IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 September 2023, accepted 30 September 2023, date of publication 10 October 2023,
date of current version 16 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323585

RESEARCH ARTICLE

Formal Verification of a MAC Protocol for Underwater Sensor Networks

N. SURESH KUMAR[1], G. SANTHOSH KUMAR[1], (Senior Member, IEEE), SHAILESH SIVAN[1], (Member, IEEE), AND A. SREEKUMAR[2], (Member, IEEE)
[1]Cyber Physical Systems Laboratory, Department of Computer Science, Cochin University of Science and Technology (CUSAT), Kochi, Kerala 682022, India
[2]Department of Computer Applications, Cochin University of Science and Technology (USAT), Kochi, Kerala 682022, India

Corresponding author: N. Suresh Kumar (cnsuresh2000@gmail.com)

ABSTRACT The use of Underwater Sensor Networks (UWSN) for underwater ocean applications such as seismic event detection, target detection, marine resource monitoring, and oil bed monitoring is growing. In contrast to conventional WSNs, these networks communicate via acoustic channels. Many communication protocols for UWSN have been proposed, including MAC layer protocols, time synchronization protocols, and routing protocols. Formal verification of these protocols is rarely investigated. In this paper, we propose two abstraction methods for UWSN that capture multi-channel models and variable propagation delay. These abstraction methods are used to create a validation model of the Time Delay Allocation MAC (TDA-MAC) protocol, which is used in UWSN. Formal verification of TDA-MAC is accomplished by performing a reachability analysis and the occurrence of design faults on certain marked states in the model. The verification results detect non-progress cycles of marked states in the event of a PING message loss. A modification to the existing protocol specification of TDA-MAC protocol is proposed. Formal verification on the refined validation model shows that the protocol is free from non-progress cycles and unreachable states. The proposed abstraction methods can be used to create formal models and perform formal verification of existing and emerging protocols used in UWSN.

INDEX TERMS Formal verification, PROMELA, SPIN model checker, TDA-MAC, underwater sensor networks (UWSN).

I. INTRODUCTION

Sensor and acoustic device advancements result in the deployment of sensors in underwater ocean beds. These sensors can be used to capture and record climatic parameters on ocean beds, marine resource detection, target tracking [9], and so on. These sensors communicate with devices on the ocean's surface via acoustic channels [10]. Unlike in terrestrial WSN, radio, and optical signals cannot be transmitted over long distances underwater due to absorption by ocean water [11]. UWSN communicates using acoustic signals. Acoustic signals propagate at around 1500 meters per second, which is very slow in comparison to radio signals. Hence propagation delay in UWSN is greater than in terrestrial networks. Furthermore, the speed of acoustic waves varies depending on ocean conditions such as temperature, salinity, and pressure depth. As a result, changes in ocean climatic conditions affect the propagation delay and velocity of sound waves in underwater acoustics. According to climatic studies [12], the acoustic speed of sound in the underwater ocean ranges from 1481 to 1510 meters per second. Another feature of the underwater sensor network is multi-path acoustic wave propagation. This acoustic wave propagation results in the same signal reaching the receiver multiple times due to curved paths, echo, and reflection on the ocean surface and bottom [7]. This is referred to as the multi-path channel condition. Therefore the operating conditions of UWSN differ from those of terrestrial networks.

Formal methods provide a systematic approach to checking plausible claims and ensuring consistent execution of a network protocol during the early design stages [19]. As UWSN deployment costs are higher, it is preferable to do formal verification of all protocols, including MAC protocols, before they are put into practical use to ensure design flaws and

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

other wrong behaviors do not reappear after deployment and thereby reduce redeployment overhead. The design faults include deadlocks, live-locks, non-progress cycles, the presence of unreachable states, etc. These design faults occur due to the interleaved execution of multiple processes that run the protocol. The effect of all the interleaved execution of multiple processes cannot be detected and analyzed using mere software testing. The various steps involved in the formal verification of protocol include a) abstracting protocol behavior by building validation models using an appropriate modeling language b) inserting design faults and other correctness properties to be verified using a temporal specification language, and finally c) formal verification using automated model checking tools. The main difficulty in formal verification is abstracting the behavior of the protocol under test. The abstraction mechanism for multi-path propagation and variable propagation delay in UWSN is proposed in this work. These behaviors must be incorporated into the modeling protocol for UWSN. Using the proposed abstraction mechanism, the TDA-MAC protocol [7] is modeled with PROMELA and verified with the SPIN model checker [20], [21]. The following are the major contributions of this work.

- Design of formal models to abstract the multi-path propagation of acoustic waves, and variable propagation delay based on changing ocean parameters in UWSN.
- Modeling and verification of TDA-MAC protocol used in UWSN.
- Reachability analysis of marked state inserted in the TDA-MAC model.
- Verification of refined TDA-MAC Protocol after detecting non-progress cycles and unreachable states.

The remainder of the paper is organized as follows. The motivation for this work is described in section II. The literature review is discussed in section III. Section IV describes the prerequisites needed to comprehend the concepts described in this work. The procedure for abstracting multi-path propagation in UWSN is described in section V-A. The procedure for abstracting variable propagation delay is described in section V-B. Section VI describes the modeling and verification of the TDA-MAC protocol. Section VII describes the comparison between our approach and similar efforts on TDA-MAC. Finally, section VIII describes the conclusions, scope, and future work.

II. MOTIVATION

Formal methods are applied in the software development life cycle to build quality software products that meet the functional requirements of the software. The hardware components used in UWSN are designed to withstand harsh ocean environments like underwater currents, corrosion effects, etc Therefore the deployment and hardware cost of UWSN components are more compared to those used in terrestrial networks. Formal verification of protocols running on UWSN components ensures the functional correctness of the behavior of the protocol early in the design stage. This will reduce the chance of the occurrence of incorrect behaviour after deployment saving redeployment and other cost overheads. Formal methods are rarely applied among protocol developers due to the vast amount of mathematical concepts involved in formal languages. The formal verification community widely uses formal model-checking tools integrated with an input specification language to conduct formal verification. Protocols used in UWSN are typically implemented directly from protocol specifications. The protocol's performance is first assessed by running it in a simulated environment. This is followed by the real-time deployment of UWSN with an implemented version of the protocol. Any failure of the protocol running in UWSN components after deployment is disastrous. The main challenge in the verification of protocols used in UWSN is to model the underwater acoustic environment where the protocol runs. This motivates us to create methods for modeling the communication environment for UWSN protocol to easily model UWSN protocols using the specification language of the model checker. Therefore, the modeling approaches for UWSN-specific variable propagation delay and multi-path propagation were proposed in this work.

III. LITERATURE REVIEW

In the literature, various protocols for UWSN were proposed, including the MAC layer, routing layer, and time synchronization protocols. Some notable works on contention-based MAC protocols include Propagation Delay Tolerant ALOHA Protocol (PDT-ALOHA) [13], Delay Tolerant MAC Protocol (DTMAC) [14], Multi-session Floor Acquisition Multiple Access Protocol (M-FAMA) [15], Full-duplex collision avoidance MAC (FDCA) [16], Depth Based Routing aware MAC Protocol (D(DBR-MAC) [17] and Traffic Adaptive Receiver Synchronized MAC Protocol (TARS) [18]. The major works on contention-free-based MAC protocols include a cross-layer MAC for underground acoustic sensor networks (CL-MAC) [5], Time Delay Aware MAC (TDA-MAC) [7], Cluster-based On-Demand Time Sharing MAC (COD-TS) [2], Throughput Efficient Super TDMA MAC (TES-TDMA) [6]. The approach used in the development of protocols for UWSN in all of these works includes a) developing protocol specifications, b) implementing the protocol using a suitable programming language, c) testing the protocol, d) analyzing the performance of the protocol using simulation tools, and e) analyzing the performance of protocols using real-world deployments in the ocean. However, the process described above has placed little or no emphasis on identifying protocol design flaws using formal verification. Protocol-related research works in UWSN are primarily focused on MAC protocols, followed by time synchronization protocols and routing protocols. Because of the long propagation delay, multi-path propagation, and turbulent ocean environment, the protocols used in UWSN have low throughput and collisions. The majority of UWSN MAC protocol research has focused on increasing throughput and reducing message collisions. Tables 1, 2, and 3 depict some

