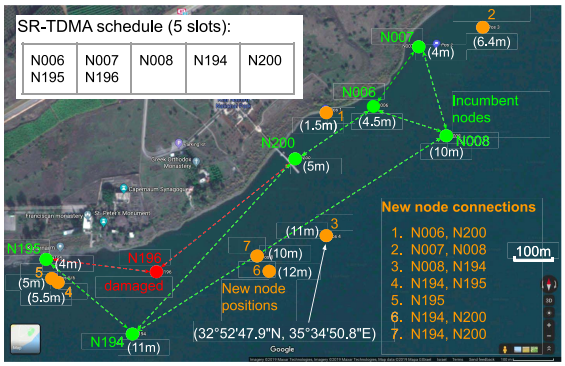


Fig. 7. Hardware used in the sea trials. All nodes were equipped with NMv3 acoustic modems developed at Newcastle University [34]. (a) Static 3" tube nodes. (b) Static buoy nodes. (c) Joining node (modem connected to laptop on boat). (d) Deployed buoy.

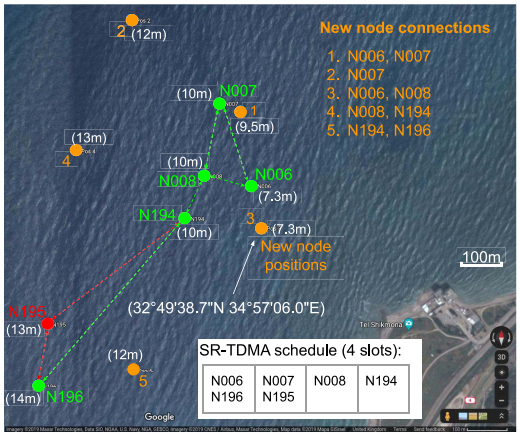
Pi or using a separate 6-V supply (4 × AA battery pack) as shown in Fig. 7(c).

JOIN was tested in 12 different scenarios over two sea trials shown in Fig. 8. In the Sea of Galilee trial on September 15, 2019, a network of seven nodes in total was deployed (N196 got damaged and was not operational), with one of the laptop nodes deployed from a pier as a static incumbent node (N200), and the other laptop, used as the joining node, deployed from the boat in seven different locations shown in Fig. 8(a). In the Mediterranean Sea trial on September 24, 2019, a network of six nodes was used (N195 malfunctioned), without the static laptop node, since the shallow depth, large rocks, and breaking waves near the shore did not allow acoustic communication with any other node. The other laptop node was deployed using the boat in five different locations shown in Fig. 8(b) as the joining node. In both trials, the sea state was calm. Sound speed measurements gathered in the Sea of Galilee showed an approximately constant water temperature of 30 °C and a sound speed of 1509 m/s (it is a sweet water lake). The sound speed in the Mediterranean Sea trial was approximately 1530 m/s with the water temperature of 27 °C.

The packet schedules and network topologies shown in Fig. 8 were uploaded to all incumbent nodes before deployment. In every frame, with a probability of 90% (nearly full-buffer traffic load), every node generated a packet addressed to one of its neighbors chosen at random and transmitted it in its assigned slot. The format of the packet payload was the following: "`DaaaFxxSy,`" where "`aaa`" is the destination address of the packet, "`xx`" is the frame counter value,



(a)



(b)

Fig. 8. Network topologies, joining node positions, and SR-TDMA schedules from the sea trials. The numbers in brackets show the sea depth. (a) Sea of Galilee, Israel, September 15, 2019. (b) Mediterranean Sea, Haifa, Israel, September 24, 2019.

and "y" is the slot index. The frame and slot index in the packet payload were not used in the protocol, but were used only to include some content in the packets. The structure of the other types of packets transmitted by the nodes in the sea experiments was the following.

1) "NEW_aaa_bbb_..."—The "new node" packet, where "aaa_bbb_..." is the list of 1-hop neighbor addresses detected by the joining node.
2) "COL_aaa"—The collision notification packet, where "aaa" is the address of the incumbent node with which a collision occurred.
3) "TOP_nnn_aaa_bbb_..."—The "topology update" packet, where "nnn" is the joining node's address and "aaa_bbb_..." is the list of 1-hop neighbor addresses detected by the joining node.

