IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 April 2023, accepted 14 May 2023, date of publication 17 May 2023, date of current version 31 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3277395

RESEARCH ARTICLE

Low-Latency Low-Energy Adaptive Clustering Hierarchy Protocols for Underwater Acoustic Networks

ABDALLAH S. GHAZY[1], GEORGES KADDOUM[1,2], (Senior Member, IEEE), AND SATINDER SINGH[3], (Member, IEEE)

[1]École de Technologie Supérieure, University of Quebec, Montreal, QC H3C 1K3, Canada
[2]Cyber Security Systems and Applied AI Research Center, Lebanese American University, Beirut 1102-2801, Lebanon
[3]Ultra Group, Columbia, IN 46725, USA

Corresponding author: Abdallah S. Ghazy (Abdallah.ghazy@etsmtl.ca)

This work was supported by the Resilient Machine Learning Institute (ReMI), Montreal, QC, Canada.

ABSTRACT Underwater acoustic cluster networks (UACNs) are commonly used due to their adaptability in dynamic underwater environments. While the low-energy adaptive clustering hierarchy (LEACH) protocol was initially designed for radio frequency (RF) cluster networks, it has also been applied to UACNs. However, the LEACH protocol uses lengthy overheads per packet due to its use of global or wide-scale IDs, leading to increased communication latency and energy usage per packet. To address this issue, we propose the low-latency low-energy adaptive clustering hierarchy (L^3EACH) protocol. The L^3EACH protocol is a comprehensive framework that integrates ID assignment, time slot reservation, packet routing, and self-network organization. The protocol uses shorter overheads by assigning local IDs instead of global or wide-scale IDs. It assigns unique IDs to nodes within a cluster and reassigns the same IDs to nodes in other clusters, i.e., spatial ID reuse. The protocol also allocates IDs and time slots on demand to maximize network resources. To further enhance the protocol, we introduce the L^3EACH-Version 2 (L^3EACH-V2) protocol, which modulates the preamble bits to embed the IDs in the overhead rather than inserting extra bits. We also provide the computational complexity of the L^3EACH-V2 protocol. Compared to the DIVE protocol, the L^3EACH protocol reduces the ID length and average energy per packet by 50% and 13%, respectively. Furthermore, the L^3EACH-V2 protocol reduces the average energy per packet by 27% and increases the network throughput by 16% compared to the L^3EACH protocol, making it an efficient and scalable solution, especially, for dense UACNs.

INDEX TERMS Underwater acoustic sensor networks, cluster topologies, node identification, preamble modulations, multiple access network, routing protocol, self-network organization.

I. INTRODUCTION

Over 70% of the Earth's surface is covered by oceans, which play a crucial role in the ecosystem [1]. Underwater wireless sensor networks (UWSNs) are widely used for various purposes, including ecosystem monitoring, resource discovery, safety, and security [2]. However, the dynamic topology of UWSNs presents deployment challenges, as highlighted in several studies [3], [4], [5], [6], [7], [8], [9], [10]. To address these challenges, UWSNs can be configured as ad hoc networks with star, tree, mesh, or cluster topologies. Cluster topologies divide the network into groups, each consisting of a cluster head and several cluster members [11], [12], [13], [14], [15]. Cluster heads establish multi-hop transmission between the source and sink, reducing energy consumption while mitigating impairments of acoustic channels, such as multi-path fading and beam bending [1], [16]. Therefore, cluster topologies are preferred over other topologies.

However, using conventional cluster-based protocols can lead to a reduction in network lifetime due to the energy depletion of cluster heads [17]. To address this issue, Heinzelman et al. introduced the low-energy adaptive clustering

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluca Cena.

hierarchy (LEACH) protocol for radio frequency (RF) networks [17], [18], which has since been adapted for underwater acoustic clustering networks (UACNs) [11], [12], [13], [14], [15], [19]. The LEACH protocol extends the network lifetime by aggregating data and rotating the cluster head role among sensor nodes. Several protocols based on the LEACH protocol have been proposed for underwater communications to improve multiple access control (MAC) schemes and routing mechanisms in cluster networks [21], [22], [23]. For example, Salva-Garau and Stojanovic proposed a time division multiple access (TDMA) scheme that utilizes code division multiple access (CDMA) to enable spatial reuse of time slots across the network [24]. Byun et al. introduced a contention-based MAC scheme that incorporates an initialization phase for time slot requests and a transmission phase for data packet transmission [25]. Ahmad et al. proposed a routing protocol that distributes functions equally among all nodes to reduce the energy consumption [26]. Khan et al. proposed a multi-layer routing protocol that addresses load transmission and energy consumption imbalances in sensor nodes [27].

Due to assigning long identifications (IDs) for sensor nodes, LEACH-based MAC and routing protocols have high overheads. Many schemes were proposed to assign IDs to sensor nodes in the literature, such as global and wide-scale ID schemes [28]. The global ID scheme uses the physical IDs of devices, which is simple to implement. However, it has a high overhead since the physical IDs are too long compared to the payload. In addition, it requires a high level of coordination between the vendors of acoustic technology, which is challenging. The wide-scale ID schemes, such as preconfiguration and auto-assignment schemes, provide relatively shorter IDs. The preconfiguration scheme assigns IDs to sensor nodes before the network deployment. However, this scheme is not scalable since adding sensor nodes to the network is impossible after deployment [28]. On the other hand, auto-assignment schemes assign unique IDs to every node automatically after the deployment, and they resolve ID conflicts [29], [30], [31], [32]. However, the wide-scale ID schemes assign IDs with a length that grows logarithmically with the number of sensor nodes. Hence, they are not appropriate for deployment in dense networks since they reduce the throughputs and energy efficiency of UACNs [28], [29], [30].

On the other hand, the preamble signal is a well-known special pattern, prepended to the packet header at the transmitter and used by the receiver to perform frame synchronization and channel estimation. In RF communication systems, there have been many works on modulating the preamble signals to carry extra information, including but not limited to control signals, beam and cell IDs, and authentication messages [33], [34], [35], [36]. In underwater communications, few works have considered modulating the preamble signals, such as *Asim et al.* [37] and *Rothan et al.* [38], who proposed embedding user IDs and cell IDs, respectively.

To the best of our knowledge, no protocol has been proposed to reduce the ID length by utilizing the structure of cluster networks and preamble signals [19], [28]. Therefore, in this paper, we propose a low-latency low-energy adaptive clustering hierarchy (L^3EACH) and L^3EACH-Version 2 (L^3EACH-V2) protocols. The protocols are complete frameworks that integrate: ID assignment, time slot reservation, packet routing, and self-network organization. The contributions of this work are summarized as the following:

- We propose the L^3EACH protocol to reduce the communication latency and energy per packet in UACNs. The protocol utilizes the clustering architecture to reuse the same IDs spatially in different clusters, i.e., local unique IDs. The protocol also reserves IDs and time slots for the sensor nodes on demand to maximize the network resources.
- For further improvement, we propose the L^3EACH-V2 protocol, which dramatically shrinks the overhead field of the packets. In contrast to the L^3EACH protocol, the L^3EACH-V2 protocol embeds the IDs implicitly in the overhead by modulating the preamble signal.
- We propose a criterion that optimizes the detection threshold to minimize the error probability of preamble demodulations in the case of the L^3EACH-V2 protocol.
- The computational complexity of the L^3EACH-V2 protocol is introduced to show the extra digital signal processing (DSP) required over the L^3EACH protocol. The computational complexity of the L^3EACH-V2 is the only significant cost paid to get its advantages.
- In numerical results, we show the average length of the IDs and average energy per packet used in the cases of the L^3EACH and DIVE protocols. Furthermore, we evaluate the probabilities of missed and false ID detection of the L^3EACH-V2 protocol. In addition, we compare the successful packet delivery cases of the L^3EACH-V2, L^3EACH, and DIVE protocols. Lastly, we analyze the computational complexity of the L^3EACH-V2 protocol versus the network density (i.e., the number of sensor nodes in the network).

The rest of the paper is organized as follows. Section II discusses the related works in the literature in more detail. In Section III, we introduce the architecture of cluster networks and link models. Section IV presents the framework of the L^3EACH protocol. Section V introduces the framework of the L^3EACH-V2 protocol and shows the analysis of its computational complexity. In Section VI, we numerically assess the network performance of the L^3EACH and L^3EACH-V2 protocols and compare their performances with the DIVE protocol. Finally, conclusions, challenges, and future works are presented in SectionVII.

II. RELATED WORKS IN THE LITERATURE

In the literature, several protocols have been proposed to assign wide-scale IDs to sensor nodes, including [29], [30], [31], [32]. These protocols aim to minimize the number of bits required for each ID, in order to reduce the overhead required by communication protocols and extend the network's lifespan. For instance, Agrawal et al. [29]

proposed a protocol that simplifies the ID assignment and resolution process, to deal with conflicts and ensure each node receives a unique ID. It also provides information about network addresses to all nodes, allowing for name resolution. The protocol is designed in a decentralized manner, removing the necessity of a central node or database to oversee address management. Zhao et al. [30] proposed an efficient topology discovery protocol (ETDP) that enables adaptive node ID assignment and topology discovery simultaneously. The protocol reduces energy consumption and packet collisions by controlling transmissions based on a local timer and dividing the network into different layers. Petroccia [31], [32] proposed the DIVE protocol, which is a self-adaptive distributed solution for assigning unique IDs and discovering topology. The protocol consists of two primary procedures that operate concurrently. The initial procedure is responsible for distributing the necessary information to assign node IDs and detect the network topology. The second procedure is necessary to ensure that node IDs are unique throughout the entire network. The protocol includes built-in procedures that guarantee the assignment of unique IDs in the network, even in unreliable communication channels.

These mentioned protocols assign IDs with lengths that increase logarithmically with the number of sensor nodes, which reduces the throughput and energy efficiency of dense networks [28], [29], [30]. On the other hand, the L^{3}EACH and L^{3}EACH-V2 protocols assign local unique IDs whose length does not increase with the number of sensor nodes. Local IDs are much shorter than wide-scale ones, especially in dense networks. Moreover, the L^{3}EACH-V2 protocol embeds the local IDs implicitly in the preamble signal rather than adding them to the overhead. Furthermore, our proposed protocols integrate ID assignment, time slot reservation, packet routing, and self-network organization. The ID assignment process is combined with the time slot reservation process to eliminate collisions. The IDs and time slots are assigned to the nodes on demand to maximize network resources. Relative to the related protocols in the literature, the proposed protocols significantly prolong the lifetime and increase the achievable throughput of UACNs.

III. NETWORK ARCHITECTURES OF THE L^{3}EACH AND L^{3}EACH-V2 PROTOCOLS

This section introduces the cluster topology and models the links of the L^{3}EACH and L^{3}EACH-V2 protocols.

A. UNDERWATER CLUSTER TOPOLOGIES

Figure 1 illustrates the cluster topology, where sensor nodes are organized into small and larger clusters. Within a small cluster, one sensor node acts as the cluster head (SeCH), and the remaining sensors function as sensor cluster members (SeCMs). Similarly, within a large cluster, one sink node acts as the sink cluster head (SiCH), and SeCH nodes as cluster members. Sensor nodes compare the strengths of the received signals from cluster heads and join the cluster of the closest cluster head. Once the clusters are created, the

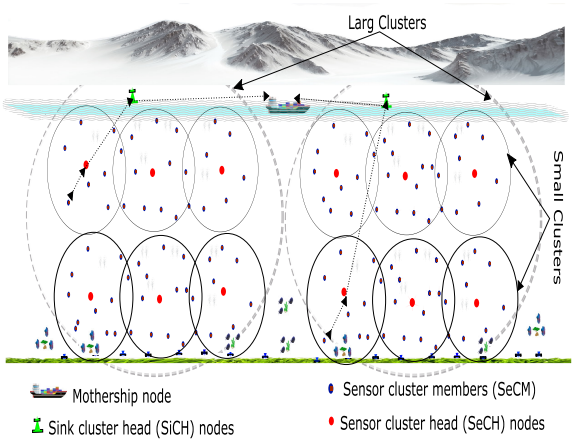

FIGURE 1. A general hierarchy shows the cluster topology of the L^{3}EACH and L^{3}EACH-V2 protocols. The clip arts, in this figure, are reproduced from [39].

communication setup proceeds from SiCHs to SeCMs (i.e., downward), and the transmission begins from SeCMs to SiCHs (i.e., upward) [17], [18], [20], [27]. Due to the broadcast nature of acoustic communications, packet collisions may occur within the same cluster (i.e., intra-collisions) or between neighboring clusters (i.e., inter-collisions). Cluster heads assign time slots and CDMA codes to their members to eliminate intra and inter-collisions, respectively [21], [22], [24]. To prolong the network's lifetime, the LEACH protocol applies the following tactics. First, the role of the cluster head is rotated among the sensor nodes. The nodes compress the correlated data into uncorrelated ones (i.e., data aggregation) before the transmissions. Cluster members turn their transmitters on only when they have new data ready for transmission. However, the cluster heads always keep their transceivers on to allocate the resources and receive and forward the packets [17].