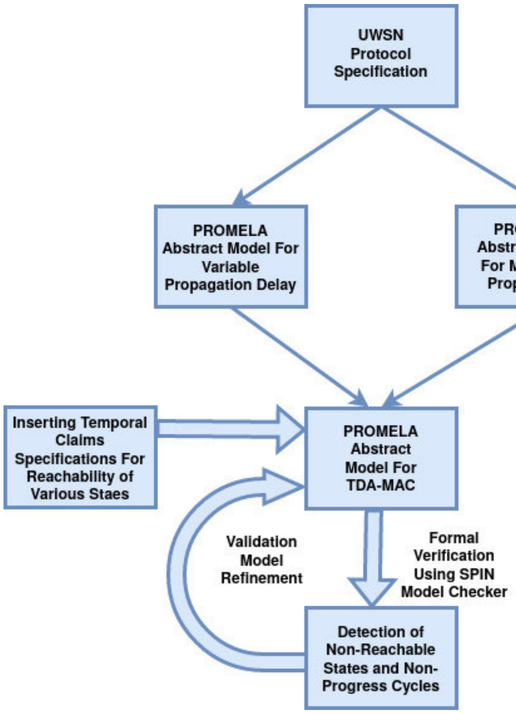

FIGURE 1. Schematic view of the proposed work.

of the MAC protocol-related works in UWSN. All of these works implement protocols from specifications, and their performance, such as throughput and message collisions, is analyzed and compared using network simulations and real-time deployments. None of the works focused on ensuring protocol behavior before implementation. The correct behavior of protocols can be ensured by identifying the design flaws inherent in the protocol as a result of concurrent process execution. These design flaws include the detection of deadlocks, live locks, non-progress cycles, dead codes, and other temporal logic-based behavior. Formal verification is a technique for detecting such design flaws. In this paper, we propose techniques for abstracting the behavior of UWSN protocols. The proposed abstraction scheme includes techniques for abstracting varying propagation delay and multi-path propagation in UWSN. These patterns are used to model and validate the UWSN TDA-MAC protocol, demonstrating its suitability for performing UWSN protocol verification. The protocol's formal model is created using Process Meta Language (PROMELA), and formal verification is performed using the SPIN model checker [21], [22]. Figure 1 depicts a schematic overview of the proposed work.

IV. PRELIMINARIES

A. COMPONENTS OF UWSN

UWSN is made up of two major arrangements. (1) Composed of components found on the ocean's surface. This includes both ocean surface buoys and terrestrial constituents. To determine the geographic location of various components, the terrestrial components transmit data via EM signal in conjunction with GPS. (2) The underwater components, which include underwater floating sensors, Autonomous Underwater Vehicles (AUVs), and seabed sensors, are the second component. Figure 2 depicts the overall architecture, which includes all of these components. UWSN protocols typically run on underwater components and communicate via acoustic waves.

TABLE 1. MAC protocols for UWSN.

Year	Type of work	Topic of Work
2007	Performance Analysis	A reservation-based MAC protocol (R-MAC) is proposed for underwater Sensor networks [1]. Analysis using simulation shows that the proposed protocol avoids packet collision and provides high energy efficiency and fairness.
2011	Performance Analysis	The Propagation Delay Tolerant ALOHA protocol (PDT-ALOHA) is proposed as an improved version of the slotted ALOHA protocol suitable for UWSN conditions. This protocol includes a guard time to compensate for the uncertainty in propagation delay in UWSN conditions [13].
2013	Performance Analysis	A MAC protocol, Multi-session Floor Acquisition Multiple Access (M-FAMA) that initiates multiple sessions from the sender side to improve throughput for UWSN is proposed [15].
2014	Performance Analysis	To study the impact of long pre-ample and low transmission rates in wireless acoustic sensor networks, collision probability, and nodal throughput models were developed for random access-based MAC and Handshake-based MAC. A time-sharing-based MAC is proposed that outperforms random access-based MAC and handshake-based MAC [2].
2014	Performance Analysis	Energy detection and the performance analysis of multiple input multiple output decision fusion of underwater sensor networks with k sensors communicate with N hydrophones are done using simulation studies [3]. The parameters considered for evaluation include sampling frequency, number of transmitting sensors, and number of hydrophones
2014	Performance Analysis	Proposed an energy-efficient multi-channel MAC protocol (DMM-MAC)that is suitable for underwater sensor networks that can handle bursty data traffic and multi-hop topology [4]. This MAC protocol runs on top of MM MAC. The performance analysis using simulation shows that DMM-MAC outperforms MM-MAC in terms of energy conservation and network performance in burst traffic.

B. INTRODUCTION TO MODEL CHECKING

Model checking is a technique for performing formal verification of the correctness of protocols. Model checking is carried out using automated, model checking tools as well as theoretical methods. The model checker receives as inputs a validation model created in a specification language and the correctness behaviors expressed in temporal logic.

TABLE 2. MAC protocols for UWSN cont'd.

Year	Type of work	Topic of Work
2015	Performance Analysis	A cross-layer MAC is proposed that makes use of the price-based rate allocation scheme used in the network layer [5]. The proposed protocol schedules contention-free packet transmission via single-hop subflows in each maximum clique. The simulation results of the proposed scheme show high end-to-end throughput and low channel access delay.
2015	Performance Analysis	A delay-tolerant MAC protocol (DT-MAC) for UWSN is proposed [14], which employs repeated transmission of packet 'm' times with transmission probability 'p'. The simulation results show that DTMAC outperforms other MAC protocols that use RTS/CTS exchange.
2016	Performance Analysis	Proposed transmission schedules in a linear UWSN [8]. The performance analysis of these transmission schedules under a single collision domain with unicast and broadcast traffic as well as a partially overlapping collision domain with unicast traffic was done. The parameters used for analysis mainly include throughput and packet delay
2016	Performance Analysis	A collision avoidance algorithm and a MAC protocol suitable for a full-duplex modem is proposed for UWSN in [16]. The throughput in comparison with other ALOHA protocols used in UWSN is analyzed.
2017	Performance Analysis	A traffic adaptive-based collision-free MAC protocol is proposed for UWSN in [18]. The proposed protocol is adaptive to varying traffic conditions including changes in the propagation delay and mobility of nodes in ocean currents.
2017	Performance Analysis	Proposed Transmit delay allocation MAC (TDA-MAC) and accelerated TDA-MAC that use Time Division Multiple Access and also does not use clock synchronization [7]. Performance analysis is done using simulation that uses real-time data. The analysis result reveals that throughput and packet loss are on par with synchronized TDMA.
2018	Performance Analysis	In [25], performance analysis of the real-time implementation of TDA-MAC that carries out environmental monitoring on a network installed on an underwater seabed is done. Based on the findings of sea trials, improvements to TDA MAC's robustness were suggested.
2018	Performance Analysis	An efficient Depth based collision-free MAC protocol (ED-MAC) that is capable of addressing spatial-temporal uncertainty and hidden terminal problems for UWSN is proposed in [32]. The packet delivery ratio and energy consumption fairness index of the proposed protocol are measured and compared with T-Lohi and UWAN MAC with varying traffic loads and node density. The analysis results reveal that ED-MAC performs better than T-Lohi and UWAN-MAC.

TABLE 3. MAC protocols for UWSN cont'd.

Year	Type of work	Topic of Work
2019	Performance Analysis	Proposed [26] is a TDA-MAC extension that offers dual-hop scheduling, energy efficiency, and significantly higher throughput. The MAC includes a routing redundancy technique that boosts channel capacity and packet delivery rates.
2019	Performance Analysis	A TDMA-based MAC protocol for UWSN that makes use of Graph coloring strategy is proposed in [28]. Performance analysis of the protocol using the Aqua-Sim simulator shows throughput improvement and energy efficiency compared with contention-based MAC protocols.
2019	Performance Analysis	The Packet delivery ratio, throughput and energy consumption of ED-MAC, DL-MAC and GC-MAC used in UWSN are evaluated with different data traffic rates and number of nodes in [29].
2019	Performance Analysis	A collision-free depth-based layering MAC (DL-MAC) for UWSN that is capable of addressing near-far effect, spatial-temporal uncertainty, and hidden/exposed terminal problems is proposed in [31]. The performance analysis of the protocol is done by measuring packet delivery ratio, energy consumption, throughput etc under varying traffic loads and node density.
2020	Review	This paper summarised the major advances in UWSN. Communication channel modeling, sustainable coverage, target detection techniques, MAC layer design, Time synchronization, localization algorithm design, and routing protocols in UWSN are among the topics covered [8].
2021	Review	A comprehensive review of characteristics, design techniques, pros, and cons of design techniques of collision-free MAC protocols used in UWSN are presented in [30].

The model checker examines each correctness behavior specification and determines whether the validation model satisfies it. This is accomplished by generating all possible sequences of the validation model's execution states. The model checker extracts the execution sequences that satisfy the negation of the correctness behavior from these sequences of execution states. In such cases, the model checker generates a violating trace, which depicts the sequence of execution states that leads to the correctness property being violated.