The nodes were synchronized using the NTP protocol before deployment and subsequently used their local system time during the experiments to keep track of slots and frames. Implementing a synchronization protocol using acoustic communications is a challenging task that was beyond the scope of these trials. Therefore, the slot duration was extended to 6 s to take into account loose synchronization with a potentially large clock drift of the Raspberry Pi-based nodes. However, since a time slot was used as the basic unit of time in this article, the results from these trials are generally applicable to any TDMA frame/slot specification, e.g., if tight synchronization is provided to the nodes, the slot duration could be much shorter, while the results from the experiments would be the same.

When the joining node was deployed in each position shown in Fig. 8, it started with the listening phase that lasted 30 or 20 slots, in the Sea of Galilee and Mediterranean Sea trials, respectively. As a result, in every experiment, the joining node successfully recorded enough packets to establish the frame length and the slot usage of its 1-hop neighbors [listed in Fig. 8(a) and (b)], and to detect all of its 2nd hop neighbors by reading the destination addresses of the packets. Afterward, it executed the slot selection algorithm proposed in this article, in some cases converging on a collision-free slot and in others—triggering a network-wide topology update. In the latter case, the incumbent nodes distributed this update across the entire network as described in Section IV-I. All experiments lasted between 2.5 and 4 min, with 2 to 3 min for the initial listening phase and approx. 30 s to 1.5 min to complete the new node integration afterward. The outcomes of the sea experiments are described in more detail as follows.

B. Results

Table I shows the results of the 12 sea experiments described in the previous subsection—7 in the Sea of Galilee [Fig. 8(a)] and 5 in the Mediterranean Sea [Fig. 8(b)]. The "outcome" column describes the final outcome of the experiment, i.e., either the joining node found a particular collision-free slot or it triggered a network-wide topology update, with the green tick and red cross symbols indicating whether these outcomes matched the theoretical expectation. The "slots" column states how many slots it took the joining node to converge on a solution or, in scenarios with no solution, how many slots it took for the topology update to propagate across the entire network. The next column shows the number of collisions caused by the joining node in each experiment. The last two columns show the simulation results of the baseline random opportunistic access scheme applied to every sea experiment scenario. These simulation results are shown in the format "mean [10th–90th percentile]" from 1000 simulations with different random seeds.

In 10 out of 12 experiments, the outcome matched the theoretical expectation and in most cases, the number of slots it took to complete the new node integration was smaller than the average baseline random access scheme, which is consistent with the simulation results in Section V. The number of collisions caused by the joining node was between 0 and 2, validating the consistently low level of network disruption observed in the simulations, and in most cases outperforming the simulated random access protocol. The two experiments that failed to produce the expected final outcome were the SG(6) and MS(5) experiments, although the number of collisions and the speed of convergence at the joining node were still consistent with the theoretical expectations and the simulation results. The expected outcome for both of them was the network-wide topology update. However, in the SG(6) experiment, having transmitted its packet in slot 2, the joining node failed to receive the "topology update" packets broadcasted by N200, and later by N194, due to an unreliable acoustic communication channel for those links, and therefore

TABLE I
OUTCOMES OF THE SEA EXPERIMENTS. SG—SEA OF GALILEE; MS—MEDITERRANEAN SEA. THE RESULTS ARE COMPARED WITH SIMULATIONS OF THE BASELINE RANDOM ACCESS SCHEME (MEAN AND 10TH–90TH PERCENTILE RANGE)

Exp.	**Outcome**		**Performance**		**Baseline rand. access sim.**	
			Slots	*Col-ns*	*Slots*	*Col-ns*
SG(1)	Network top. update completed	✓	6	1	8.2 [4 - 15]	2.1 [1 - 5]
SG(2)	Joining node found slot 5	✓	3	0	13 [3 - 29]	3.0 [0 - 8]
SG(3)	Network top. update completed	✓	5	1	7.2 [4 - 14]	2.1 [1 - 5]
SG(4)	Network top. update completed	✓	6	1	6.0 [4 - 8]	1.0 [1 - 1]
SG(5)	Joining node found slot 3	✓	7	2	7.4 [1 - 17]	1.0 [0 - 3]
SG(6)	Joining node found slot 2	✗	7	1	7.2 [4 - 15]	2.1 [1 - 5]
SG(7)	Network top. update completed	✓	10	1	7.2 [4 - 15]	2.1 [1 - 5]
MS(1)	Joining node found slot 4	✓	4	2	6.5 [2 - 15]	2.0 [0 - 6]
MS(2)	Joining node found slot 4	✓	7	1	13 [1 - 32]	2.0 [0 - 5]
MS(3)	Network top. update completed	✓	4	1	9.0 [4 - 18]	3.0 [1 - 7]
MS(4)	Network top. update completed	✓	7	2	6.5 [6 - 7]	2.0 [2 - 2]
MS(5)	Network top. update did not reach all nodes	✗	-	1	9.0 [8 - 10]	1.5 [1 - 2]