B. UNDERWATER ACOUSTIC LINK MODEL

Underwater acoustic links are affected by path loss, interference, thermal noise, multi-path fading, Doppler dispersions, and beam bending [1], [16]. In UACNs, long links between nodes are partitioned into shorter links of a few kilometers [40], [41]. Although sensor nodes are assumed to be stationary in seawater, they can undergo unintentional movements with a small velocity of 0.5 m/s due to currents and waves [42]. Short-range quasi-stationary links can be modeled while neglecting multi-path fading, beam bending, and Doppler dispersion [37], [42], [43]. The effects of path loss, interference, and noise are frequency dependent. Turbulence and ship interferences only affect low frequencies (tens to a few hundred Hz), while wind interference and thermal noise only affect high frequencies (above a few hundred Hz) [1], [16], [45]. In this paper, we assume a high carrier frequency, and we consider only the impacts of wind interference and thermal noise in addition to the path loss [1], [16].

The path loss of seawater is induced by absorption and spreading losses. The path loss between the i^{th} SiCH node

and the j^{th} SeCH node over a distance L_{ji} is given as [16]

$$10\ \log A_{ji}(f) = 10\, L_{ji} \log a(f) + 10\, \mu\ \log(\,10^3\, L_{ji}),\ \text{dB}. \tag{1}$$

In this equation, the first term represents the absorption loss, where $a(f)$ is the absorption coefficient. The second term represents the spreading loss, where μ is the spreading factor.[1] The distance, L_{ji}, is given in kilometers, $i \neq j \in \{1, 2, \ldots, N\}$, N is the total number of nodes (i.e., sinks and sensors), and f is the frequency in kHz. For high carrier frequencies, the absorption coefficient can be expressed empirically using Thorp's formula as [1]

$$10\log(a(f)) = \frac{0.11 f^2}{1+f^2} + \frac{44 f^2}{4100+f^2} + 2.75 \times 10^{-4} f^2 + 0.003,\ \text{dB/Km}. \tag{2}$$

This equation shows that path-loss increases nonlinearly with the frequency, implying that high-frequency communications require higher transmission powers than low-frequency ones. For a given spectrum of transmission power, $P^t_{ji}(f)$, the received power is given as

$$P^r_{ji} = \int_{f_{min}}^{f_{max}} A^{-1}_{ji}(f)\, P^t_{ji}(f)\, df,\ \text{Watt}, \tag{3}$$

where f_{min} and f_{max} are the minimum and maximum frequencies of the spectrum, respectively.

The wind interference plus thermal noise of the j^{th} receiver, $IN_j(f)$, is expressed as

$$IN_j(f) = \frac{6.3492}{10^{17}} \left(I^w_j(f) + N^{th}_j(f)\right),\ \text{Watt/Hz}, \tag{4}$$

where the wind interference, $I^w_j(f)$, is computed as [45]

$$I^w_j(f) = 10^{(7.5\, w^{1/2} + 20\ \log(f) - 40\ \log(f+0.4) + 50)/10}, \tag{5}$$

where w is the wind speed in m/s. The thermal noise, $N^{th}_j(f)$, is computed as [45]

$$N^{th}_j(f) = 10^{(-15 + 20\ \log(f))/10}. \tag{6}$$

The signal-to-interference plus noise ratio ($SINR_{ji}$) is obtained as

$$SINR_{ji} = \frac{P^r_{ji}}{\int_{f_{min}}^{f_{max}} IN_j(f)\, df}. \tag{7}$$

The $SINR_{ji}$ is a function of the link range, and cluster members use it to identify the closest cluster heads. Also, $SINR_{ji}$ is frequency dependent, and it can be optimized by selecting the appropriate operating frequency based on the link range. For instance, the communication links between SiCH and SeCH nodes (i.e., long-range in order of tens of kilometers) should use lower operating frequencies. In contrast, the communication links between SeCM and SeCH nodes (i.e., short-range in order of a few kilometers) should use higher operating frequencies [1], [16], [45].

[1]The spreading factor μ describes the geometric spreading of the acoustic waves, where $\mu = \{1, 1.5, 2\}$ for cylindrical, practical, and spherical spreads, respectively.

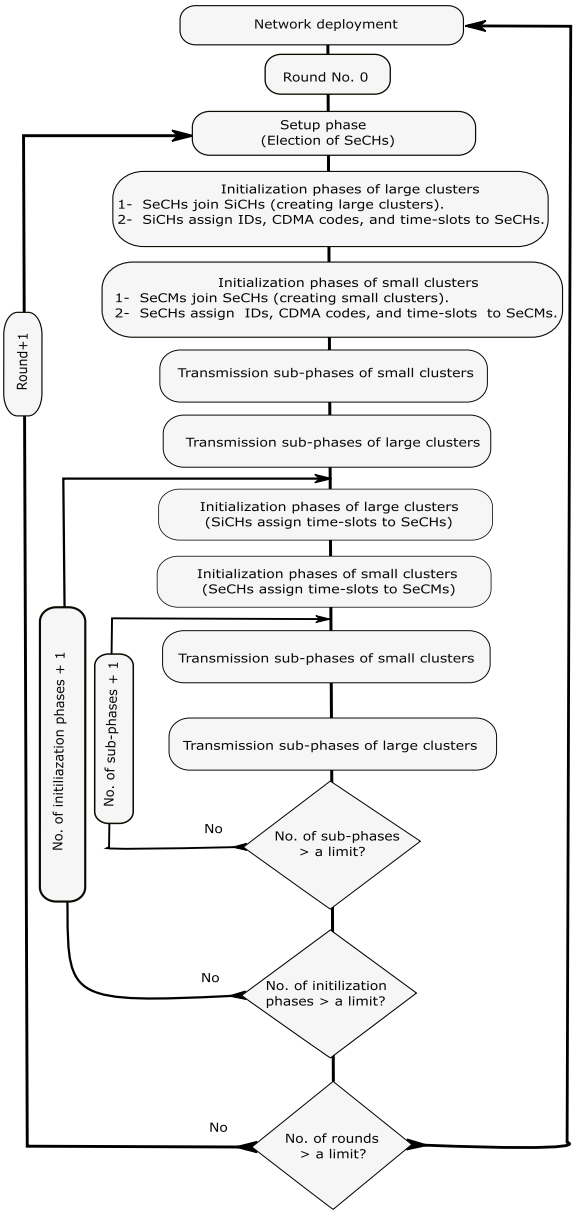

FIGURE 2. Process flow of the L^{3}EACH protocol.

IV. LOW-LATENCY LOW-ENERGY ADAPTIVE CLUSTERING HIERARCHY (L^{3}EACH) PROTOCOL

The L^{3}EACH protocol is a comprehensive framework incorporating ID assignment, time slot reservation, packet routing, and self-network organization. The protocol dynamically allocates resources on demand,[2] including IDs, CDMA codes, and time slots. The ID and CDMA code assignments are based on spatial reusing used in radio frequency (RF) networks [46], [47]. Additionally, time slot reservation is inspired by reserved-channel MAC and Aloha-random backoff schemes [25], [48]. The L^{3}EACH protocol reduces packet overheads, enhances the energy efficiency of the networks, and reduces network congestion. Section IV-A bellow introduces the protocol in more detail.

[2]The protocol does not allocate resources to nodes that do not have data for transmission, such as sleeping, dead, and lost nodes.

A. THE PROTOCOL MODEL

Figures 2 and 3 depict the process flow and time diagram of the L^{3}EACH protocol, respectively. Additionally, Algo. 1 presents the pseudocode of its implementation.

The process flow depicted in Fig. 2 divides the process into rounds, each consisting of several phases: one setup phase and multiple initialization and transmission phases. The cluster heads are elected for a current round during the setup phase. Initialization and transmission phases are repeated in each round to allow all cluster members to access the medium within the same round without waiting for the next one. The first initialization phase in each round performs cluster creation, and ID, CDMA, and time slot assignments. Specifically, the SiCH nodes create large clusters and allocate the resources to the SeCH nodes. Then, the SeCH nodes create small clusters and allocate the resources to the SeCM nodes. The subsequent initialization phases only repeat the time slot assignment process since the rest of the processes (i.e., cluster creation, and ID and CDMA assignments) are performed once per round to minimize control messages. Transmission phases are subdivided into sub-phases, allowing nodes to send large packets within the same transmission phase and maximize fairness.

Figure 3 provides profound insight into the time process of the protocol.[3] All parameters shown in the figure are defined in Table 1. The initialization phase involves bidirectional communication between the cluster heads (SiCH and SeCH nodes) and cluster members (SeCH and SeCM nodes), as denoted by the blue and green colors, respectively. They exchange three messages: "*Hi*", "*request to send (RTS)*", and "*clear to send (CTS)*" messages. Initially, cluster heads broadcast "*Hi messages*" to sensor nodes for creating clusters **(line No. 16 in Algo. 1)**. Then, sensor nodes respond to the cluster heads with "*RTS messages*" to request memberships and resources **(line No. 27 in Algo. 1)**. Finally, cluster heads broadcast "*CTS messages*" to confirm the memberships and allocate the resources **(line No. 43 in Algo. 1)**. On the other hand, transmission phases are unidirectional communications, where cluster members send data packets to cluster heads in the reserved time slots **(lines No. 46-52 in Algo. 1)**.

The time intervals for each phase are illustrated in Fig. 3. The round interval, denoted as τ_r, is calculated by summing up the intervals of all phases as

$$\tau_r = t^o_{r+1} - t^o_r = \tau_s + M_t\,(\tau_{in} + M_{sp}\,\tau_{sp}), \tag{8}$$

where t^o_r is the beginning time of the r^{th} round, τ_s is the setup interval, τ_{in} is the initialization interval, τ_{sp} is the sub-phase interval, and M_t and M_{sp} are the number of transmission phases per round and sub-phase per transmission phase, respectively. The number of sub-phases per transmission phase equal to

$$M_{sp} = \left\lfloor \frac{\tau_t}{\tau_{sp}} \right\rfloor, \tag{9}$$

where τ_t is the transmission phase interval, and the symbol $\lfloor x \rfloor$ means the maximum integer number equal to or less than the value of x.

The details of the phases are introduced in the following subsections.

[3] In the figure, ten sensor nodes per cluster are considered for illustration purposes only; however, the protocol can be applied to a larger number of nodes.

TABLE 1. Definitions of the parameters shown in Figure 3.

τ_r	Round duration.
τ_s	Setup phase duration.
τ_{in}	Initialization phase interval.
τ_t	Transmission phase duration.
τ_{sp}	Transmission sub-phase duration.
τ_{cts}	*CTS message* duration.
τ_{rts}	*RTS message* duration.
τ_h	*Hi message* duration.
τ^p_{ji}	Time slot duration, including data and guard intervals, assigned to i^{th} node by j^{th} node.
τ^d_{ji}	Data interval assigned to j^{th} node by i^{th} node.
τ^g_i	Guard interval between two consecutive time slots of i^{th} node.
τ^p_i	Time period of transmission of j^{th} sensor node.
τ^w_j	Waiting interval of j^{th} sensor node after receiving *CTS messages* and before beginning the transmission.
t^o_r	Instant beginning time of the r^{th} round.
t^t_{ji}	Instant of transmitting time.
t^{cts}_{ji}	Arrival time of the *CTS message* to the j^{th} SeCH.
t^a_{ji}	Arrival time of the packet to the i^{th} node sending by j^{th} node.
t^p_{ji}	Propagation time between j^{th} node and i^{th} nodes.
t^w_i	Waiting time of i^{th} of sensor node.
m_t	Index of the current transmission phase.
$M_t - 1$	Number of transmission phases per round.
M_{sp}	Number of transmission sub-phases per transmission phase.
r	Index of the current round.
$R + 1$	A total number of rounds per network deployment.