C. PROMELA AND SPIN MODEL CHECKER

Process Meta Language (PROMELA) is a specification language used to create validation models for formal verification. The PROMELA constructs that adhere to the requirements listed in the protocol specifications are used to build the validation models. Formulas using linear temporal logic (LTL) are used to specify the properties that must be verified after the validation model has been created. Simple PROMELA Interpreter (SPIN) is a model checker that examines each correctness behavior and determines whether

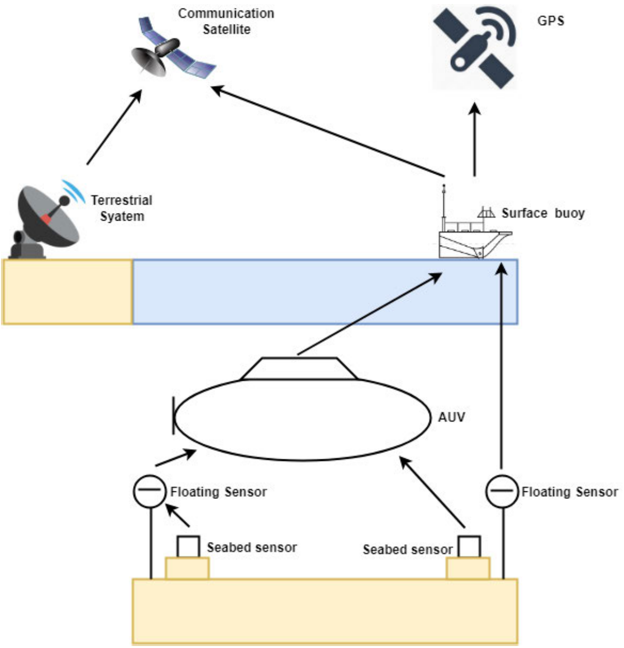

FIGURE 2. Components of UWSN.

the validation model satisfies it. This is accomplished by generating all possible sequences of the validation model's execution states. The SPIN model checker extracts the execution sequences that satisfy the negation of the correctness behavior from these sequences of execution states. In such cases, the SPIN model checker generates a violating trace, which depicts the sequence of execution states that leads to the correctness property being violated. An overview of the model-checking process using PROMELA and SPIN is outlined in Figure 3.

V. MODELING BEHAVIORAL PATTERNS FOR UWSN

This section deals with modeling some specific behavior of UWSN. The behavior patterns identified include multi-path propagation and varying propagation delay. The deployment pattern of sensor nodes in ocean beds follows a two-dimensional deployment strategy [8]. The network topology used to describe the behavior consists of 8 X 8 Km coverage area. The distance between the consecutive nodes is 200 meters. The gateway node is placed in the center of the coverage area with a vertical depth of 500 meters to seabed [7]. The notations used in the algorithms are based on pseudo-code conventions described in [23] and [24] and are reproduced in Table 4 for clarity and understandability.

A. MODELLING MULTI-PATH PROPAGATION

The phenomena of multi-path propagation are caused by curving routes, echo, and reflection of the acoustic wave on the ocean surface and bottom, resulting in the same signal reaching the receiver many times. This acoustic wave feature can be abstracted by transmitting and receiving identical data several times for each message transmission and reception. Before each data transmission, the number of times the identical data is transmitted is chosen in a non-deterministic way. Multiple transmissions of the same data

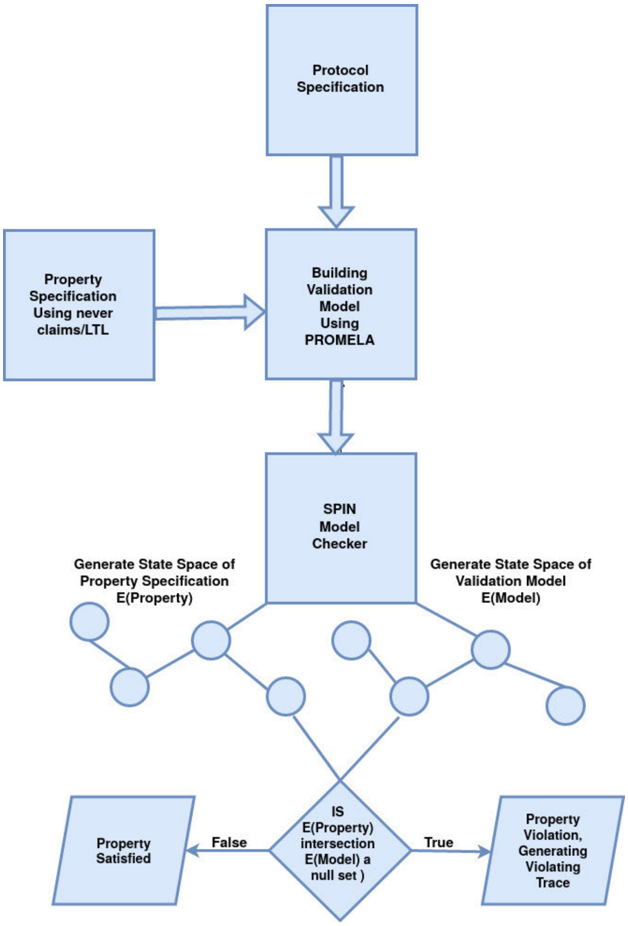

FIGURE 3. Model checking process using SPIN and PROMELA.

TABLE 4. Summary of notations used in Algorithms [23], [24].

Slno	Notation	Meaning
1	$a \leftarrow b$	Assignment
2	$a = b$	Logical Equal To
3	$a \leftarrow$ copy b	Copying of objects
4	$a{=}{<}e_0, e_1, , ...e_{n-1}{>}$	A sequence 'a' containing a linear ordering to a collection of n elements. The elements in sequence do not have to be distinct.
5	$\|a\|$	Length of a sequence a
6	a_i	i^{th} element of sequence 'a' where 'i' belongs to $\{0..\ \|a\| - 1\}$
7	In	Argument parameters passed to the algorithm
8	Out	It is the result that is returned to the caller algorithm
9	local	local values and the scope is local to the algorithm
10	$S{=}{<}{<}a_0,b_0{>},{<}a_1,b_1{>}{>}$	A table S containing 2 rows and columns
11	$S[i,j]$	j^{th} element in the i^{th} row of table 'S'
12	$a\|b$	The symbol 'a' is concatenated with the value of 'b'
13	choice $e_1, e_2, ...e_n$	Any one episode from e_1 to e_n is selected in a non-deterministic way.

occur in a very short time frame when compared to a standard data transmission time interval. The multi-path propagation algorithm for data transmission and reception is depicted in Algorithms 1 and 2 respectively.

The example shown in Algorithm 1 involves a gateway node sending a request message (**reqmsg**) to all N nodes placed on a seabed. Each **reqmsg** request to a specific node is sent multiple times to abstract the multi-path propagation behavior. Step 5 of Algorithm 1 describes how to determine the number of times additional transmissions of the same message happen. This is done by choosing a value between 1 and NPROP in a non-deterministic fashion. The repeated transmission of **reqmsg** in steps 6 through 11 of Algorithm 1 describes the abstraction of multi-path propagation during transmission. The constant **MPI**, which stands for a multi-path interval, specifies the time window for the repeated delivery of **reqmsg**. In the **mpathtimer**, this value is initialized. The constant **MI** specifies the time window in which the subsequent round of **reqmsg** is to be transmitted to all nodes. The **msgtimer** in step 3 of Algorithm 1 initializes this value. Step 13 of the Algorithm 1 describes how the next round of message transmission is started after one round of message transmission to all the **N** nodes.

The scenario shown in Algorithm 2 is where a node placed on the sea bed receives a request message (**reqmsg**) that was sent by the gateway node by the procedures indicated in Algorithm 1. The same constants **MPI**, **MI**, and **NPROP** are used in this algorithm. Additionally, it makes use of the boolean variable **multipath**, which is initially set to **false**. This algorithm processes the message sent by the gateway node only when the value of the variable **multipath** is **false**. The boolean variable **multipath** is set to **true** after receiving the first message sent by the gateway. Until the value of the **multipath** variable is changed to **false**, this will disregard multiple receptions of the same message sent by the gateway node. When the limit time period for a number of transmissions of the same message expires, the value of the **multipath** variable is changed to **false**. This is explained in steps 5 to 6 of Algorithm 2. The protocol designer fixes the values of the constants **MPI**, **MI**, and **NPROP** used in Algorithms 1 and 2 in a way that the value of MI is sufficiently greater than the entire maximum time needed to propagate the **reqmsg** to all N nodes.