concluded that slot 2 is collision free. In the subsequent experiment SG(7), the joining node was moved to a different location and depth to improve the link quality, and the same scenario resulted in the expected outcome. Similarly, the failed outcome of the MS(5) experiment was due to the loss of the "topology update" packet from N196 to N194 because of strong interference from N006 reusing the same slot. Since in this topology N194 was the only node connecting two parts of the network, the packet loss at this node blocked the distribution of the topology update to the rest of the network.

Both of these experiments highlight that there is scope for future work on increasing the reliability of JOIN, e.g., by slightly increasing the control overhead and occasionally using ACKs/retransmissions for crucial tasks such as the network topology update. However, overall the sea experiments have demonstrated the good performance and practical feasibility of the JOIN protocol, while also validating the simulation results from Section V.

VII. CONCLUSION

In this article, the JOIN protocol was presented to allow the integration of new nodes into an existing IoUT network without the control signaling overhead of typical state-of-the-art methods, and without reducing the network throughput capacity by using dedicated communication resources for this task. In the JOIN solution, a new node can join an active network by gathering the local topology and packet schedule information via an initial listening phase and incorporating it into a probabilistic model that allows it to choose when to communicate with the nearby nodes such that the expected number of collisions is minimized. A large set of Monte Carlo simulations showed that the proposed intelligent slot selection approach converges to a collision-free solution more quickly and significantly reduces the number of collisions caused by the joining node, compared with the baseline opportunistic random access protocol. Crucially, the protocol performance is insensitive to the network size, demonstrating its scalability, which is a key feature of IoUT networks. Furthermore, the proactive network topology updates carried out by the incumbent network nodes, upon predicting the no-solution outcome at the joining node, considerably streamline the network adaptation process. The empirical evaluation of the JOIN protocol was completed with full-scale trials in the Sea of Galilee and the Mediterranean Sea, thus demonstrating its practical feasibility in real UAN deployments. The trial outcomes were consistent with the simulation results and highlighted the need for future work on the tradeoff between control overhead and reliability of the protocol in a harsh acoustic communication environment. For example, useful future work on the JOIN protocol is to quantify the reliability benefits of increasing the amount of control information shared by the one-hop neighbors with the joining node and the throughput cost of such communications.

ACKNOWLEDGMENT

The authors would like to thank Shlomi Dahan and Liav Nagar for their help with the sea trials, and Dr. Ben Sherlock and Jeff Neasham from Newcastle University for providing the acoustic modems.

REFERENCES

[1] C.-C. Kao, Y.-S. Lin, G.-D. Wu, and C.-J. Huang, "A comprehensive study on the Internet of underwater things: Applications, challenges, and channel models," *Sensors*, vol. 17, no. 7, p. 1477, 2017.

[2] G. Cario, A. Casavola, P. Gjanci, M. Lupia, C. Petrioli, and D. Spaccini, "Long lasting underwater wireless sensors network for water quality monitoring in fish farms," in *Proc. IEEE OCEANS*, 2017, pp. 1–6.

[3] A. K. Mohapatra, N. Gautam, and R. L. Gibson, "Combined routing and node replacement in energy-efficient underwater sensor networks for seismic monitoring," *IEEE J. Ocean. Eng.*, vol. 38, no. 1, pp. 80–90, Jan. 2013.

[4] B. J. Boom *et al.*, "A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage," *Ecol. Inform.*, vol. 23, pp. 83–97, Sep. 2014.