B. SETUP PHASE

The L^{3}EACH protocol initiates rounds with setup phases to elect SeCH nodes. In the first setup phase, all sensor nodes have an equal chance of being elected as SeCH nodes with an election probability P. Once elected, a sensor node cannot be selected again in the next $1/P$ setup phases. After $1/P$ setup phases, all sensor nodes are eligible to be elected again as SeCH nodes **(lines No. 5-9 in Algo. 1)**. The i^{th} node generates a random number between 0 and 1, and if this number is less than a threshold $\Lambda(i)$, the node becomes a SeCH node for the current round. The threshold $\Lambda(i)$ is calculated as [18]

$$\Lambda(i) = \begin{cases} \dfrac{P}{1 - P \times (r \times mod(\frac{1}{P}))}, & \text{if} \quad i \in G \\ 0, & \text{Otherwise,} \end{cases} \tag{10}$$

where $i = \{1, 2, \ldots, N_s\}$, N_s is the total number of sensor nodes, $r = \{0, 1, 2, \ldots, R\}$ is the index of the current round, $R+1$ is the total number of rounds, and G is the set of sensor nodes that have not been elected as SeCH in the last $1/P$ rounds. The practical value of P is usually taken in the range [0, 0.5] [17].

C. INITIALIZATION PHASES

Once SeCH nodes are elected, the SiCH nodes wait for random times before broadcasting *Hi messages* to all SeCH nodes saying *"My local ID is {LID} and the CDMA code*

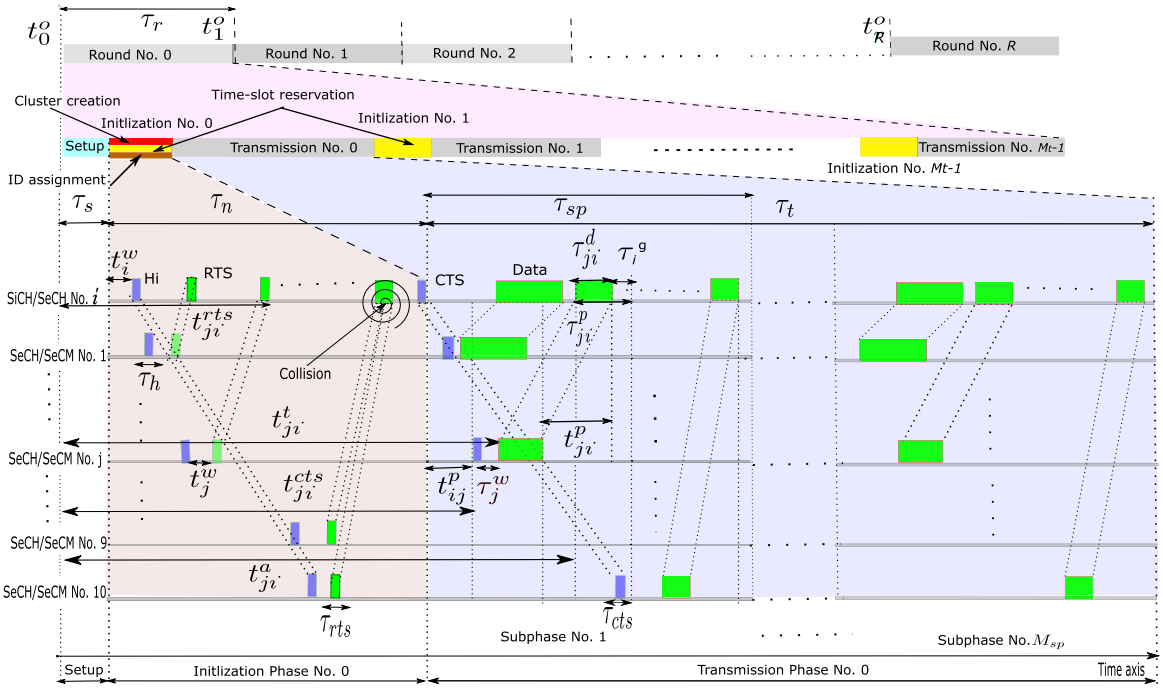

FIGURE 3. Time diagram of the L[3]EACH protocol.

is {CDMA},[4] *Send your globally unique ID"*. Interested SeCH nodes respond to *Hi messages* with *RTS messages* to request resources, while non-interesting nodes discard them. SeCH nodes receive *Hi messages* from many SiCH nodes but respond only to the closest ones based on $SINR_{ji}$ values. SeCH nodes modulate the *RTS messages* using the CDMA codes and wait for random times before sending the *RTS messages* to eliminate collisions. The *RTS messages* should say *"My unique global ID is {GID}"*. They contain the unique global IDs of the SeCH nodes and the local IDs of the SiCH nodes. SeCH nodes generate the unique global IDs using a pseudo-random number generator.

SiCH nodes allocate time resources to SeCH nodes based on the arrival times of *RTS messages*. SiCH nodes only assign resources to SeCH nodes whose *RTS messages* arrived intact, i.e., without collisions **(lines No. 33-34 in Algo. 1)**. As illustrated in Fig. 3, *RTS messages* arrive at SiCH nodes at different times, depending on message length, waiting times, and propagation delay. When the arrival times of two nodes are similar, *RTS messages* collide, as shown in the example of sensor nodes No. 9 and 10 in Fig. 3. The *RTS message* of the j^{th} SeCH node arrives to the i^{th} SiCH at arrival time

$$t_{ji}^{rts} = \underbrace{t_r^o}_{Round\ begining} + \underbrace{\tau_s + m_t\,(\tau_{in} + \tau_t)}_{m_t\ phases} + \underbrace{t_i^w + t_{ji}^p + \tau_h}_{Hi\ message} + \underbrace{t_j^w + t_{ji}^p + \tau_{rts}}_{RTS\ message}, \quad (11)$$

where τ_h and τ_{rts} are the intervals of *Hi* and *RTS messages*, respectively. Symbol m_t is the index of the current transmission phase, where $m_t = \{0, 1, 2, \ldots, M_t - 1\}$. Parameters t_{ji}^p and t_{ij}^p are the propagation times between the j^{th} SeCH and the i^{th} SiCH nodes and vice versa,[5] respectively, where $i \neq j \in \{1, 2, \ldots, N\}$. Symbols t_i^w and t_j^w are the waiting times of the i^{th} SiCH and the j^{th} SeCH nodes, respectively.

SiCH nodes assign new IDs to SeCH nodes by mapping the global IDs to local ones. SiCH nodes extract the global IDs from the received *RTS messages* and order them in ascending order. Then, the SiCH nodes verify that the ordered IDs are unique, and that there is no ID duplication in their clusters **(lines No. 37-40 in Algo. 1)**. Suppose there are duplicated IDs[6]; in that case, the SiCH nodes send unicast messages to the corresponding SeCH nodes saying *"Regenerate a new global unique ID"*. The SiCH nodes keep sending unicast messages until they receive globally unique IDs from the SeCH nodes in their clusters. Then, the SiCH nodes map the unique global IDs into unique local IDs and record them in a cluster members list for the current round. Each round, SiCH nodes update the cluster members list according to the demands of the SeCH nodes and $SINR_{ji}$ values. SiCH nodes could map the global IDs to the local IDs using the Huffman encoder since it shortens the IDs length [46]. The Huffman encoder is commonly used for source encoding in communication systems. Huffman probabilities could be computed based on the channel conditions of the links between SiCH and SeCH nodes **(line No. 41 in Algo. 1)**. The Huffman probability, P_{ji}, for the j^{th} SeCH node clustered with the i^{th}

[4]The mothership assigns local IDs and CDMA codes to the SiCH nodes.

[5]Values of t_{ji}^p and t_{ij}^p should be identical in symmetric channels.

[6]Increasing the length of global IDs reduces the probability of IDs duplication in the initialization phase.

Algorithm 1 Pseudocode for the L^{3}EACH Protocol. In the Shown Pseudocode, Symbols Y and X Represent SiCH/SeCH and SeCH/SeCM Nodes, Respectively

Input: No. of sink and sensor nodes, network dimensions, election probability, and pseudo-random number and waiting time generators.
Output: Unique IDs and non-collided time slots.
/*1- Rounds */
for round $r = \{0, 1, 2, \ldots, R\}$ **do**
  /*2- Setup Phase */
  for sensor node No. $i = \{1, 2, \ldots, N_s\}$ **do**
    Node generates a random number.
    Node computes Eq. (10).
  end for
  New SeCH nodes are elected.
  /*3- Initialization Phases*/
  for Initialization phase No. $\{0, 1, 2, \ldots, M_t - 1\}$ **do**
    for Y number $i = \{1, 2, \ldots\}$ **do**
      Y waits time t_i^w.
      if Initialization phase number == 1 **then**
        Y broadcasts a Hi message asking for global IDs.
      else Y broadcasts Hi message asking local IDs.
      end if
    end for
    for X No. $j = \{1, 2, \ldots\}$ **do**
      if Initialization phase number == 1 **then**
        X computes $SINR_{ji}$ using Eq. (7).
        X joints the closest Y based on the $SINR_{ji}$.
        X gets CDMA code associated with Y
        X generates global ID.
        X waits time t_j^w.
        X sends an RTS message with a global ID.
      else X waits time t_j^w.
        X sends RTS message with local ID.
      end if
    end for
  for Y number $i = \{1, 2, \ldots\}$ **do**
    Y checks the arrival times, t_{ji}^{rts}, of the *RTS messages*.
    Y computes the reserved time slots { t_{ji}^t, τ_{ji}^p, τ_i^p}, Eqs. (15), (18), (19).
    if Initialization phase number == 1 **then**
      Y orders the global IDs.
      if There are duplicated global IDs **then**
        Y sends unicast messages asking for new IDs.
        Go to step No. 7.
      end if
      Y maps the global IDs to local IDs using Eq. (12).
    end if
    Y broadcasts *CTS message* assigning the resources to X.
  end for
  /*4- Transmission Phases*/
  for Sub-phase transmission No. $\{1, 2, \ldots, M_{sp}\}$ **do**
    for X No. $j = \{1, 2, \ldots\}$ **do**
      if X finds its ID in the *CTS message* **then**
        X computes the waiting time τ_j^w, Eq. (17).
        X sends data packets in interval τ_{ji}^d.
      end if
    end for
  end for
end for

SiCH node is computed as

$$P_{ji} = \frac{1 - L_{ji}/L_i}{J_i - \sum_{j=1}^{J_i} L_{ji}/L_i}, \tag{12}$$

where L_i is the radius of the i^{th} cluster. Cluster radius equals the maximum distance between the cluster head and its members, i.e., $L_i = \max\{\ldots, L_{ji}, \ldots\}$. J_i is the total number of SeCH nodes clustered with the i^{th} SiCH node. Equation (12) assigns long and short IDs to SeCH nodes that experience low and high $SINR_{ji}$ values, respectively. This assignment is analogous to source encoding techniques used in classic communication systems [46].

Assuming the network deploys N nodes and runs for $R + 1$ rounds, the average length of the node IDs is computed as

$$l_{av} = \frac{\sum_{r=0}^{R} \sum_{i=1}^{N} l(i, r)}{N \times (R + 1)}, \tag{13}$$

where $l(i, r)$ is the ID length of the i^{th} node in the r^{th} round. For the same network setting, the ratio of average ID length of L^{3}EACH to DIVE protocol is computed as

$$\gamma_l = \frac{\sum_{r=0}^{R} \sum_{i=1}^{N} l_L(i, r)}{\sum_{r=0}^{R} \sum_{i=1}^{N} l_D(i, r)}, \tag{14}$$

where $l_L(i, r)$ and $l_D(i, r)$ are the ID lengths assigned to the i^{th} node by the L^{3}EACH and DIVE protocols, respectively, in the r^{th} round.

D. TRANSMISSION PHASES

SiCH nodes confirm the IDs and time slot resources by broadcasting *CTS messages* to SeCH nodes. *CTS messages* saying *"Global IDs {...,GID_{ji},...} are mapped to local IDs {...,LID_{ji}, ...} for the current round, and the corresponding time slot reservation is {...,t_{ji}^t,τ_{ji}^p,τ_i^p,...} for the this transmission phase."* **(line No. 34 in Algo. 1)**. *CTS messages*

assign transmitting times, t_{ji}^{t}, intervals of transmissions, τ_{ji}^{p}, and periodicities, τ_{i}^{p}. The j^{th} SeCH begins the transmission at the transmitting time

$$t_{ji}^{t} = t_{ji}^{cts} + \tau_{cts} + \tau_{j}^{w}, \tag{15}$$

where t_{ji}^{cts} is the arrival time of the *CTS message* to the j^{th} SeCH, τ_{cts} is the packet length of the *CTS messages*, and τ_{j}^{w} is the waiting time. The arrival time t_{ji}^{cts} is calculated as

$$t_{ji}^{cts} = \underbrace{t_{r}^{o}}_{Round\ begining} + \underbrace{\tau_{s} + m_{t}\,(\tau_{in} + \tau_{t})}_{m_t\ phases} + \underbrace{\tau_{in}}_{Initialization} + \underbrace{t_{ji}^{p} + \tau_{cts}}_{CTS\ message}. \tag{16}$$

SeCH nodes must wait for τ_{j}^{w} after receiving *CTS messages* and before beginning the transmission to avoid collisions. SiCH nodes determine the waiting time as

$$\tau_{j}^{w} = t_{ji}^{a} - (t_{ji}^{cts} + \tau_{cts} + t_{ji}^{p}), \tag{17}$$

where t_{ji}^{a} is arrival time of the packet to the i^{th} SiCH node.