Algorithm 1 Procedure MULTIPATH_SEND

This algorithm describes the abstraction steps for implementing multi-path propagation
during transmission in UASN
In : reqmsg
Constant MPI,MI,NPROP
Local msgtimer,mpathtimer,i

```
1:  while true do
2:      i ←0
3:      set (msgtimer,MI)
4:      for $i \leftarrow 1 to N$ do
5:          choice among rnd ← 1 to NPROP
6:          while rnd $\neq$ 0 do
7:              send reqmsg via channel allotted to the transmitting node
8:              rnd ← $rnd - 1$
9:              set (mpathtimer,MPI)
10:             wait till expire(mpathtimer)
11:         end while
12:     end for
13:     wait till expire(msgtimer)
14: end while
```

Algorithm 2 Procedure MULTIPATH_RECV

This algorithm describes the abstraction steps for implementing multi-path propagation
during reception in UASN
Constant MPI,MI,NPROP
Boolean multipath=false
Local mpathtimer,recvpmsg

```
1: while true do
2:     if multipath = false then
3:         receive the message via the channel allotted to the receiving node and store it in recvmsg
4:         multipath ← true
5:         set(mpathtimer,MPI $\times$ (NPROP-1))
6:         wait till expire(mpathtimer)
7:     end if
8:     multipath ← false
9: end while
```

B. MODELLING VARIABLE PROPAGATION DELAY

A method for abstracting variable propagation delay is presented in Algorithm 3. This algorithm assumes N nodes deployed in 8 X 8 KM in an underwater sea bed. The gateway node is deployed in the center of the coverage area with a vertical depth of around 500 meters. The algorithm uses two sequences D and V as input and the sequence TP and PD as output. The sequence $D = < D_1, D_2, \ldots .D_N >$ represents distances from the gateway to N nodes deployed in the underwater seabed. The sequence $V =< V_1, V_2..V_K >$ represents the velocity of sound waves depending on ocean conditions. $TP =\ll_{11}, TP_{12} \ldots TP_{1K} >< TP_{21}, TP_{22} \ldots .TP_{2K} > \ldots < TP_{N1}, TP_{N2} \ldots .TP_{NK} \gg$ is a table of N rows and K columns. Each row stores the propagation delay corresponding to the velocities V_1 to V_K for a particular Node 'i' where 'i' ranges from 1 to N. Steps 1 to 5 of Algorithm 3 calculate the propagation delay of each node 'i' for different velocities V_1 to V_K. These values are stored in a two-dimensional array TP that contains N rows and K columns. Each row of TP contains the propagation delay of each node from the gateway node under various velocities V_1 to V_K. Steps 6 to 9 of Algorithm 3 perform

Algorithm 3 Procedure PDSELECTGATEWAYNODE

This algorithm describes the abstraction steps for implementing variable propagation delay in UASN

In : D,V

Out: TP, PD enriched with values values fixed by this Algorithm

Local h,i,j

Constant N,K

```
1:  for $i \leftarrow 1 to N$ do
2:      for $j \leftarrow 1 to K$ do
3:          $TP[i, j] \leftarrow V_j \div D_i$
4:      end for
5:  end for
6:  choice among h ← 1 to K
7:  for $i \leftarrow 1 to N$ do
8:      $PD_i \leftarrow TP[i, h]$
9:  end for
10: while true do
11:     Abstraction steps requesting data to nodes deployed on seabed.
12:     Abstraction steps receiving data from each node deployed on the seabed.
13:     choice among rnd= 1 to 100
14:     if rnd is greater than a threshold value then goto step 6.
15: end while
```

a non-deterministic selection of a value from 1 to K and is stored in a variable ‘h’. This value is used as a column index of TP to select a particular propagation delay for each of the N nodes. The selected values of the propagation delay of N nodes are stored in the sequence PD. Each node used by the validation model of the protocol makes use of the selected propagation delay. Steps 10 through 12 of Algorithm 3 specify a high-level description of the protocol used in UASN for collecting sensor data from the nodes deployed in the underwater seabed. The detailed steps vary depending on UASN applications. The abstraction steps for change in the propagation delay due to varying ocean conditions are done by a non-deterministic jump to step 6 of the Algorithm which repeats the selection of another propagation delay from the table TP for all nodes and runs the same protocol.

VI. DESIGNING, MODELING AND VERIFICATION OF TDA-MAC

Underwater acoustic sensor networks (UASN) use the Time Delay Allocation (TDA) protocol, which was first described in [7]. This MAC protocol does not necessitate node clock synchronization, in contrast to other MAC protocols used in UASN. The protocol is presumptively used in a topology-defined environment as indicated in Section V. The TDA MAC protocol design process entails four steps: a) identifying its design elements; b) determining the major states that the protocol passes c) developing communicating finite state machines for the protocol and d) building a validation model of the protocol using a specification language. The Process Meta Language (PROMELA) is used as a specification language to model the TDA MAC. Finally, the PROMELA model of TDA MAC’s formal verification includes checking for design flaws and the reachability of various marked states in the model using SPIN model checker.

A. DESIGN ELEMENTS OF TDA-MAC

The design elements for TDA MAC were taken from the studies described in [7]. The various procedures used to determine the design elements of the TDA MAC are based on the theories presented in [21] and [22].

1) **Protocol Service**
This protocol will enable slotted sensor data receiving, similar to TDMA, at the UASN gateway node from nodes installed in the seabed below. This protocol is capable of handling variable propagation delay under UASN ocean conditions.

2) **Assumptions about the Protocol Environment**
The protocol needs to be able to function in an environment with wireless acoustic signal propagation with a partially lossy environment where some messages propagated via channels may be lost. In contrast to terrestrial networks, UASN communication propagation delay is greater and varies with ocean factors such as depth, salinity, pressure, temperature, etc.

3) **Protocol Vocabulary**
The protocol starts by sending a **PING** message from the gateway node to each node installed on the bottom of the ocean. The nodes positioned beneath the seabed relay the same message to the gateway node after receiving a PING request. From the gateway node to each additional node, this procedure is repeated. The propagation delay from the gateway node to each adjacent node is measured. The transmit delay intervals of data packets from each node are computed in the gateway node. This time period represents the amount of time it takes a sensor node to deliver a data packet to a gateway node in response to a **REQ** message. Each node receives this estimated transmit delay interval via a **TDI** packet. To all the nodes, the gateway node broadcasts the **REQ** request packet. Each node that receives a **REQ** packet from the gateway node waits for the transmit delay interval (TDI) before sending a **DATA** packet containing the deployed node’s sensor reading. As a result, the gateway node gets the data packets in a time-slotted fashion from all sensor nodes installed in the sea bed.

4) **Protocol Message Format**
The message format consists of a message type **msgtype**, which can be one of the following: **PING**, **TDI**, **REQ**, or **DATA**; a source and destination id

(source, dest); and a message **msg**. The message format is as follows:
{**msgtype**, **source**, **dest**, **msg** }.

5) **Procedural Rules**
There are typically two different implementations of the TDA MAC protocol, one running on the gateway node and the other on nodes placed on the seabed. The following procedural requirements must be followed by the protocol.

a) A gateway node must wait for the same **PING** packet to arrive at the gateway node from the node it was originally sent.

b) The **PING** packet must be relayed back to the gateway node if it is received by the sensor node installed in the seabed.

c) Calculate the time elapsed since the last **PING** packet was transmitted and received (propagation delay) at the gateway node if the gateway node receives a **PING** packet.

d)To all **N** deployed nodes on the seabed below, apply procedure rules a, b, and c.

e) Use the propagation delay that was estimated using procedure rules a, b, and c to calculate the transmit delay from the gateway node to all nodes.

f) Update the sensor node's transmit delay to the data message received from the **TDI** packet if the gateway node sends a TDI packet to the sensor node installed in the seabed.

g) Upon receiving a **REQ** packet from the gateway node, a sensor node installed on the bottom of the sea will then send out a data packet containing sensor data after the transmit delay designated for that node.

h)Calculate the difference between the actual time delay and the anticipated time delay between the transmission of a REQ packet to the reception of a DATA packet if the gateway node receives a DATA packet from a sensor node. Apply procedure rules from e to h and adjust the propagation delay and transmit delay interval based on the timing inaccuracies if the difference exceeds a threshold.