[5] S. Ali *et al.*, "SimpliMote: A wireless sensor network monitoring platform for oil and gas pipelines," *IEEE Syst. J.*, vol. 12, no. 1, pp. 778–789, Mar. 2018.

[6] A. Vasilijevic, D. Nad, and N. Miskovic, "Autonomous surface vehicles as positioning and communications satellites for the marine operational environment—Step toward Internet of underwater things," in *Proc. IEEE USYS*, 2018, pp. 1–5.

[7] H. Holma and A. Toskala, *LTE for UMTS: Evolution to LTE-Advanced*. New York, NY, USA: Wiley, 2011.
[8] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "IEEE 802.11ax: Challenges and requirements for future high efficiency WiFi," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 130–137, Jun. 2017.
[9] M. A. Ronai and E. Kail, "A simple neighbour discovery procedure for Bluetooth ad hoc networks," in *Proc. IEEE GLOBECOM*, vol. 2, 2003, pp. 1028–1032.
[10] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.
[11] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2005, pp. 1–8.
[12] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: Applications, advances and challenges," *Philos. Trans. R. Soc. A*, vol. 370, no. 1958, pp. 158–175, 2012.
[13] S. Jiang, "State-of-the-art medium access control (MAC) protocols for underwater acoustic networks: A survey based on a MAC reference model," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 96–131, 1st Quart., 2018.
[14] K. Kredo, II, P. Djukic, and P. Mohapatra, "STUMP: Exploiting position diversity in the staggered TDMA underwater MAC protocol," in *Proc. IEEE INFOCOM*, 2009, pp. 2961–2965.
[15] P. Anjangi and M. Chitre, "Propagation-delay-aware unslotted schedules with variable packet duration for underwater acoustic networks," *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 977–993, Oct. 2017.
[16] S. Lmai, M. Chitre, C. Laot, and S. Houcke, "Throughput-efficient super-TDMA MAC transmission schedules in ad hoc linear underwater acoustic networks," *IEEE J. Ocean. Eng.*, vol. 42, no. 1, pp. 156–174, Jan. 2017.
[17] X. Zhuo, F. Qu, H. Yang, Y. Wei, Y. Wu, and J. Li, "Delay and queue aware adaptive scheduling-based MAC protocol for underwater acoustic sensor networks," *IEEE Access*, vol. 7, pp. 56263–56275, 2019.
[18] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 116–126, Sep. 1997.
[19] M. Orlinski and N. Filer, "Neighbour discovery in opportunistic networks," *Ad Hoc Netw.*, vol. 25, pp. 383–392, Feb. 2015.
[20] A. Kanzaki, T. Hara, and S. Nishio, "An adaptive TDMA slot assignment protocol in ad hoc sensor networks," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2005, pp. 1160–1165.
[21] J. Liu, F. Ren, L. Miao, and C. Lin, "A-ADHOC: An adaptive real-time distributed MAC protocol for vehicular ad hoc networks," *Mobile Netw. Appl.*, vol. 16, no. 5, pp. 576–585, 2011.
[22] B. A. Sharp, E. A. Grindrod, and D. A. Camm, "Hybrid TDMA/CSMA protocol for self managing packet radio networks," in *Proc. IEEE ICUPC*, 1995, pp. 929–933.
[23] S. Gobriel, D. Mosse, and R. Cleric, "TDMA-ASAP: Sensor network TDMA scheduling with adaptive slot-stealing and parallelism," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2009, pp. 458–465.
[24] A. K. Othman, A. E. Adams, and C. C. Tsimenidis, "Node discovery protocol and localization for distributed underwater acoustic networks," in *Proc. AICT-ICIW*, 2006, pp. 1–6.
[25] M. Watfa, S. Selman, and H. Denkilkian, "UW-MAC: An underwater sensor network MAC protocol," *Int. J. Commun. Syst.*, vol. 23, no. 4, pp. 485–506, 2010.
[26] R. Petroccia, "A distributed ID assignment and topology discovery protocol for underwater acoustic networks," in *Proc. IEEE UComms*, 2016, pp. 1–5.
[27] Z. Guo, G. Colombi, B. Wang, J. Cui, D. Maggiorini, and G. P. Rossi, "Adaptive routing in underwater delay/disruption tolerant sensor networks," in *Proc. IEEE Wireless Demand Netw. Syst. Service (WONS)*, 2008, pp. 31–39.
[28] M. K. Park and V. Rodoplu, "UWAN-MAC: An energy-efficient MAC protocol for underwater acoustic wireless sensor networks," *IEEE J. Ocean. Eng.*, vol. 32, no. 3, pp. 710–720, Jul. 2007.
[29] R. Diamant, G. N. Shirazi, and L. Lampe, "Robust spatial reuse scheduling in underwater acoustic communication networks," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 32–46, Jan. 2014.
[30] P. Björklund, P. Värbrand, and D. Yuan, "A column generation method for spatial TDMA scheduling in ad hoc networks," *Ad Hoc Netw.*, vol. 2, no. 4, pp. 405–418, 2004.
[31] Z.-J. Zhou, G.-Y. Hu, B.-C. Zhang, C.-H. Hu, Z.-G. Zhou, and P.-L. Qiao, "A model for hidden behavior prediction of complex systems based on belief rule base and power set," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1649–1655, Sep. 2018.
[32] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[33] S. Atmaca, "Improving TDMA channel utilization in random access cognitive radio networks by exploiting slotted CSMA," *Wireless Pers. Commun.*, vol. 71, no. 4, pp. 2417–2430, Aug. 2013.
[34] N. Morozs *et al.*, "Robust TDA-MAC for practical underwater sensor network deployment: Lessons from USMART sea trials," in *Proc. ACM WUWNet*, 2018, pp. 1–18.