The transmission interval τ_{ji}^{p} is computed based on the interval between the arrival times of two consecutive *RTS messages*. Assuming the *RTS message* of the k^{th} SeCH node arrives at time $t_{k\,i}^{rts}$, and it is the next arrival after the *RTS message* of j^{th} SeCH node, where $k \neq j \in \{1, 2, \ldots, N\}$. The i^{th} SiCH node allocates interval τ_{ji}^{p} to the j^{th} SeCH node according to

$$\tau_{ji}^{p} = \tau_{ji}^{d} + \tau_{i}^{g} = t_{k\,i}^{rts} - t_{ji}^{rts}, \tag{18}$$

where τ_{ji}^{p} is the time slot that includes data and guard intervals, τ_{ji}^{d} and τ_{i}^{g}, respectively. The guard interval is a guard time between consecutive time slots and it is used to overcome intra-collisions due to clock drifts[7] and spatial-temporal uncertainty of acoustic channels [21].

In the current transmission phase, SeCH nodes transmit their data to the i^{th} SiCH with periodicity

$$\tau_{i}^{p} = \sum_{j=1}^{J_i} \tau_{ji}^{p} + J_i\,\tau_{rts}, \tag{19}$$

where J_i is the number of SeCH nodes clustered with the i^{th} SiCH node. SeCH nodes send one or more data packets in each sub-phase based on the lengths of the data packets and the transmitting intervals τ_{ji}^{p}. Short packet length and longer transmitting interval mean more transmitted data packets per sub-phase. If a SeCH node has more data packets than the number of sub-phases, it should wait for the next transmission phase to send the rest. If a SeCH node has fewer data packets than the sub-phase numbers, it should switch to sleeping mode after finishing its transmission. When the current transmission phase is finished, SeCH nodes stop data transmissions and begin a new initialization phase to get new time slots. SeCH nodes that do not find their IDs in the *CTS message* understand that their *RTS messages* have collided **(lines No. 48-51 in Algo. 1)**. Therefore, such nodes are not eligible for transmissions in the current transmission phase and should demand resources in the next initialization phase. For example, in Fig. 3, SeCH No. 9 does not transmit data in the first transmission phase since it did not find its ID in the *CTS message*.

The average energy per packet could be computed as

$$E_{av} = \frac{\sum_{r=0}^{R}\sum_{m_t=0}^{M_t-1}\sum_{i=1}^{N}\left(E^{c}(i,m_t,r) + E^{d}(i,m_t,r)\right)}{\sum_{r=0}^{R}\sum_{m_t=0}^{M_t-1}\sum_{i=1}^{N}\left(\epsilon^{c}(i,m_t,r) + \epsilon^{d}(i,m_t,r)\right)}, \tag{20}$$

where $E^{c}(i, m_t, r)$ and $E^{d}(i, m_t, r)$ are the total energies consumed by the i^{th} node in the m_t^{th} transmission phase and r^{th} round for processing control and data packets, respectively. $m_t \in \{0, 1, 2, \ldots, M_t - 1\}$, where $M_t - 1$ is the total transmission phases per round. $\epsilon^{c}(i, m_t, r)$ and $\epsilon^{d}(i, m_t, r)$ are the total numbers of processed control and data packets, respectively. Mathematically, $E^{c}(i, m_t, r)$ and $E^{d}(i, m_t, r)$ can be formulated as

$$\begin{aligned} E^{c}(i,m_t,r) &= E^{tx}\,\delta^{tc}(i,m_t,r) + E^{rx}\,\delta^{rc}(i,m_t,r),\\ E^{d}(i,m_t,r) &= E^{tx}\,\delta^{td}(i,m_t,r) + E^{rx}\,\delta^{rd}(i,m_t,r)\\ &\quad + E^{ag}\delta^{ag}(i,m_t,r), \end{aligned} \tag{21}$$

where E^{tx} and E^{rx} are the sending and receiving energy/bit, respectively. Parameters $\delta^{tc}(i, m_t, r)$ and $\delta^{rc}(i, m_t, r)$ are the numbers of sending and receiving control bits by the i^{th} node in the m_t^{th} transmission phase and r^{th} round, respectively. As well, $\delta^{td}(i, m_t, r)$ and $\delta^{rd}(i, m_t, r)$ are numbers of sending and receiving data bits, respectively. E^{ag} and $\delta^{ag}(i, m_t, r)$ are the aggregating energy/bit and number of aggregated bits, respectively. For the same network setting, the ratio of the average energy per packet of the L^{3}EACH to DIVE protocol is computed as

$$\gamma_{E} = \frac{E_{av}^{L}}{E_{av}^{D}}, \tag{22}$$

where E_{av}^{L} and E_{av}^{L} are the average energy per packet of the L^{3}EACH and DIVE protocols, respectively, and they are computed using Eq. (20).

To maintain the efficiency of the L^{3}EACH protocol, some notes should be considered during its implementation. Firstly, the time durations of rounds and transmission phases should be much longer than those of the setup and initialization phases, respectively, i.e., $\tau_r \gg \tau_s$ and $\tau_t \gg \tau_{in}$. Secondly, the number of transmission sub-phases, M_{sp}, should be optimized. Increasing the number of sub-phases may result in wasted network resources since many SeCH nodes may not fill the reserved time slots. Conversely, too few sub-phases could increase overheads due to extra control messages (i.e., *Hi, RTS,* and *CTS*). The optimization process should consider the expected length of the gathered data per round, which is related to the application type. Thirdly, the protocol requires strict synchronization between the nodes since it is

[7]The errors in clock synchronization between the nodes.

time-based. Clock drifts (i.e., in the transmitter or receiver) and time variances (i.e., in the channel) could impair performance. Thus, the guard time τ_i^g should be adequately designed, and its size should be related to the propagation delay of the links [21]. The i^{th} SiCH node can set the guard time as

$$\tau_i^g = \beta \frac{L_i}{c_i}, \quad 0 < \beta \leq 1, \tag{23}$$

where L_i and c_i are the cluster radius and the acoustic speed[8] in the region of the i^{th} SiCH node, respectively. Equation (23) indicates that long links require long guard time between consecutive slots. Increasing β reduces the collision; however, it reduces the channel utilization as well. What is challenging is how to set the β value that trade-offs between minimizing the collision and maximizing the channel utilization.

V. LOW-LATENCY LOW-ENERGY ADAPTIVE CLUSTERING HIERARCHY VERSION 2 (L^{3}EACH-V2) PROTOCOL

Like the L^{3}EACH protocol, the L^{3}EACH-V2 follows the same process flow and time diagram shown in Figs. 2 and 3. However, it modulates the preamble bits to embed the local IDs rather than inserting overhead bits. This results in a significant reduction in network latency and average energy consumption per packet. The protocol is inspired by the preamble modulation techniques used in cellular communication systems [33], [34], [35], [36]. Section V-A bellow introduces the L^{3}EACH-V2 protocol in more detail and evaluates its computational complexity.

A. THE PROTOCOL MODEL

Figure 4 shows how the protocol modulates and demodulates the preamble bits. Algorithm 2 shows the pseudocode of the implementation.

As shown in Fig. 4a, the source nodes modulate the preamble signals using orthogonal codes to ensure this process is invertible at the receiving side. Firstly, source nodes map the preamble signal (i.e., the original preamble), source ID, and destination ID to three orthogonal codes. Then, they superimpose the orthogonal codes to produce a new preamble (i.e., a modulated preamble) signal. Next, the modulated preamble is inserted in the packet's overhead before the transmission instead of the original one. Finally, The destination nodes demodulate the preamble signal to recover the IDs using four processes; correlating, time-shifting, subtracting, and comparing processes, as shown in Fig. 4b.

Many orthogonal codes were proposed in the literature; however, the Zadoff-Chu (ZC) code is commonly used in cellular communications due to its outstanding correlation features. The ZC code is complex discrete samples with constant amplitude. The ZC code associated with the i^{th} node

[8]The speed of the acoustic wave depends on the temperature and pressure of the seawater in the region of the cluster.

Algorithm 2 Pseudocode of the L^{3}EACH-V2 Protocol Shows the Preamble Modulation and Demodulation Processes

Input: Length of ZC sequence, l_{zc}, reference preamble, $s_1[n]$, and the threshold of cross-correlations, α_{th}.
Output: IDs of the source and destination.
/*1- Preamble Modulation Process:*/
Transmitting node ID $u_i = \{2, 3, \ldots, (1-l_{zc})\}$ and GCD $(u_i, l_{zc}) = 1$.
Destination node ID $u_j = \{2, 3, \ldots, (1 - l_{zc})\}$, GCD $(u_j, l_{zc}) = 1$ and $j \neq i$.
The i^{th} node generates ZC sequences using indexes $\{1, u_i, u_j\}$ and Eq. (24).
for Packet No. $= \{1, 2, \ldots\}$ **do**
    Modulated preamble: $x_{ji}[n] = s_{u_1}[n] + s_{u_i}[n] + s_{u_j}[n]$.
end for
/*2- Channel:*/
Generate random channel using Eq. (26)
/*3- Preamble Demodulation Process:*/
for Packet No. $= \{1, 2, \ldots\}$ **do**
    Received packet: $y_{ji}[n] = H_{ji}\, x_{ji}[n] + w_j[n]$.
    /*3.1- Synchronization Process:*/
    Cross-correlate: $C(s_1, y_{ji}, q)$, $q \in \{0, 1, 2, \ldots, l_{zc} - 1\}$.
    Optimize: $q^* = \text{argmax}_{q\in\{0,1,2,\ldots,l_{zc}-1\}} |C(s_{u_1}, y_{ji}, q)|$.
    Synchronize the received packet: $y_{ji}[n] \rightarrow y_{ji}[n, q^*]$.
    /*3.2- Detection of Destination ID:*/
    Subtract the preamble: $y'_{ji}[n, q^*] = y_{ji}[n, q^*] - \frac{1}{H} s_{u_1}[n]$.
    Cross-correlate: $C(s_{u_i}, y'_{ji}, q^*)$.
    if $C(s_j, y'_{ji}, q^*) \geq \alpha_{th}$ **then**
        /*3.3- Detection of Source ID:*/
        Subtract destination ID: $y''_{ji}[n, q^*] = y'_{ji}[n, q^*] - \frac{1}{H} s_{u_i}[n]$.
        for $u_i = 2 : l_{zc} - 1, i \neq j$ **do**
            Cross-correlate: $C(s_j, y''_{ji}, q^*)$.
        end for
        Optimize: $u_i^* = \text{argmax}_{u_i\in\{2,3,\ldots,l_{zc}-1\}} |C(s_{u_i}, y''_{ji}, q^*)|$,
    else
        Discard the received packet.
    end if
end for

is defined as [49]

$$s_{u_i}[n] = \exp\left(\frac{-\pi\sqrt{-1}\, u_i\, n\, (n + \varphi_1 + 2\varphi_2)}{l_{zc}}\right), \tag{24}$$

where n indicates the sample number, $n = \{0, 1, 2, \ldots, l_{zc} - 1\}$, and l_{zc} is the length of the ZC code. Symbol u_i is the index of the ZC code, where $u_i = \{1, 2, \ldots, l_{zc} - 1\}$. The ZC code must satisfy the condition of the greatest common divisor (GCD), i.e., GCD of $(u_i, l_{zc}) = 1$. Symbol φ_1 is a complex

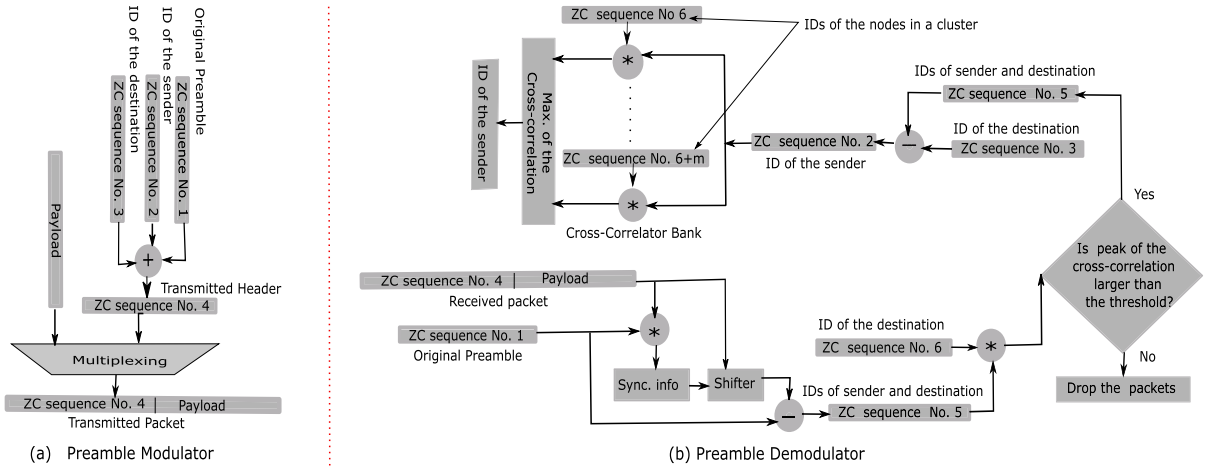

FIGURE 4. Block diagram of the L^{3}EACH-V2 shows the preamble modulation and demodulation processes. In this figure, operator {∗} indicates auto and cross-correlation processes.

number, and φ_2 is a real number equal to (l_{zc} mode 2). Correlation features of the ZC codes could be mentioned as follows: first, the autocorrelation between a ZC code with a shifted version of itself is zero. Second, the cross-correlation between two ZC codes is inversely proportional to the code length, i.e., $1/\sqrt{l_{zc}}$, given that the GCD of $(u_j - u_i, l_{zc}) = 1$, where $\{u_j, u_i\}$ are the ZC indexes associated with the j^{th} and the i^{th} nodes, respectively. The residual error of the cross-correlation process approaches zero when the code length is long enough.