B. FINITE STATE MACHINE OF TDA MAC

The Finite State model for the TDA-MAC protocol is shown in Figures 4 and 5 for gateway nodes and sensor nodes placed on the seabed below. Tables 4 and 5, respectively, show a description of the various state transition activities of the finite state machines of the gateway node and sensor nodes. The total number of sensors placed on the seafloor is represented by the constant MAXNODE shown in Figure 4. The transition from the PING STATE to TRANSMIT COMPUTE action $i > MAXNODE$ in Figure 4 denotes the end of the transmission and reception of PING messages to all sensor nodes with the numbers 1 to MAXNODES. In Figure 4, the action $i > MAXNODE$ for the transition from TRANSMIT COMPUTE to TTX STATE denotes the completion of the computation of transmit timings for all sensor nodes numbered 1 to MAXNODES. Each sensor node will wait until this transmission time has passed after receiving a REQ message before sending a DATA message to ensure that the DATA messages are received at the gateway node in a timed slotted way. The completion of the transmission of the TDI message to all of the sensor nodes with the numbers 1 to MAXNODES is indicated by the action $i > MAXNODE$ for the transition from TTX STATE to REQUEST STATE in Figure 4. According to Figure 5, the action set(Ttx,msg1) for the transition from UPDATE TTX STATE to INIT involves writing the value of msg1 obtained from the TDI packet into a variable Ttx in the sensor node. In Figure 5, the action expire(timer) for the transition from DATA TRANSMIT STATE to INIT state denotes the expiration of a timer that is set to Ttx received as a message from TDI packet.

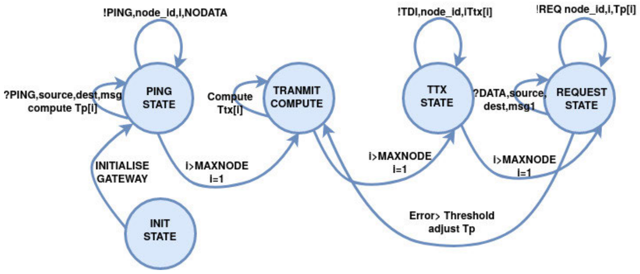

FIGURE 4. Finite state machine of gateway node.

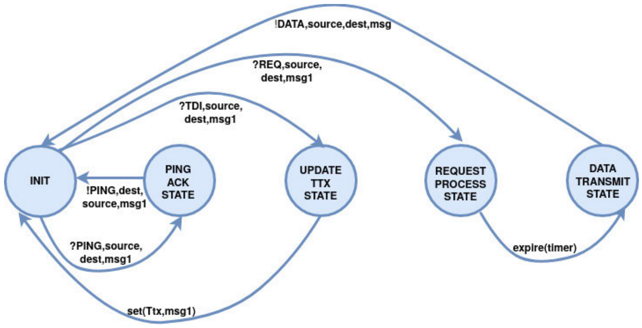

FIGURE 5. Finite state machine of sensor node.

C. MODELLING TDA-MAC PROTOCOL USING PROMELA

The topology presented in section V is used in TDA-MAC modeling. With 200 meters separating each node, a coverage area of 8 × 8 KM is utilized. The gateway is positioned in the middle of the coverage region, 50 meters vertically below the ocean's surface. The ocean's surface extends 500 meters above its submerged surface. The 4 boundary nodes of the 8 X 8 KM coverage region are instead selected as depicted in Figure 6 due to the state space explosion experienced during formal verification. The labels "G," "N1," "N2," "N3," and "N4" refer to the gateway node and the other nodes, accordingly. The set distance between each node and the gateway is 7485 meters. According to [12], the estimated propagation delay for sound waves traveling at velocities between 1450 m/sec and 1540 m/sec

TABLE 5. State transition table of gateway node For TDA-MAC.

Slno	Present State	Next State	Actions
1	INIT STATE	PING STATE	Initialise Gateway Parameters
2	PING STATE	PING STATE	Sending a PING message to one of the sensor node
3	PING STATE	PING STATE	On receiving a PING message from one of the sensor nodes. Computation of propagation delay from gateway node to sensor node
4	PING STATE	TRANSMIT COMPUTE	On receiving PING message back from all the sensor nodes
5	TRANSMIT COMPUTE	TRANSMIT COMPUTE	Compute Ttx for all sensor nodes
6	TRANSMIT COMPUTE	TTX STATE	On completion of computation of Ttx for all sensor nodes
7	TTX STATE	TTX STATE	Sending TDI message to each nodes
8	TTX STATE	REQUEST STATE	On completion of sending TDI message to all sensor nodes
9	REQUEST STATE	REQUEST STATE	Sending REQ message to each nodes
10	REQUEST STATE	REQUEST STATE	On receiving a DATA message from each sensor nodes
11	REQUEST STATE	TRANSMIT COMPUTE	Delay between sending a REQ packet and receiving DATA packet exceed a threshold limit

TABLE 6. State transition table of sensor nodes for TDA-MAC.

Slno	Present State	Next State	Action
1	INIT	PING ACK STATE	On receiving a PING message from gateway node
2	INIT	UPDATE TTX STATE	On receiving a TDI message from gateway node
3	INIT	REQUEST PROCESS STATE	On receiving a REQ message from gateway node
4	PING ACK STATE	INIT	Sending back the same PING message to gateway node
5	UPDATE TTX STATE	INIT	update the Ttx value of a node with the message received from TDI message
6	REQUEST PROCESS STATE	DATA TRANSMIT STATE	on expiry of Ttx value assigned to the timer of a sensor node
7	DATA TRANSMIT STATE	INIT	Send a DATA packet to a gateway node

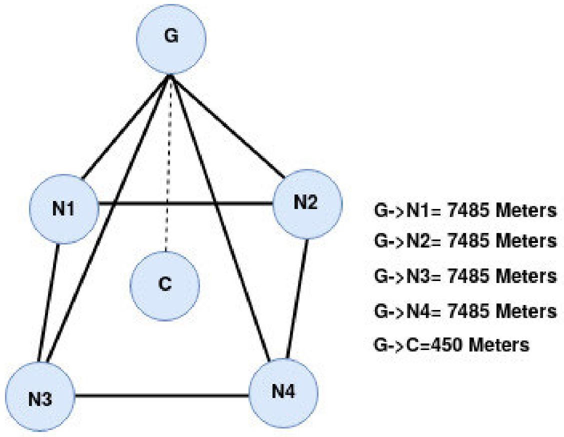

FIGURE 6. Topology used by the PROMELA model.

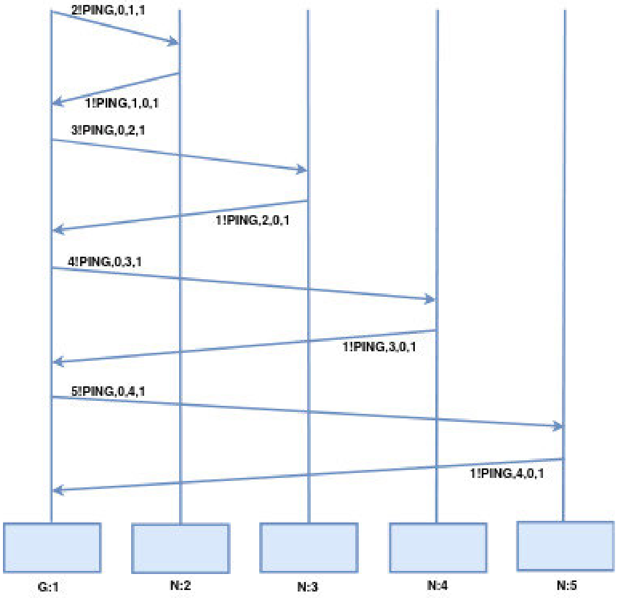

FIGURE 7. PING message sequence.

ranges from 5162 to 4860 milliseconds. To model variable propagation delay, the Algorithm 3 described in section V-B is used. The input sequence D for Algorithm 3 contains the values $D =< 7485, 7485, 7485, 7485 >$ and has a length of 4. These values stand for the distance between the gateway node and each boundary node N1, N2, N3, and N4. The input sequence V utilized in Algorithm 3 has V=<1450, 1451, . . . 1539, 1540> representing the varied velocities of acoustic waves in the deep ocean. After applying Algorithm 3, the output table TP is created. It has 4 rows and 91 columns. With 91 different velocities contained in sequence V, each row of the table contains the propagation delay values for each node. For each node in the model, the non-deterministic Algorithm 3 chooses a certain propagation delay corresponding to a specific velocity. Figures 7, 8 and 9 each provide a description of the message sequence that the model generated in accordance with the procedural rule outlined in section VI-A.