Nils Morozs (Member, IEEE) received the M.Eng. and Ph.D. degrees in electronic engineering from the University of York, York, U.K., in 2012 and 2015, respectively.

His Ph.D. research was part of the EU FP7 ABSOLUTE project, where he developed LTE-compliant dynamic spectrum access methods for disaster relief and temporary event networks. He worked as a Researcher in Wi-Fi and wireless convergence with BT, Martlesham, U.K. He is currently a Research Associate with the Department of Electronic Engineering, University of York, working on channel modeling and protocol design for underwater acoustic networks. His research interests include the development of protocols and architectures for wireless radio and acoustic networks.

Paul D. Mitchell (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees in electronic engineering from the University of York, York, U.K., in 1999 and 2003, respectively.

He is a Reader of electronic engineering with the University of York, York, U.K. He has previously worked with BT, Martlesham, U.K., and QinetiQ, Farnborough, U.K. He has authored over 130 refereed journal and conference papers and has served on numerous international conference programme committees, including ICC and VTC. He has secured more than $>$ £2.3M + €4.7M funding as a Principal and Co-Investigator. Current projects include Research Council grants on smart dust for large-scale underwater wireless sensing, full-duplex underwater acoustic communications, as well as industrial projects. Primary research interests lie in radio resource management and medium access control for wireless communication systems, including underwater acoustic communication networks, terrestrial wireless sensor networks, and satellite systems.

Dr. Mitchell was the General Chair of the International Symposium on Wireless Communications Systems in 2010. He currently serves as an Associate Editor for *IET Wireless Sensor Systems*, the *International Journal of Distributed Sensor Networks*, and *MDPI Electronics*, and has experience as the Guest Editor and as a reviewer for a number of IEEE, ACM, and IET journals. He is a member of IET.

Roee Diamant (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, Israel, in 2002 and 2007, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, in 2013.

From 2001 to 2009, he worked as a Project Manager and a Systems Engineer with Rafael Advanced Defense Systems, Haifa, where he developed a commercial underwater modem with network capabilities. From 2015 to 2016, he was a Visiting Professor with the University of Padua, Padua, Italy. He is currently the Coordinator of the EU H2020 Project SYMBIOSIS (BG-14 track, ending on October 2020), and leads the Underwater Acoustic and Navigation Laboratory as an Assistant Professor with the Department of Marine Technologies. His research interests include underwater acoustic communication, underwater navigation, object identification, and classification.

Dr. Diamant received the Israel Excellent Worker First Place Award from the Israeli Presidential Institute in 2009 and the NSERC Vanier Canada Graduate Scholarship in 2010. He has received three best paper awards. He serves as an Associate Editor for the IEEE JOURNAL OF OCEANIC ENGINEERING.