Figure 4a shows the modulation process. Assuming the i^{th} sensor node sends packets to the j^{th} node, the i^{th} node modulates the preamble by superimposing the ZC codes of the original preamble, its local ID, and destination's local ID as

$$x_{ji}[n] = s_{u_i}[n] + s_{u_i}[n] + s_{u_1}[n],\ u_j \neq u_i, \tag{25}$$

where $\{s_{u_1}, s_{u_i}, s_{u_j}\}$ are ZC codes[9] of the original preamble, sender ID, and receiver ID, respectively. Due to channel impairments, the received sequence differs from the transmitted symbols, $x_{ji}[n]$. Let H_{ji} be the channel coefficient for the link between the i^{th} and the j^{th} nodes which is a complex quantity defined as

$$H_{ji} = h_{ji} \exp\left(2\pi \sqrt{-1} f_c\, q^*\right), \tag{26}$$

where h_{ji} is the DC channel gain which is the inverse of the path loss defined in Eq. (1), i.e., $h_{ji} = A_{ji}^{-1}$. f_c is the frequency carrier, and q^* is the quantized channel delay. The received sequence, $y_{ji}[n + q^*]$, is often captured by the following expression

$$y_{ji}[n + q^*] = H_{ji}\, x_{ji}[n] + IN_j[n], \tag{27}$$

where $IN_j[n]$ is the interference and noise time sequence of the j^{th} receiver, and it is obtained by taking inverse-Fourier transform of Eq. (4), then sampling with a sampling rate that satisfies the Nyquist rate.

As shown in Fig. 4b, the demodulation process is based on the autocorrelation characteristics of ZC's codes **(lines No. 17-20 in Algo. 2)**. The j^{th} node begins the demodulation process by synchronizing itself with the received packets **(lines No. 13-16 in Algo. 2)**. Synchronization is accomplished when the beginning of the preamble signal is determined. The synchronization is achieved by correlating the received packet and a delayed version of the preamble. The correlation coefficient, $C(s_1, y_{ji}, q)$, between the received signal $y_{ji}[n + q^*]$ and the delayed version of the preamble $s_{u_1}[n+q]$ is computed as

$$\begin{aligned} &C(s_{u_1}, y_{ji}, q) \\ &\quad = \sum_{u=1}^{l_{zc}} s^{\star}_{u_1}[u+q]\, y_{ji}[u+q^*] \\ &\quad = h_{ji}\left(\sum_{u=1}^{l_{zc}} s^{\star}_{u_1}[u+q]\, s_{u_1}[u+q^*] + \frac{2}{\sqrt{l_{zc}}}\right) + IN_j[n], \end{aligned} \tag{28}$$

where $s^{\star}_{u_1}[u+q]$ is the complex conjugate of the delayed preamble sequence $s_{u_1}[u+q]$, $u \in \{1, 2, \ldots, l_{zc}\}$, and q is the quantized delay value. Equation (28) contains three terms: the first represents the received preamble power at delay time $u+q$, the second represents the residual power resulting from cross-correlations between the preamble and the IDs, and the last term represents interference and noise at the j^{th} node. The value of the correlation coefficient is zero when $s_{u_1}[n+q]$ is orthogonal to all the sequences of $y_{ji}[n+q^*]$. However, the correlation coefficient reaches its peak when $s_{u_1}[n+q]$ is non-orthogonal and synchronized with one sequence of $y_{ji}[n+q^*]$. In order to synchronize with incoming packets, the j^{th} node checks the correlation coefficient versus q values and picks the q value that yields the largest correlation coefficient.

[9]In this paper, we assign an index of 1 to the original preamble signal to simplify the illustration. However, any index could be used in the implementation.

This optimization problem can be formulated as

$$q^* = \underset{q\in\{0,1,2,\ldots,l_{zc}-1\}}{\operatorname{argmax}} |C(s_{u_1}, y_{ji}, q)|. \tag{29}$$

The peak value of the correlation coefficient is

$$C(s_{u_1}, y_{ji}, q)|_{q=q^*} = h_{ji}\left(\sum_{u=1}^{l_{zc}} |s_{u_1}[u+q^*]|^2 + \frac{2}{\sqrt{l_{zc}}}\right) + IN_j[n]. \tag{30}$$

After synchronization is achieved, the j^{th} node first recovers the destination ID before the source ID to simplify the ID detection. In order to recover the destination ID, the node eliminates the preamble signal from the received packets as follows:

$$y_{ji_1}[n+q^*] = y_{ji}[n+q^*] - \frac{1}{h_{ji}} s_{u_1}[n+q^*]. \tag{31}$$

The DC gain, h_{ji}, is roughly estimated by dividing the amplitude of Eq. (25) over the amplitude of Eq. (27). The node cross-correlates the resulting sequence, $y_{ji_1}[u+q^*]$, with its ID sequence, $s_{u_j}[n+q^*]$, and compares the results with a threshold α_{th} as follows:

$$C(s_{u_j}, y_{ji_1}, q^*) = \frac{1}{h_{ji}} \sum_{u=1}^{l_{zc}} s^{\star}_{u_j}[u+q^*]\, y_{ji_1}[u+q^*] \geq \alpha_{th}. \tag{32}$$

If the correlation coefficient value is smaller than the threshold, α_{th}, the node discards the packet assuming it is sent to another node. Otherwise, the node accepts the packet and detects the sender ID. In order to detect the sender's ID, the node first eliminates the destination ID from the received packets as follows

$$y_{ji_2}[n+q^*] = y_{ji_1}[n+q^*] - \frac{1}{h_{ji}} s_{u_j}[n+q^*]. \tag{33}$$

The node then cross-correlates the resulting signal, $y_{ji_2}[n+q^*]$, with all possible ID sequences in the cluster. Then, it picks the ID that maximizes the correlation coefficient **(lines No. 21-29 in Algo. 2)**. The maximum value of the correlation coefficient is formulated as

$$u_i^* = \underset{u_i\in\{2,3,\ldots,l_{zc}-1\},\, u_i\neq u_j}{\operatorname{argmax}} |C(s_{u_i}, y_{ji_2}, q^*)|, \tag{34}$$

where $C(s_{u_i}, y_{ji_2}, q^*)$ is computed as

$$C(s_{u_i}, y_{ji_2}, q^*) = \frac{1}{h_{ji}} \sum_{u=1}^{l_{zc}} s^{\star}_{u_i}[u+q^*]\, y_{ji_2}[n+q^*]. \tag{35}$$

The optimization process in Eq. (34) is simple since it considers only ZC codes of the cluster and not the whole network. The modulation and demodulation steps are summarized in Algo. 2.

Careful selection of the threshold value, α_{th}, is crucial for the performance of Algo. 2. Algorithm 3 provides a criterion for computing the optimal α_{th} values based on $SINR_{ji}$, as shown in the figure. The algorithm evaluates the error probability of ID detection for different threshold values in the range $\alpha_{th} > 1$ for a specific $SINR_{ji}$ value. For each value of α_{th}, it iterates over thousands of packets using Algo. 2, computes the error probability, and determines the α^*_{th} value that minimizes the error. This process is repeated for each new $SINR_{ji}$ value. Algorithm 3 is executed only once before network deployment, and the results are stored in a lookup table in the memory of sensor nodes. In the field, the nodes estimate the $SINR_{ji}$ values and dynamically adjust the threshold value to α^*_{th} in real-time, thereby enhancing the performance of Algo. 2.

Algorithm 3 Pseudocode of Optimizing the Threshold, α^*_{th}, Versus the $SINR_{ji}$ for Algo. 2

1: **Input:** Algorithm (2).
2: **Output:** Optimal values of the threshold α^*_{th} versus $SINR_{ji}$ values.
3: **for** $SINR_{ji}$ values in range … **do**
4: **for** α_{th} values in range … **do**
5: **for** Iteration No. = {1, 2, …, Φ} **do**
6: Call Algo. (2)
7: **if** Detected IDs = transmitted IDs **then**
8: No. correct IDs = No. correct IDs + 1,
9: **end if**
10: **end for**
11: Error probability of IDs at α_{th} = (Φ − No. correct IDs) / Φ
12: **end for**
13: Select α^*_{th} value that gives the minimum error probability
14: **end for**
15: List optimal values α^*_{th} versus $SINR_{ji}$ values

When implementing the L^{3}EACH-V2 protocol, several points should be considered. First, the ZC code's length should be at least $l_{zc} \geq J_i + 1 + \zeta$ to assign a unique code to each node in the i^{th} cluster and allocate ζ codes for broadcast and multicast control messages, e.g., *Hi messages*. Increasing the code length reduces residual errors in the cross-correlation process, thereby improving the ID detection's error probability. $SINR_{ji}$ values can be estimated using the preamble signal to select the optimal threshold value, i.e., α^*_{th}. When the $SINR_{ji}$ is low, errors in ID detection are expected, so high transmission power is recommended. The Doppler shift's effect on the modulated preambles can be neglected in quasi-stationary links but must be considered in mobile links (e.g., sensor nodes attached to underwater automotive vehicles) [37].

B. COMPUTATIONAL COMPLEXITY

The L^{3}EACH-V2 protocol outperforms the L^{3}EACH, but it requires additional DSP due to the modulation and demodulation of preamble signals. This subsection aims to quantify the extra computational complexity needed for the L^{3}EACH-V2 over the L^{3}EACH protocol.

As illustrated in Fig. 4 and Algo. 2, the correlation process dominates the other processes. Given the short initialization phases, we can calculate the computational complexity by only considering the correlation process in the transmission phases. Since the correlation process of synchronization is performed in both L^{3}EACH and L^{3}EACH-V2 protocols, we focus only on the correlation process for ID demodulations. The correlation process for the destination ID is executed only once per packet, so we can ignore it. However, the correlation process for the source ID is performed multiple times per packet, and thus we must consider it.

Assuming the cluster network is composed of one mothership, I sink nodes, and J SeCH nodes. Let J_i be the number of SeCH nodes per the i^{th} SiCH node, where $i = \{1, 2, \ldots, I\}$. K_{j_i} is the number of SeCM nodes per the j_i^{th} SeCH node in the cluster of the i^{th} SiCH node, where $j_i = \{1, 2, \ldots, J_i\}$. Let lengths of the ZC codes be l_{zc}, the election probability as P, the $SINR_{ji}$ threshold as $SINR_{th}$, and the total received packets be $Q(SINR_{th})$. The computation complexity, C_x, of L^{3}EACH-V2 is computed as

$$\begin{aligned} C_x(SINR_{th}, l_{zc}, P, J, K_{j_i}) \\ = \mathcal{O}\Bigg(\sum_{i=1}^{I} \Bigg[Q_i(SINR_{th}) \\ \times\ C_i(l_{zc})\ \times\ J_i(P, J) + \sum_{j_i=1}^{J_i} Q_{j_i}(SINR_{th}) \\ \times\ C_{j_i}(l_{zc})\, K_{j_i}(P, J) \Bigg] \Bigg), \end{aligned} \tag{36}$$

where $\mathcal{O}$ is the notation of the function complexity. In this equation, the first summation is performed over the number of SiCH nodes per the mothership, i.e., I. The second summation is performed over the number of SeCH nodes per i^{th} SiCH node, i.e., J_i. The first term presents the demodulation complexity of SiCH nodes, where $Q_i(SINR_{th})$ is the number of received packets by the i^{th} SiCH node. $C_i(l_{zc})$ is the DSP associated with the processor of the i^{th} SiCH node. The second term presents the demodulation complexity processed by SeCH nodes, where $Q_{j_i}(SINR_{th})$ is the number of received packets by the $j_i{}^{th}$ SeCH node, and $C_{j_i}(l_{zc})$ is the DSP associated with the processor of the $j_i{}^{th}$ SeCH node.