D. VERIFICATION OF TDA-MAC PROTOCOL

The PROMELA model that was previously built includes the states that are described in the FSM of TDA-MAC that is discussed in section VI-B. Table 7 shows the states introduced to the PROMELA model for the gateway node and the other four nodes, along with their acronyms. The verification is carried out by examining the reachability of marked states

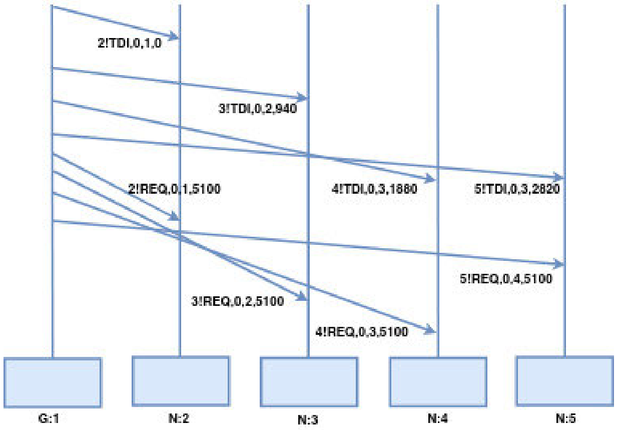

FIGURE 8. **TDI-REQ message sequence.**

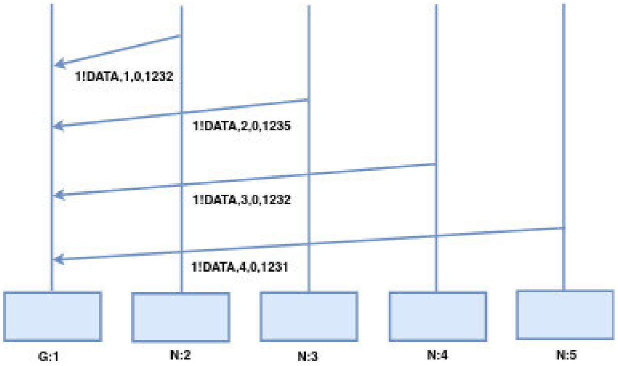

FIGURE 9. **Data message sequence.**

TABLE 7. **Marked state description of TDA-MAC model.**

Slno	Node Type	State	Abbreviation
1	Gateway	PING_STATE	PS
2	Gateway	TRANSMIT_COMPUTE	TC
3	Gateway	TTX_STATE	TTX
4	Gateway	REQUEST_STATE	RQ
5	N1,N2,N3,N4	INIT	IN
6	N1,N2,N3,N4	PING_ACK_STATE	PAS
7	N1,N2,N3,N4	REQUEST_PROCESS_STATE	RQP
8	N1,N2,N3,N4	UPDATE TTX_STATE	UTTX
9	N1,N2,N3,N4	DATA_TRANSMIT_STATE	DTS

in the gateway node and the other four nodes in a partially lossy channel environment where some PING, TDI, REQ, and DATA packets were lost. By including a Daemon process capturing the PING, TDI, REQ, and DATA packets from the channels designated to N1, N2, N3, and N4, respectively, various packet losses can be emulated. Figures 10 and 11 show the Finite State Machine for the Never Claims used for testing the reachability of each state in the gateway node in case of PING packet loss. It is evident from the FSM shown in Figures 10 and 11 that when a PING packet is received in the Daemon process, the state change from S4 to S8 takes place. This suggests that the PING packet that the gateway node transmitted was unsuccessful. The gateway node's inability to reach the TDI_STATE is indicated by the transition from $(!((G[pid0]_{p} == TDI_{S}TATE)))$ to the same state. The never claim has an acceptance label next to the state s8. If the concurrent execution of the model results in an acceptance cycle, the state S8 of the never claim is cycled infinitely. Additionally, it shows that the PROMELA model of TDA MAC is not progressing toward its TDI_STATE. Similar to this, the never claim and FSM for other states in the gateway and other nodes' non-reachability can be obtained. In Figure 12, the never claim corresponding to the non-reachability of the states TRANSMIT_COMPUTE, TDI_STATE, and REQUEST_STATE is depicted. Similar to this, the never claims for additional states in the model's non-reachability are derived and verified. The Bit state Hashing/Super Trace approach was used to lower the amount of states produced by the validation model. In addition to the aforementioned, the maximum search depth and physical memory size are fixed at 2048 Megabytes and 10000000, respectively. The model also verifies the

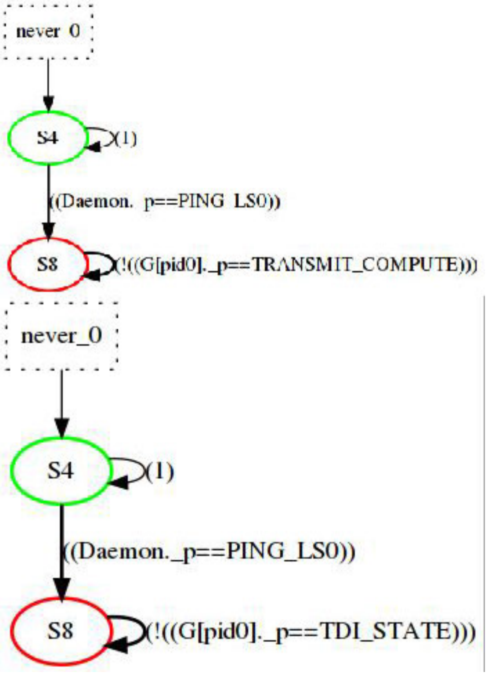

FIGURE 10. **FSM never claim for reachability to TRANSMIT_COMPUTE, TDI_STATE for PING packet loss.**

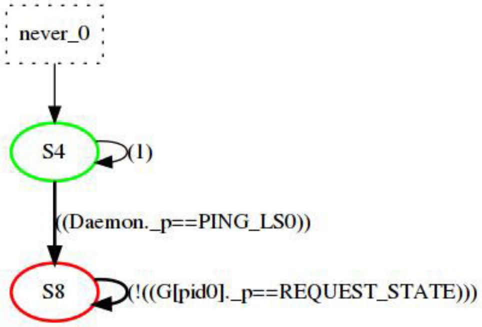

FIGURE 11. **FSM never claim for reachability to REQUEST_STATE for PING packet loss.**

```
never
{
progress_rpt:do
::  Daemon@PING_LS0->break;
::  true;
od;
acceptall:  do
::!(G[pid0]@TRANSMIT_COMPUTE);
od;
}

do
::  Daemon@PING_LS0->break;
::  true;
od;
acceptall:  do
::!(G[pid0]@TDI_STATE);
od;
}

never
{
do
::  Daemon@PING_LS0->break;
::  true;
od;
acceptall:  do
::!(G[pid0]@REQUEST_STATE);
od;
}
```

FIGURE 12. Never claim for reachability to TRANSMIT_COMPUTE, TDI_STATE and REQUEST_STATE for PING packet loss.

TABLE 8. Reachability of marked states of gateway node TDA-MAC model in case of various message loss.

State	PING Send	TDI Send	REQ Send	PING Recieve	PING Recieve
PS	✗	✓	✓	✓	✓
TC	✗	✓	✓	✓	✓
TTX	✗	✓	✓	✓	✓
RQ	✗	✓	✓	✓	✓

reachability of the marked states in the gateway node, namely the PING_STATE, TRANSMIT_COMPUTE, TTX_STATE, and REQUEST_STATE in the case of PING, TDI, and REQ message loss from the gateway node to N1, N2, N3, and N4, as well as loss of PING, DATA message reception from N1, N2, N3, and N4. In addition to the foregoing, the existence of non-progress cycles in each of these cases is also verified. Table 8 displays the verified results.

When a specific message send or receive indicated in the column of the table by the gateway node to any one of the nodes N1, N2, N3, and N4 are lost, a ✗in the Table 8 shows the presence of a non-progress cycle and non-reachability to the state's PS, TC, TTX, and RQ. When a specific message send or receive defined in the table's column by the gateway node to any one of the nodes N1, N2, N3, or N4 is lost, a ✓in the Table 8 shows the absence of a non-progress cycle and reachability to the states PS, TC, TTX, and RQ. When the PING packet is sent by the gateway nodes to any of the nodes N1, N2, N3, and N4, it is evident from Table 8 that the reachability to the states PS, TC, TTX, and RQ is not satisfied. Additionally, non-progress cycles were seen in each of these cases. This finding leads us to the conclusion that the protocol specification of the TDA MAC published in [7] contains several design flaws. As a result, the existing TDA MAC model needs the following improvements. To guarantee reachability to the states PS, TC, TTX, and RQ in the gateway node and to prevent a non-progress cycle, these enhancements are recommended.