Equation (36) could be approximated under the following assumptions. Assume $J_i(P, J)$ and $K_{j_i}(P, J)$ equal to the average values

$$J_i(P, J) = \frac{P \times J}{I}, \quad K_{j_i}(P, J) = \frac{1 - P}{P}. \tag{37}$$

Also, let $C_i(l_{zc})$ and $C_{j_i}(l_{zc})$ be equal to the average values: $C_i(l_{zc}) = \hat{C}_i(l_{zc})$ and $C_{j_i}(l_{zc}) = \hat{C}_J(l_{zc})$. Assuming all received packets by SeCH nodes are forwarded to SiCH nodes and received successfully, i.e., $\sum_{j_i=1}^{J_i} Q_{j_i}(SINR_{th}) = Q_i(SINR_{th})$. Also, let $\sum_{i=1}^{I} Q_i(SINR_{th}) = Q_i(SINR_{th})$. The autocorrelation DSP, $\hat{C}_J(l_{zc})$ and $\hat{C}_i(l_{zc})$, equal to an average value $C(\hat{l}_{zc})$, i.e., $\hat{C}_J(l_{zc}) \approx \hat{C}_i(l_{zc}) \approx C(\hat{l}_{zc})$, where $\hat{l}_{zc}$ is the average length of the ZC codes used in the entire network. So, Eq. (36) can be approximated as

$$\begin{aligned} C_x(SINR_{th}, l_{zc}, P, J, I) \\ \approx \mathcal{O}\Bigg(Q_i(SINR_{th}) \Bigg[\frac{P \times J}{I} + \frac{1 - P}{P} \Bigg] C(\hat{l}_{zc}) \Bigg). \end{aligned} \tag{38}$$

In this equation, the computational complexity is a linear function of the number of SeCH nodes, J, the number of received packets, $Q_K(SINR_{th})$, and the autocorrelation DSP $C(\hat{l}_{zc})$. However, the computational complexity is also a non-linear function of the election probability, P, and the number of sinks, I. In the next section, we investigate this equation numerically and show the influence of P and I on the computational complexity of the L^{3}EACH-V2 protocol.

TABLE 2. Simulation parameters of the numerical results [50], [51], [52].

Network Parameters	
Network dimension	$10 \times 10 \times 10$ Km^3
Number of nodes	$[125, 216, \ldots, 2197]$ nodes
Network topology	3-D Square Grids
Mobility model	3-D Gaussian
Variance in x, y and z axes	$\{0.05, 0.05, 0.005\}$ of the grid length
Transceiver Parameters	
Bit Rate	80 bps
Payload	32 bits
Overheads (preamble+packet type+CRC)	$14 + 2 + 8 = 24$ bits
Maximum transmitted power	6.3492×10^{-12} Watt
Power threshold of the receiving packet	6.3492×10^{-20} Watt
Energy of bit aggregation	5×10^{-9} Joule/bit
Spreading factor of the acoustic beam	1.5
Acoustic carrier	12 kHz
Bandwidth	4 kHz

VI. NUMERICAL RESULTS

This section compares the performance of the L^{3}EACH and L^{3}EACH-V2 protocols to the DIVE protocol. We selected the DIVE protocol because it is tested by CMRE-NATO [31], [32]. To ensure a fair comparison in the numerical results, we assume that the DIVE, L^{3}EACH, and L^{3}EACH-V2 protocols are identical in terms of resource allocation (i.e., node IDs, CDMA codes, and time slots) and packet routing through multi-hop transmissions, as already shown in Figs. 1, 2, and 3. However, they are different in how they assign the IDs. The DIVE protocol assigns unique IDs on a wide scale, the L^{3}EACH protocol assigns unique IDs on a local scale, and the L^{3}EACH-V2 protocol embeds local IDs in the preamble signals.

A. SIMULATION SETUP AND PARAMETERS

The simulation parameters for the considered scenarios are presented in Table 2, which are based on the commercial JANUS standard and a commercial micro-modem [50], [52].

We consider a 3-D grid topology with a size of $10 \times 10 \times 10$ km^3 and a varying number of sensor nodes, i.e., $[125, 216, \ldots, 2197]$. The numbers of nodes are chosen to ensure uniform distribution with equal inter-distancing in the grid. The inter-distance between the nodes is proportional to the number of nodes and ranges between 0.9 and 2.8 km. Although the sensor nodes are assumed to be stationary, they are displaced around their centers in the X, Y, and Z directions due to waves and currents in the seawater. The

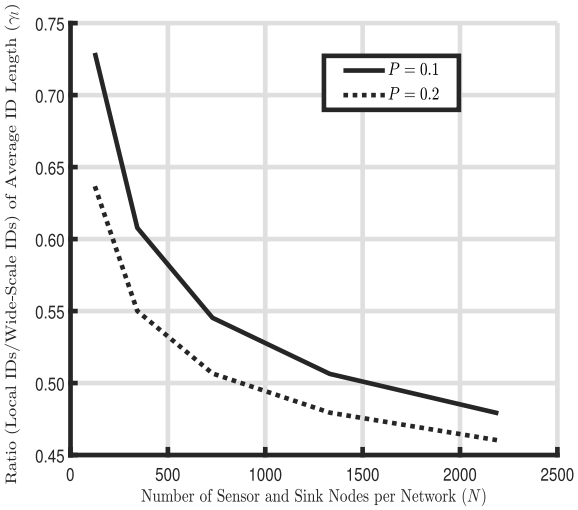

FIGURE 5. The ratio between average ID lengths of L^{3}EACH and DIVE protocols (Eq. (14)).

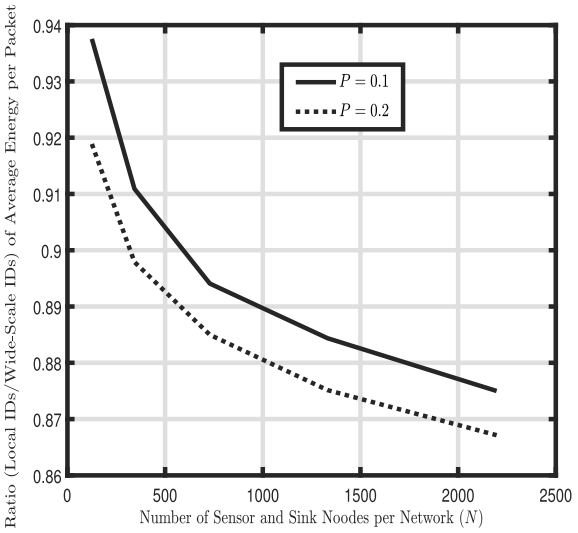

FIGURE 6. The ratio between average energies per packet of L^{3}EACH and DIVE protocols (Eq. (22)).

displacement is modeled using a 3-D Gaussian random variable, and the displacement variances for the X and Y axes are equal and more significant than that for the Z axis, as given in the table. Assuming short inter-distances between the nodes with slight mobility, we ignore the effects of Doppler shifting, multi-path fading, and beam-bending.

We assume a low bit rate of 80 bps, with an overhead of 24 bits and a payload of 32 bits per packet. The overhead bits consist of 14 for the preamble signal, 2 for packet type, and 8 for cyclic redundancy check (CRC) [50], [51]. They adapt their transmission power based on the channel conditions, with a maximum power of 6.3492×10^{-12} Watt. We consider packets to be received correctly if their received powers exceed 6.3492×10^{-20} Watt. The SeCH nodes consume 5×10^{-9} Joule per bit for data aggregation [51], [52].

B. THE L^{3}EACH PROTOCOL: AVERAGE LENGTH OF THE IDs AND AVERAGE ENERGY PER PACKET

Figure 5 shows the average ID length ratio between the L^{3}EACH and DIVE protocols, computed using Algo. 1 and Eqs. (12)-(14) for two different election probability values, $P = \{0.1, 0.2\}$. The results show that the L^{3}EACH protocol assigns significantly shorter IDs compared to the DIVE protocol. As the number of nodes increases, the L^{3}EACH protocol assigns relatively much shorter IDs, which can be attributed to the increase in the number of clusters per network. This, in turn, reduces the number of nodes per cluster, leading to a further reduction in the ID length. For networks with 125 and 2300 nodes, the average length ratio between the L^{3}EACH and DIVE protocols is 73% and 48%, respectively, when $P = 0.1$. Consequently, the L^{3}EACH protocol reduces ID lengths by 27% and 52%, respectively. The reduction in ID length is more significant when $P = 0.2$ than $P = 0.1$, with the L^{3}EACH protocol shrinking ID lengths by 36% and 55% for networks with 125 and 2300 nodes, respectively.

Figure 6 shows the ratio between the average energy per packet for the L^{3}EACH protocol and the DIVE protocol. The ratio is computed using Algo. 1 and Eqs. (20)-(22), and it is depicted for two values of the election probability, $P = \{0.1, 0.2\}$. The L^{3}EACH protocol consumes less energy per packet than the DIVE protocol due to its use of shorter IDs and, i.e., fewer overhead bits. As shown in the figure, the energy savings become more significant as the number of nodes increases. Increasing the election probability increases the number of clusters per network which reduces the number of nodes per cluster and decreases the average energy per packet. For $P = 0.1$, the ratio of the average energy is 94% and 88% when $N = \{125, 2300\}$, respectively. Thus, the L^{3}EACH protocol reduces the average energy per packet by 6% and 12%, respectively. When $P = 0.2$, the energy savings are even more significant, with the L^{3}EACH protocol saving 8% and 13% of the energy, respectively.

C. THE L^{3}EACH-V2 PROTOCOL: PREAMBLE MODULATION AND DEMODULATION

Figure 7 illustrates the missed and false ID detection probabilities of the L^{3}EACH-V2 protocol versus the $SINR_{ji}$ in dB, with $\alpha_{th} = 2$. Missed ID detection occurs when the intended destinations dismiss received packets, while false ID detection occurs when the unintended destinations accept received packets. The figure shows that as $SINR_{ji}$ increases, the probability of missed packets decreases, while the probability of false packets increases. For instance, at $SINR_{ji} = -20$ dB, the probability of missed and false packets are 1 and 0.025, respectively. On the other hand, at $SINR_{ji} = 20$ dB, the probability of missed and false packets are 0.125 and 0.075, respectively.

Figure 8 presents the error probability of the L^{3}EACH-V2 protocol and the optimal threshold values, α^*_{th}, on the left and right Y-axes, respectively. The error probability is the joint probability of missed and false ID detection shown in Fig. 7. The figure evaluates the error probability against $SINR_{ji}$ in dB at three threshold values, i.e., $\alpha_{th} = \{1.5, 2, 2.5\}$. The results reveal that the error probability performance varies

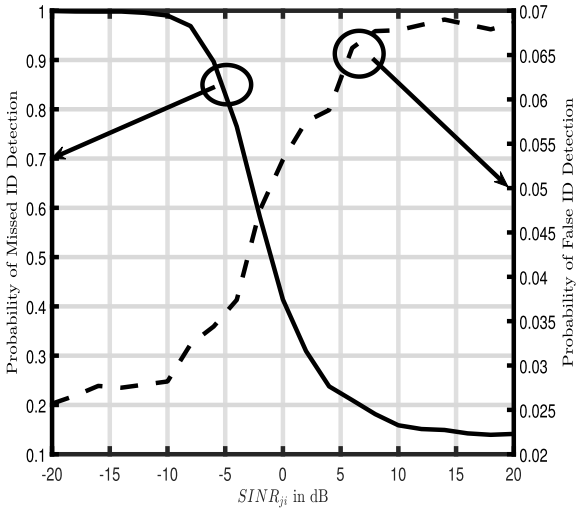

FIGURE 7. The probabilities of missed and false ID detections for L^{3}EACH-V2 protocol versus the $SINR_{ji}$ in dB, assuming $\alpha_{th} = 2$.