- The gateway node sends the same PING packet to the same node again if it doesn't receive an acknowledgment of it after a predetermined amount of time. The work in [25] proposes a retransmission of all the packets of TDA-MAC a fixed number of times as a part to address the error in transmit delay computation and collisions identified due to the loss of various packets sent to a node during sea trials. The same recommendation is applied in this work to prevent a non-progress cycle when the PING packet acknowledgment is not received from a particular node. This issue was not addressed in the specification of TDA-MAC that was put forth in [7]. We verified the model with a retry count ranging from 2 to 10 in our validation model for the gateway node that employs a partly lossy environment and the topology shown in Figure 6. For the model to guarantee reachability to all of the model's states and the lack of non-progress cycles, a retry count value of 3 is sufficient.
- Even after the gateway node has sent the PING packet an allocated number of times if the acknowledgment of the PING packet is not received by the gateway node, avoid sending the PING packet again to that node and continue to send PING packet to the next node. This recommendation is done to prevent infinite cycling through PS state and also to guarantee reachability to the states TC, TTX and RQ states in the gateway node even if the connectivity to a particular node is fully lost during the PING packet exchange. This issue was not addressed in the specification of TDA-MAC that was put forth in [7].
- If the acknowledgment to the PING packet sent to the last node by the gateway node is not received even after a specific number of retries, the protocol should move to the TC state. This recommendation is done to ensure the reachability to TC state and to avoid a non-progress cycle if the connectivity to the last node is fully lost during a PING packet exchange. This issue was not addressed in the specification of TDA-MAC that was put forth in [7].

The verification result of the revised TDA-MAC model after incorporating these recommendations is shown in Table 9. It is evident from the observations in Table 9 that the refined TDA-MAC model's requirements for reachability to all states

TABLE 9. Reachability of marked states of gateway node refined TDA-MAC Model in case of various message loss.

State	PING Send	TDI Send	REQ Send	PING Recieve	DATA Recieve
PS	✓	✓	✓	✓	✓
TC	✓	✓	✓	✓	✓
TTX	✓	✓	✓	✓	✓
RQ	✓	✓	✓	✓	✓

TABLE 10. Reachability of marked states of N1, N2, N3, and N4 in the refined TDA-MAC model in case of various message loss from and to gateway node.

State	PING Send	TDI Send	REQ Send	PING Receive	DATA Receive
PAS	✓	✓	✓	✓	✓
UTTX	✓	✓	✓	✓	✓
RQP	✓	✓	✓	✓	✓
DTS	✓	✓	✓	✓	✓

in the gateway node and the absence of non-progress cycles in the model are met. As a result, the TDA-MAC protocol's gateway node is free of design flaws and is reachable to the designated states.

The reachability of the marked states, PING_ACK_STATE, UPDATE_TTX_STATE, REQUEST_PROCESS_STATE, and DATA_TRANSMIT_STATE in the nodes N1, N2, N3, and N4 of the model in case of PING, TDI, REQ message loss from the gateway node as well as loss of PING, DATA message reception from N1, N2, N3, and N4 to gateway node, is also confirmed. In addition to the foregoing, the existence of non-progress cycles in each of these cases is also verified. Table10 shows the verified results. It is evident from the findings in Table10 that the refined TDA-MAC model's reachability to all of the states in nodes N1, N2, N3, and N4 as well as the model's lack of non-progress cycles are satisfied. As a result, there are no design flaws or non-reachability to the designated states on the nodes N1, N2, N3, and N4 that model the TDA-MAC.

VII. COMPARISON BETWEEN OUR APPROACH AND SIMILAR EFFORTS ON TDA-MAC

From its initial proposal to its expanded version, the findings in the recent works related to TDA-MAC and our findings in this work are compared in Table 11. The original suggestion for TDMA-based protocol for UWSN was without employing time synchronization of clocks and the performance study was carried out using simulation [7]. As per the formal verification results obtained in this work, a non-progress cycle and non-reachability to the states PS, TS, TTX, and REQ in the gateway node are detected when the PING packets are lost. This indicates that the protocol proposed in [7] do not address the actions to be taken due to the loss of PING, TDI, REQ, and DATA packets. In a later work, [25], an analysis of the real-time deployment of TDA-MAC on hardware in underwater ocean conditions was proposed. When PING, TDI, REQ, and DATA packets are lost, there are practical problems with transmit delay and collision of DATA and REQ packets. To address these practical concerns, this work suggests changes to the TDA-MAC. Though the work [25] propose to switch to the next node in the event of loss of PING packet, the protocol goes to unreachable states and non-progress cycles at the gateway node when the PING packet to the last node is lost. This indicate that the modifications proposed do not address the actions to be taken due to the loss of PING packet sent to the last node. It is evident from these two works that the performance of TDA-MAC in a simulation environment cannot fully address and detect the practical problems encountered during real-time deployment. Before a protocol is deployed, many of these practical concerns can be found using formal verification methods, giving developers the assurance that the protocol is valid. By doing this, extra overheads required to restructure the protocol after deployment will be greatly reduced. In this work, we concentrated solely on detecting design flaws that were included in the initial specification of TDA-MAC since the validation models utilizing PROMELA are primarily meant to discover design flaws before implementation. The study mentioned in [26] offers a novel TDA-MAC variant that can operate in the dual-hop topology employed in UWSN. This will address the issue of TDA-MAC nodes requiring a lot of power in single-hop environments. The authors also suggest a redundant routing scheme that improves TDA-MAC's reliability. It is also possible to model and

TABLE 11. Comparison between our approach and similar efforts on TDA-MAC.

Slno	Findings on the recent works on TDA-MAC	Findings based on the current work
1	A TDMA-based TDA-MAC for UWSN that does not require time synchronization whose performance comparable to other MAC protocols that use time synchronization is proposed in [7]	Unreachable states and non-progress cycles at gateway node in lossy channel environment indicate that the protocol proposed in [7] does not address the actions to be taken due to the loss of PING, TDI, REQ, and DATA packets.
2	Sea trials for the TDA-MAC protocol proposed in [7] were conducted in [25]. The results obtained after sea trials identify the practical issues in computing time delays due to the loss of PING, TDI, REQ and DATA packets and suggest modifications to TDA-MAC to address these practical issues	Unreachable states and non-progress cycles at the gateway node when the PING packet to the last node is lost, indicates that the modifications proposed in [25] does not address the actions to be taken due to the loss of PING packet send to the last node.
3	A modified form of TDA-MAC capable of working in dual-hop communication with less transmit power and redundant routing path feature is proposed in [26]. Performance analysis using simulation study reveals high packet delivery ratio compared to the previous variants	Correctness on the behavior of this modified form of TDA-MAC needs to be verified using formal verification before conducting sea trials such that refinement of the protocol after deployment can be avoided.

formally verify the timing concerns found in [25] due to the loss of PING, TDI, REQ, and DATA packets by applying our abstraction methods and finite state machines introduced in this work and utilizing UPPAAL model checker [27] specifically meant to verify timing issues. Before undertaking sea trials, it is recommended to use model checking methods for the verification of the modified TDA-MAC proposed in [26] in order to confirm the correctness of the protocols prior to real deployment.

VIII. CONCLUSION, SCOPE AND FUTURE WORK

We have put forth abstraction strategies for UWSN protocols. Multi-path propagation and varied propagation delays which are typical in ocean conditions are abstracted and specifications are represented using PROMELA. Algorithms for sending and receiving of messages in scenarios like multi-path propagation and varied propagation delay are presented. The TDA-MAC protocol used in UWSN is modeled and tested to show the usefulness of the suggested algorithms. The gateway nodes and other nodes' reachability to various indicated states are verified. In a lossy channel context, the TDA-MAC model is also checked for non-progress cycles and design flaws. The verification outcome demonstrates that PING packet loss has an impact on the reachability of various marked states in the gateway node. Additionally, non-progress cycles can also be identified when a PING packet is lost. With certain extra requirements proposed to the current protocol standards, the validation model is improved. The redesigned validation model, in the end, satisfies the reachability criteria and specification under a variety of message loss scenarios.

The reach of this work is wide. The suggested techniques can be applied for the formal verification of any existing or developing protocols in the context of UWSN. It is possible to test the functional correctness and design flaws of protocols used in UWSN using the idea and methods proposed in this paper. The current work has given us an opportunity to model and formally verify both underwater acoustic communication environments along with the procedural rules of TDA-MAC protocol proposed in [7].

We plan to expand our modeling techniques to other protocols used in UWSN as well as to further examine the impact of multi-path propagation and varying propagation delay on the correctness of MAC protocols used in UWSN especially Sequential Dual Hop TDA-MAC (SDH-TDA-MAC) and its redundant routing mechanism. Additionally, it is intended to develop semantics and process algebraic models that explain multi-path propagation and variable propagation delay, as well as to do static analysis of both current and evolving UWSN protocols.

REFERENCES

[1] P. Xie and J.-H. Cui, "R-MAC: An energy-efficient MAC protocol for underwater sensor networks," in *Proc. Int. Conf. Wireless Algorithms, Syst. Appl. (WASA)*, Aug. 2007, pp. 187–198.

[2] Y. Zhu, Z. Peng, J.-H. Cui, and H. Chen, "Toward practical MAC design for underwater acoustic networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 872–886, Apr. 2015.

[3] P. Salvo Rossi, D. Ciuonzo, T. Ekman, and H. Dong, "Energy detection for MIMO decision fusion in underwater sensor networks," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1630–1640, Mar. 2015.

[4] C.-M. Chao, M.-W. Lu, and Y.-C. Lin, "Energy-efficient multichannel MAC protocol design for bursty data traffic in underwater sensor networks," *IEEE J. Ocean. Eng.*, vol. 40, no. 2, pp. 269–276, Apr. 2015.

[5] X. Su, S. Chan, and M. Bandai, "A cross-layer MAC protocol for underwater acoustic sensor networks," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4083–4091, Jun. 2016.

[6] 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.

[7] N. Morozs, P. Mitchell, and Y. V. Zakharov, "TDA-MAC: TDMA without clock synchronization in underwater acoustic networks," *IEEE Access*, vol. 6, pp. 1091–1108, 2017.

[8] A. Boukerche and P. Sun, "Design of algorithms and protocols for underwater acoustic wireless sensor networks," *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–34, Nov. 2021.

[9] G. Acar and A. E. Adams, "ACMENet: An underwater acoustic sensor network protocol for real-time environmental monitoring in coastal areas," *IEE Proc.-Radar, Sonar Navigat.*, vol. 153, no. 4, p. 365, 2006.

[10] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: Applications, advances and challenges," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 370, no. 1958, pp. 158–175, Jan. 2012.

[11] L. Liu, S. Zhou, and J.-H. Cui, "Prospects and problems of wireless communication for underwater sensor networks," *Wireless Commun. Mobile Comput., Underwater Sensor Netw., Archit. Protocols*, vol. 8, no. 8, pp. 977–994, 2008.

[12] D. Seidov, O. K. Baranova, M. Biddle, T. P. Boyer, D. R. Johnson, A. V. Mishonov, C. Paver, and M. Zweng, "The Greenland, Iceland and Norwegian seas regional climatology," 2013.

[13] J. Ahn, A. Syed, B. Krishnamachari, and J. Heidemann, "Design and analysis of a propagation delay tolerant Aloha protocol for underwater networks," *Ad Hoc Netw.*, vol. 9, no. 5, pp. 752–766, Jul. 2011.

[14] C. Li, Y. Xu, C. Xu, Z. An, B. Diao, and X. Li, "DTMAC: A delay tolerant MAC protocol for underwater wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4137–4146, Jun. 2016.

[15] S. Han, Y. Noh, U. Lee, and M. Gerla, "M-FAMA: A multi-session MAC protocol for reliable underwater acoustic streams," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 665–673.

[16] C. Li, Y. Xu, Q. Wang, B. Diao, Z. An, Z. Chen, and Z. Luo, "FDCA: A full-duplex collision avoidance MAC protocol for underwater acoustic networks," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4638–4647, Jun. 2016.

[17] C. Li, Y. Xu, B. Diao, Q. Wang, and Z. An, "DBR-MAC: A depth-based routing aware MAC protocol for data collection in underwater acoustic sensor networks," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3904–3913, May 2016.

[18] Y. Han and Y. Fei, "TARS: A traffic-adaptive receiver-synchronized MAC protocol for underwater sensor networks," *ACM Trans. Sensor Netw.*, vol. 13, no. 4, pp. 1–25, Nov. 2017.

[19] K. Hofer-Schmitz and B. Stojanovic, "Towards formal verification of IoT protocols: A review," *Comput. Netw.*, vol. 174, Jun. 2020, Art. no. 107233.

[20] Z. Brezocnik, B. Vlaovic, and A. Vreže, "SpinRCP: The Eclipse rich client platform integrated development environment for the Spin model checker," in *Proc. Int. SPIN Symp. Model Checking Softw.*, 2014, pp. 125–128.

[21] G. J. Holzmann, *The SPIN Model Checker: Primer and Reference Manual*, vol. 1003. Reading, MA, USA: Addison-Wesley Reading, 2004.

[22] M. Ben-Ari, *Principles of the Spin Model Checker*. Berlin, Germany: Springer, 2008.

[23] J. Smed and H. Hakonen, *Algorithms and Networking for Computer Games*, 2nd ed. Hoboken, NJ, USA: Wiley, 2017.

[24] J. Cohen, "Non-deterministic algorithms," *ACM Comput. Surv.*, vol. 11, no. 2, pp. 79–94, Jun. 1979.

[25] N. Morozs, P. D. Mitchell, Y. Zakharov, R. Mourya, Y. R. Petillot, T. Gibney, M. Dragone, B. Sherlock, J. A. Neasham, C. C. Tsimenidis, M. E. Sayed, A. C. McConnell, S. Aracri, and A. A. Stokes, "Robust TDA-MAC for practical underwater sensor network deployment: Lessons from USMART sea trials," in *Proc. 13th ACM Int. Conf. Underwater Netw. Syst.*, Dec. 2018, pp. 1–8.

[26] N. Morozs, P. D. Mitchell, and Y. Zakharov, "Dual-hop TDA-MAC and routing for underwater acoustic sensor networks," *IEEE J. Ocean. Eng.*, vol. 44, no. 4, pp. 865–880, Oct. 2019.

[27] F. Heidarian, J. Schmaltz, and F. Vaandrager, "Analysis of a clock synchronization protocol for wireless sensor networks," *Theoretical Comput. Sci.*, vol. 413, no. 1, pp. 87–105, 2012.

[28] F. A. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, "A collision-free graph coloring MAC protocol for underwater sensor networks," *IEEE Access*, vol. 7, pp. 39862–39878, 2019.

[29] F. Alfouzan, A. Shahrabi, S. Ghoreyshi, and T. Boutaleb, "A comparative performance evaluation of distributed collision-free MAC protocols for underwater sensor networks," in *Proc. 8th Int. Conf. Sensor Netw.*, 2019, pp. 85–93.

[30] F. A. Alfouzan, "Energy-efficient collision avoidance MAC protocols for underwater sensor networks: Survey and challenges," *J. Mar. Sci. Eng.*, vol. 9, no. 7, p. 741, Jul. 2021.

[31] F. A. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, "An energy-conserving collision-free MAC protocol for underwater sensor networks," *IEEE Access*, vol. 7, pp. 27155–27171, 2019.

[32] F. Alfouzan, A. Shahrabi, S. Ghoreyshi, and T. Boutaleb, "An efficient scalable scheduling MAC protocol for underwater sensor networks," *Sensors*, vol. 18, no. 9, p. 2806, Aug. 2018.

N. SURESH KUMAR received the master's degree in software engineering from the Cochin University of Science and Technology, Kochi, Kerala, India, in 2008. He is a Ph.D. Scholar with the Department of Computer Science, Cochin University of Science and Technology. He is also an Associate Professor with the Computer Science and Engineering Department, College of Engineering and Management, Punnapra, Alappuzha, Kerala. He has 22 years of teaching experience. His current research interests include formal modeling and verification of protocols in the IoT and cyber physical systems.

G. SANTHOSH KUMAR (Senior Member, IEEE) received the M.Tech. degree in computer and information science and the Ph.D. degree in wireless sensor networks from the Cochin University of Science and Technology (CUSAT), Kochi, Kerala. Currently, he is a Professor with the Department of Computer Science, Faculty of Technology, CUSAT. His areas of interests include cyber-physical systems, formal modeling and verification, computer vision, and NLP. He is a Senior Professional Member of IEEE and ACM.

SHAILESH SIVAN (Member, IEEE) received the B.Tech. degree in computer science and engineering from Kerala University, in 2012, the M.Tech. degree in computer and information science from the Cochin University of Science and Technology (CUSAT), in 2015, the M.Sc. degree in mathematics from Madras University, in 2018, and the Ph.D. degree from the Department of Computer Applications, CUSAT, in 2021. He is currently an Assistant Professor with the Department of Computer Science, CUSAT. He has six years of teaching and research experience and four years of relevant industrial experience as a senior software engineer in IT industry. He qualified UGC, NET/JRF, and GATE exams, for several times. His research interests include algorithms and theoretical computer science, artificial intelligence, computer vision and image processing, information retrieval and data mining, machine learning and deep learning, text mining, and natural language processing. He served as a resource person and a trainer for the different invited talks, technical sessions, and hands-on workshops at universities, colleges, IT industry, and government institutions in the domain of AI/ML/CV. He is a reviewer of international journals and conferences.

A. SREEKUMAR (Member, IEEE) received the Ph.D. degree from the Department of Computer Applications, Cochin University of Science and Technology (CUSAT), Kerala, in 2010. He is a Professor with the Department of Computer Applications, CUSAT. He has more than 29 years of teaching experience. He has published numerous research articles in peer-reviewed journals and conferences both nationally and internationally. His areas of interests include number theory, secret sharing, visual cryptography, cryptography, and compiler design.

• • •