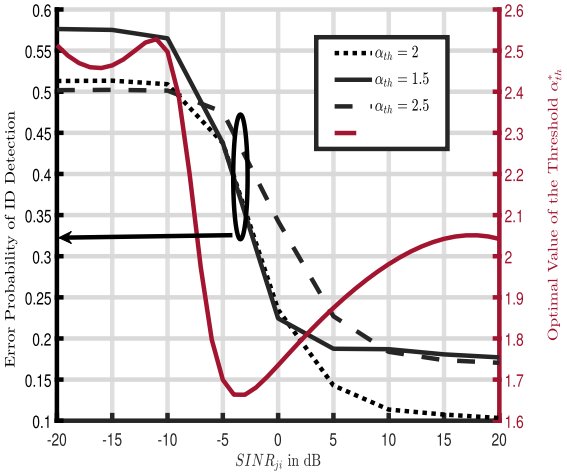

FIGURE 8. On the left Y-axis, the error probability of missed and false packets versus the $SINR_{ji}$ at different threshold values. While on the right Y-axis, the optimal threshold versus the $SINR_{ji}$.

significantly with the threshold value. For instance, $\alpha_{th} = 2.5$ achieves better performance in the low $SINR_{ji}$ range, i.e., $[-20, -10]$ dB, while $\alpha_{th} = 2$ achieves better performance in the high $SINR_{ji}$ range, i.e., [10, 20] dB. On the right Y-axis of the figure, the optimal threshold α^*_{th} is computed versus $SINR_{ji}$ using Algo. 3. The optimal threshold depends on the channel quality (i.e., $SINR_{ji}$) and changes non-linearly. As shown, the optimal threshold values are $\alpha^*_{th} = 2.5$ and 2.05 at poor and good channel conditions, $SINR_{ji} = -15$ and 15 dB, respectively.

D. THE L^{3}EACH-V2 PROTOCOL: SUCCESSFULLY RECEIVED PACKETS AND AVERAGE ENERGY PER PACKETS

Figure 9 shows the number of successfully received packets versus the number of nodes per network for the L^{3}EACH-V2, L^{3}EACH, and DIVE protocols. The figure indicates an increase in the number of successfully received packets with the increase in the number of nodes per network for all protocols. However, the increase is more significant for the L^{3}EACH-V2 protocol and less significant for the DIVE protocol. The L^{3}EACH-V2 protocol creates shorter packet lengths by embedding the source and destination IDs in the preamble signal, rather than using additional overhead bits. This reduces the duration of L^{3}EACH-V2 packets, allowing the channel to serve a larger number of nodes without experiencing packet collisions or channel congestion. Hence, adding more nodes can significantly increase the number of delivered packets in L3EACH protocols, while this may not be the case for other protocols. For instance, when the number of sensor nodes per network is 1000 nodes, the L^{3}EACH-V2, L^{3}EACH, and DIVE protocols successfully process 3.75, 3.3, and 3.2 million packets, respectively. This implies that the L^{3}EACH-V2 protocol improves the network throughput by 12% and 16% compared to the L^{3}EACH and DIVE protocols, respectively.

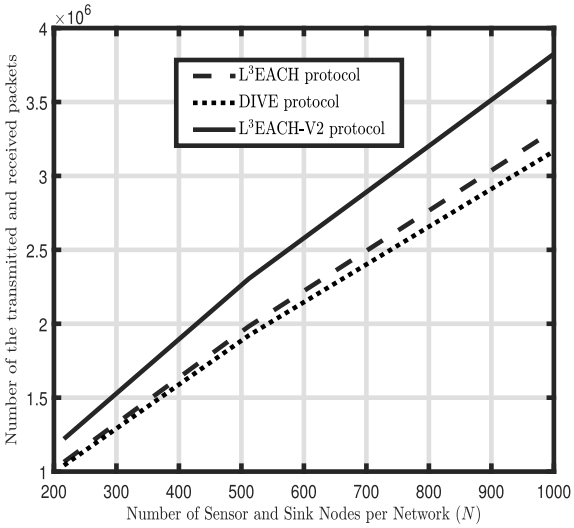

FIGURE 9. Number of transmitted and received packets for L^{3}EACH-V2, L^{3}EACH, and DIVE protocols versus the number of nodes per network.

Figure 10 illustrates the average energy consumption per packet in Joules versus the number of nodes per network for the L^{3}EACH-V2, L^{3}EACH, and DIVE protocols. The figure only takes into account the energy consumed in packet transmission and reception, which constitutes a relatively large portion of the total energy consumption compared to that of the processors (i.e., DSP energy).[10] It is observed that the DIVE protocol has the highest average energy consumption per packet, while the L^{3}EACH-V2 protocol has the lowest. Moreover, the performance of the L^{3}EACH-V2 protocol does not vary significantly with the number of nodes in the network. However, the performance of the DIVE protocol shows significant variability with changes in the number of nodes per network due to its longer packets that lead to a higher number of packet collisions as depicted in Fig. 5. Adding more nodes to the network results in more channel congestion, leading to more packet collisions, which increases the average energy consumption per packet and, consequently, decreases the energy efficiency of the network.

[10]Computing DSP energy is a challenging task since it depends on tens of unknown hardware parameters.

TABLE 3. Comparison between the DIVE, L^{3}EACH, and L^{3}EACH-V2 protocols.

Features	DIVE	L^{3}EACH	L^{3}EACH-V2
Average Packet Length	Long	Medium	Short
Average Energy per Packet	High	Medium	Low
Communication Latency	Large	Medium	Small
Successful Delivery Rate	Low	Medium	High
Completed Frame Work?	No	Yes	Yes
Required $SINR_{ji}$ Values	Small	Small	Large
Computational Complexity	Medium	Medium	High
Network Topology	Any Topology	Clustering Topology Only	Clustering Topology Only
Type of Applications	Energy-Limited Applications	Energy-Limited Applications	Bandwidth-Limited Applications

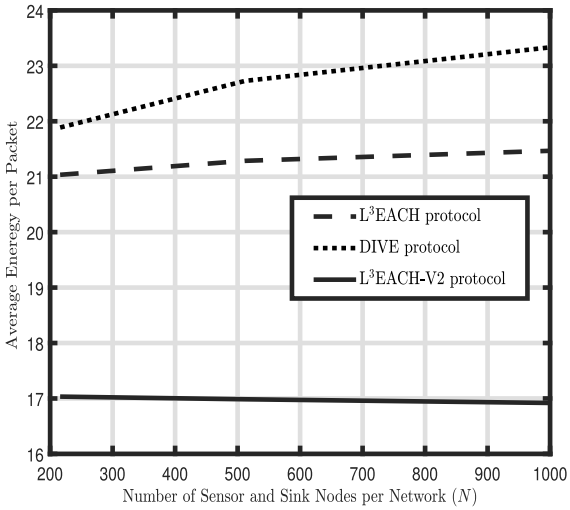

FIGURE 10. Average energy per packet for L^{3}EACH-V2, L^{3}EACH, and DIVE protocols versus the number of nodes per network.

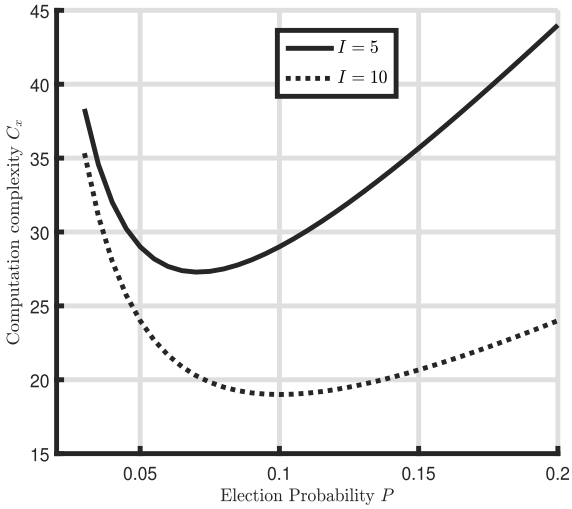

FIGURE 11. Computational complexity of the L^{3}EACH-V2 protocol versus the election probability of SeCH nodes.

For instance, for a network with 600 nodes, the L^{3}EACH-V2, L^{3}EACH, and DIVE protocols consume 17, 21.3, and 23 Joules, respectively.

E. THE L^{3}EACH-V2 PROTOCOL: COMPUTATIONAL COMPLEXITY

In Fig. 11, the computational complexity of the L^{3}EACH-V2 protocol is depicted as a function of the election probability, P, while assuming a network of $N = 10^3$ nodes and $I = \{5, 10\}$ sink nodes. The complexity is calculated using Eq. (38) and normalized by the number of received packets and correlation complexity. The curve of the computational complexity takes the shape of a U-curve, where it decreases initially and then increases as P increases. The reason behind this behavior is that increasing P results in more SeCH nodes and fewer SeCM nodes per cluster, leading to fewer correlation processes per SeCH node and, thus, lower computational complexity. However, the increase in P also leads to more SeCH nodes per SiCH, resulting in more correlation processes per SiCH node and, therefore, more computations. For example, when the number of sink nodes is 5, increasing P by a factor of four (from $P = 0.05$ to $P = 0.2$) results in a 1.5 times increase in computational complexity (from $C_x = 30$ to $C_x = 43$). However, when the number of sink nodes is 10, increasing P by a factor of four (from $P = 0.025$ to $P = 0.1$) leads to a 0.5 times reduction in computational complexity (from $C_x = 35$ to $C_x = 18$).

F. A SUMMARY OF THE NUMERICAL DISCUSSIONS

Table 3 summarizes the outcomes of the numerical analysis conducted for the DIVE, L^{3}EACH, and L^{3}EACH-V2 protocols. Among the three, the L^{3}EACH-V2 protocol offers several advantages, including a longer network lifetime, lower communication latency (i.e., end-to-end delay), and higher communication reliability (i.e., successful delivery rate). The protocol is suitable for real-time bandwidth-limited applications, such as underwater video surveillance and instant-message systems. It can only be implemented in cluster networks and requires high $SINR_{ji}$ values and additional DSP. On the other hand, due to its wide-scale ID assignment, the DIVE protocol has the poorest performance in terms of network lifetime, communication latency, and communication reliability. The protocol is appropriate for non-real-time applications, such as underwater telemetry and remote-control systems. It can be employed in any network topology, and it does not require high $SINR_{ji}$ values or additional DSP.

VII. CONCLUSION AND FUTURE WORKS

This paper presents two innovative cluster network protocols, L^{3}EACH and L^{3}EACH-V2, which improve communication latency and energy efficiency of UACNs compared to the DIVE protocol. These protocols are suitable for latency and energy-sensitive applications such as video surveillance and instant-message systems. However, they have limitations, such as being only suitable for cluster topologies, requiring additional DSP, and optimal performance requiring high

$SINR_{ji}$ values. In addition, clock drift or variations in propagation delays may also impair performance.

There are several future research opportunities in this area. One such opportunity is to explore the incorporation of additional information into preamble signals. Another is to combine the L^{3}EACH and L^{3}EACH-V2 protocols and select them based on channel conditions. Finally, it is also necessary to analyze the protocols' performance with new physical-layer technologies, study the impact of clock drift, evaluate the L^{3}EACH-V2 protocol under the Doppler effect, and implement the protocols in the laboratory using the JANUS standard and the Micro-Modem device.

ACKNOWLEDGMENT

The authors would like to thank Verdier Assoume and Surabhi Karandikar for the technical discussion about implementing the JANUS standard and datasheet of the WHOI micro-modems. In addition, they would also like to thank Nadir Adam, Tau Rasethuntsa, and Surabhi Karandikar for proof-reading the article.

REFERENCES

[1] L. Brekhovskikh and Y. Lysanov, *Fundamentals of Ocean Acoustics* (Springer Series in Electronics and Photonics). Berlin, Germany: Springer, 1982.

[2] M. Jouhari, K. Ibrahimi, H. Tembine, and J. Ben-Othman, "Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and internet of underwater things," *IEEE Access*, vol. 7, pp. 96879–96899, 2019.

[3] J. Ling, H. He, J. Li, W. Roberts, and P. Stoica, "Underwater acoustic communications," *J. Acoustic Soc. Amer.*, vol. 128, no. 5, pp. 2898–2909, Nov. 2010.

[4] A. S. Ghazy, H. S. Khallaf, S. Hranilovic, and M. Khalighi, "Undersea ice diffusing optical communications," *IEEE Access*, vol. 9, pp. 159652–159671, 2021, doi: 10.1109/ACCESS.2021.3131276.

[5] A. S. Ghazy, S. Hranilovic, and M. Khalighi, "Angular MIMO for underwater wireless optical communications: Link modeling and tracking," *IEEE J. Ocean. Eng.*, vol. 46, no. 4, pp. 1391–1407, Oct. 2021, doi: 10.1109/JOE.2021.3055477.

[6] A. S. Ghazy, S. Hranilovic, and M. Khalighi, "Angular MIMO for underwater wireless optical communications: Channel modelling and capacity," in *Proc. 16th Can. Workshop Inf. Theory (CWIT)*, Jun. 2019, pp. 1–6, doi: 10.1109/CWIT.2019.8929907.

[7] A. S. Ghazy. (Jan. 1970). *Reliable High-Speed Short-Range Underwater Wireless Optical Communication Systems*. Handle Proxy. [Online]. Available: http://hdl.handle.net/11375/27263

[8] M. Stojanovic, J. G. Proarkis, J. A. Rice, and M. D. Green, "Spread spectrum underwater acoustic telemetry," in *Proc. IEEE Ocean. Eng. Soc. OCEANS Conf.*, Sep. 1998, pp. 650–654.

[9] K. O. McCoy, B. Tomas, and G. Zappa, "JANUS: The genesis, propagation and use of an underwater standard," in *Proc. 10th Eur. Conf. Underwater Acoust. (ECUA)*, Jul. 2010.

[10] *Digital Underwater Signaling Standard for Network Node Discovery and Interoperability*, Edition A, Version 1, NATO Standardization Office, NATO STANDARD, Brussels, Belgium, 2017, doi: 10.25607/OBP-813.

[11] G. Han, J. Jiang, N. Bao, L. Wan, and M. Guizani, "Routing protocols for underwater wireless sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 72–78, Nov. 2015.

[12] N. Li, J.-F. Martínez, J. M. Chaus, and M. Eckert, "A survey on underwater acoustic sensor network routing protocols," *Sensors*, vol. 16, no. 3, p. 414, Mar. 2016.

[13] M. Tariq, M. S. A. Latiff, M. Ayaz, and M. Z. Abbas, "Beacon-based routing protocols for underwater acoustic sensor networks," *Int. J. Commun. Syst.*, vol. 30, no. 18, Dec. 2017, Art. no. e3375.

[14] A. Khan, I. Ali, A. Ghani, N. Khan, M. Alsaqer, A. Rahman, and H. Mahmood, "Routing protocols for underwater wireless sensor networks: Taxonomy, research challenges, routing strategies and future directions," *Sensors*, vol. 18, no. 5, p. 1619, May 2018.

[15] J. Luo, Y. Chen, M. Wu, and Y. Yang, "A survey of routing protocols for underwater wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 137–160, 1st Quart., 2021.

[16] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, p. 3443, Oct. 2007.

[17] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[18] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, p. 10.

[19] K. Sathish, R. C. Venkata, R. Anbazhagan, and G. Pau, "Review of localization and clustering in USV and AUV for underwater wireless sensor networks," *Telecom*, vol. 4, no. 1, pp. 43–64, Jan. 2023, doi: 10.3390/telecom4010004.

[20] J. G. Proakis, E. M. Sozer, J. A. Rice, and M. Stojanovic, "Shallow water acoustic networks," *IEEE Commun. Mag.*, vol. 39, no. 11, pp. 114–119, Nov. 2001, doi: 10.1109/35.965368.

[21] 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.

[22] A. Al Guqhaiman, O. Akanbi, A. Aljaedi, and C. E. Chow, "A survey on MAC protocol approaches for underwater wireless sensor networks," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3916–3932, Feb. 2021.

[23] S. Khisa and S. Moh, "Survey on recent advancements in energy-efficient routing protocols for underwater wireless sensor networks," *IEEE Access*, vol. 9, pp. 55045–55062, 2021.

[24] F. Salva-Garau and M. Stojanovic, "Multi-cluster protocol for ad hoc mobile underwater acoustic networks," in *Proc. Oceans Celebrating Past Teaming Toward Future*, Sep. 2003, pp. 91–98.

[25] A.-R. Byun, K. Son, H.-S. Cho, and Y.-S. Jang, "Channel reserved MAC protocol for underwater sensor nerworks," in *Proc. 5th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2013, pp. 110–111.

[26] M. Ahmed, M. Salleh, and M. I. Channa, "CBE2R: Clustered-based energy efficient routing protocol for underwater wireless sensor network," *Int. J. Electron.*, vol. 105, no. 11, pp. 1916–1930, Nov. 2018.

[27] W. Khan, H. Wang, M. S. Anwar, M. Ayaz, S. Ahmad, and I. Ullah, "A multi-layer cluster based energy efficient routing scheme for UWSNs," *IEEE Access*, vol. 7, pp. 77398–77410, 2019.

[28] Y. Liu, H. Wang, L. Cai, X. Shen, and R. Zhao, "Fundamentals and advancements of topology discovery in underwater acoustic sensor networks: A review," *IEEE Sensors J.*, vol. 21, no. 19, pp. 21159–21174, Oct. 2021.

[29] R. Agrawal, M. Chitre, and A. Mahmood, "Design of an address assignment and resolution protocol for underwater networks," in *Proc. OCEANS Shanghai*, Apr. 2016, pp. 1–7.

[30] R. Zhao, Y. Liu, O. A. Dobre, H. Wang, and X. Shen, "An efficient topology discovery protocol with node ID assignment based on layered model for underwater acoustic networks," *Sensors*, vol. 20, no. 22, p. 6601, Nov. 2020.

[31] R. Petroccia, "A distributed ID assignment and topology discovery protocol for underwater acoustic networks," in *Proc. IEEE 3rd Underwater Commun. Netw. Conf. (UComms)*, Aug. 2016, pp. 1–5.

[32] R. Petroccia, "DIVE: A distributed ID assignment and topology discovery protocol for underwater acoustic networks," *Ad Hoc Netw.*, vol. 122, Nov. 2021, Art. no. 102610.

[33] X. Ji, J. Wang, M. Liu, Y. Yan, P. Yang, and Y. Liu, "Hitchhike: Riding control on preambles," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2014, pp. 2499–2507, doi: 10.1109/INFOCOM.2014.6848196.

[34] Y. J. Kim, D. G. Han, and Y. S. Cho, "A preamble sequence design techniqe for efficient beam ID detection in millimeter-wave cellular systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 10106–10111, Dec. 2016, doi: 10.1109/TVT.2016.2528263.

[35] X. Ji, J. Wang, M. Liu, Y. Yan, P. Yang, and Y. Liu, "Hitchhike: A preamble-based control plane for SNR-sensitive wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1239–1251, Feb. 2016, doi: 10.1109/TWC.2015.2487961.

[36] H. Rahbari and M. Krunz, "Exploiting frame preamble waveforms to support new physical-layer functions in OFDM-based 802.11 systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3775–3786, Jun. 2017, doi: 10.1109/TWC.2017.2688405.
[37] M. Asim, R. Pec, T. H. Im, and Y. S. Cho, "Cell search techniques for underwater acoustic cellular systems," *IEEE Access*, vol. 7, pp. 106019–106033, 2019, doi: 10.1109/ACCESS.2019.2932721.
[38] R. Pec, M. Khan, M. Asim, and Y. Cho, "Random access for underwater acoustic cellular systems," *Sensors*, vol. 18, no. 2, p. 432, Feb. 2018, doi: 10.3390/s18020432.
[39] Dreamstime. *Dreamstime*. Accessed: Nov. 21, 2022. [Online]. Available: https://www.dreamstime.com
[40] J. Huang and R. Diamant, "Adaptive modulation for long-range underwater acoustic communication," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6844–6857, Oct. 2020, doi: 10.1109/TWC.2020.3006230.
[41] A. Khan, N. Javaid, G. Latif, L. Jatta, A. Fatima, and W. Khan, "Cluster based and adaptive power controlled routing protocol for underwater wireless sensor networks," in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, Apr. 2018, pp. 1–6, doi: 10.1109/NCG.2018.8592955.
[42] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 84–89, Jan. 2009, doi: 10.1109/MCOM.2009.4752682.
[43] F. Qu, Z. Wang, L. Yang, and Z. Wu, "A journey toward modeling and resolving Doppler in underwater acoustic communications," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 49–55, Feb. 2016, doi: 10.1109/MCOM.2016.7402260.
[44] Y. Lysanov and L. Brekhpvskikh, *Fundamentals of Ocean Acoustics*. New York, NY, USA: Springer, 1982.
[45] R. F. W. Coates, *Underwater Acoustic Systems*. Hoboken, NJ, USA: Wiley, 1989.
[46] J. Proakis and M. Salehi, *Digital Communications*. New York, NY, USA: McGraw-Hill, 2008.
[47] M. W. Eltokhey, M. A. Khalighi, A. S. Ghazy, and S. Hranilovic, "Hybrid NOMA and ZF pre-coding transmission for multi-cell VLC networks," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 513–526, 2020.
[48] N. Parrish, L. Tracy, S. Roy, P. Arabshahi, and W. Fox, "System design considerations for undersea networks: Link and multiple access protocols," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1720–1730, Dec. 2008.
[49] S. Beyme and C. Leung, "Efficient computation of DFT of Zadoff–Chu sequences," *Electron. Lett.*, vol. 45, no. 9, Apr. 2009, Art. no. 461463.
[50] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, "The Janus underwater communications standard," in *Proc. Underwater Commun. Netw. (UComms)*, Sep. 2014, pp. 1–4, doi: 10.1109/UComms.2014.7017134.
[51] H. Yu, N. Yao, T. Wang, G. Li, Z. Gao, and G. Tan, "WDFAD-DBR: Weighting depth and forwarding area division DBR routing protocol for UASNs," *Ad Hoc Netw.*, vol. 37, pp. 256–282, Feb. 2016.
[52] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proc. IEEE OCEANS05 Conf*, Sep. 2005, pp. 1086–1092, doi: 10.1109/OCEANS.2005.1639901.

ABDALLAH S. GHAZY was born in Giza, Egypt, in 1983. He received the B.Sc. degree in electrical engineering from Al-Azhar University, Egypt, in 2007, the M.Sc. degree in electrical engineering from the Electrical and Computer Engineering School, Egypt–Japan University for Science and Technology, Alexandria, Egypt, in 2016, and the Ph.D. degree in electrical engineering from the Electrical and Computer Engineering School, McMaster University, ON, Canada, in 2021. From 2008 to 2017, he was with Unified Communication Systems and Networks (UCSN), Telecom-Egypt, and AVAYA Inc., Cairo, Egypt. In 2012, he joined the Electrical Engineering School, Al-Azhar University. He is on leave from Al-Azhar University. He is currently a Ph.D. Research Fellow, École de Technologie Supérieure School, University of Quebec, Montreal, QC, Canada. His main research interests include communication systems and networks, signal processing, and machine learning.

GEORGES KADDOUM (Senior Member, IEEE) received the bachelor's degree in electrical engineering from École Nationale Supérieure de Techniques Avancées (ENSTA Bretagne), Brest, France, the M.S. degree in telecommunications and signal processing (circuits, systems, and signal processing) from Université de Bretagne Occidentale and Telecom Bretagne (ENSTB), Brest, in 2005, and the Ph.D. degree (Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences (INSA), University of Toulouse, Toulouse, France, in 2009. He is currently an Associate Professor and the tier-two Canada Research Chair with École de Technologie Supérieure (ÉTS), Université du Québec,Montréal, QC, Canada, and a faculty fellow in the Cyber Security Systems and Applied AI Research Center, Lebanese American University, Beirut, Lebanon. Since 2010, he has been a scientific consultant of space and wireless telecommunications for several USA and Canadian companies. He has published more than 200 journal and conference papers and has two pending patents. His recent research interests include mobile communication systems, modulations, security, and space communication and navigation. He received the Best Papers Awards from the 2014 IEEE International Conference on Wireless and Mobile Computing, Networking, Communications (WIMOB), with three coauthors, and from the 2017 IEEE International Symposium on Personal Indoor and Mobile Radio Communications, with four coauthors. In 2014, he was awarded the ÉTS Research Chair in physical-layer security for wireless networks. Moreover, he received the IEEE TRANSACTIONS ON COMMUNICATIONS Exemplary Reviewer Award, in 2015, 2017, and 2019. In addition, he received the Research Excellence Award from the Université du Québec, in 2018. In 2019, he received the Research Excellence Award from the ÉTS in recognition of his outstanding research outcomes. He is currently serving as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE COMMUNICATIONS LETTERS.

SATINDER SINGH (Member, IEEE) received the B.E. degree in wireless communications and digital signal processing from École de Technologie Supérieure. He has spent most of his career working in the wireless communications domain. He is most inspired when existing technology can be integrated with the latest innovation to create a solution that was deemed impossible to solve. He is currently the Director of Technology with Ultra, Intelligence and Communications, where he is focused on the development of wireless communication hardware and waveforms for tactical backhaul networks. He is also the Data Science Director of the Resilient Machine Learning Institute (REMI), an extension of Ultra. As the leader of REMI, his interests are aligned with AI/ML development for modern tactical networks, including situational and spectrum awareness, development of AI/ML-empowered waveforms, AI/ML-assisted Auto-PACE, and ease of use and end-user acceptance of machine learning algorithms.