A security system for spreading information about malicious behaviors in underwater wireless sensor networks

Filippo Donegà, Edoardo Bortolozzo, Roberto Francescon, Filippo Campagnaro, Michele Zorzi
Department of Information Engineering (DEI)
University of Padova, Italy
{filippo.donega, edoardo.bortolozzo}@studenti.unipd.it
{frances1, campagn1, zorzi}@dei.unipd.it

Abstract—**Despite being often used in mission critical scenarios, such as surveillance and coastal monitoring, Underwater Acoustic Networks (UANs) are inherently vulnerable to Denial of Service (DoS) attacks performed by malicious nodes. With enough knowledge about the communication protocol, reputation systems are proven to be effective on identifying misbehaviors in the network [1], [2]. However, while a node can assess the behavior of a neighbor node with high accuracy, it might have poor or no information about nodes deployed a few hops away. Moreover, given the disastrous consequences that DoS attacks may have on UANs, information about the presence of an intruder must be notified and spread across the network as quickly as possible, in order to permit countermeasures to be put in place and without impacting the performance of the UAN.**

In this paper we present `UWSentinel`, a transparent layer, i.e., independent of the reputation mechanism, added to the DESERT Underwater protocol stack to alert all nodes in the network of a possible threat, such as a malicious node performing a DoS attack. Results from simulations show that our protocol is lightweight and has little impact on the network throughput, while being comparatively fast in alerting all the nodes in the network, providing them with information about the possible misbehaving node.

Index Terms—**Security in Underwater Acoustic Networks, DESERT Underwater Framework, Reputation**

I. Introduction

The strong attenuation of electromagnetic fields underwater makes radio-frequency signals applicable only to very short range (up to few tens of centimeters) communication links, while optical communication can enable broadband communication up to a range of few tens of meters [3]. These communication technologies are therefore suitable only for some specific applications, such as docking stations or data retrieval with underwater unmanned vehicles, able to approach a sensor node and use the radio-frequency or the optical link only when in range. Acoustic signals, instead, can propagate up to a few kilometers and, despite the limited bandwidth, are nowadays the most widely used technology for underwater data transmissions.

UANs are often employed in critical applications such as disaster monitoring [4], [5], assisted navigation [6], military surveillance [7] and a number of other strategic tasks [8], [9] which usually require high data reliability and timely fault detection and recovery. These security requirements, while being crucial to the effectiveness and efficiency of the network, are difficult to satisfy in the underwater communication environment [10], [11]. This landscape is highly susceptible to active attacks aimed at injecting, destroying or altering data because of the low transmission speed and high ambient noise that characterize the channel [12]. For this reason, being promptly capable of recognizing the presence of malicious behaviors within the network is essential in order to guarantee a sufficient level of security. This task is indeed quite challenging, as solutions used in the terrestrial wireless domain cannot be directly applied to UANs due to the limited resources of the acoustic channel, characterized by low bandwidth and bitrate, long propagation delay and poor performance in the presence of multipath and acoustic noise [3].

General security schemes for underwater communication networks are only recently being implemented and tested, and a few proposals for security architecture standardization have also been proposed. Schemes for spreading information about potential threats have not been widely investigated, although some works tackle the problem and advance proposals. Among these, [13] proposes an implementation of security as a feature of the whole architecture rather than of a specific protocol. Their proposal focuses on the awareness of each node of the network of the behavior of the network itself. In addition, the authors in [1], [2] designed a security mechanism for UANs based on the introduction of a "watchdog" layer within each node of the network that overhears neighboring transmissions and keeps track of which ones are successful or still have to occur. This insight is then used to update a local table that stores reputation values for each surrounding node, which translate to one of four possible reputation states. Nodes whose reputation goes below a predefined threshold enter the "blacklist" state and are considered unreliable, while being able to redeem themselves by behaving correctly for at least some time. The method proved to be effective in counteracting sinkhole and

This work was supported in part by the FSE REACT EU, PON Research and Innovation 2014-2020 (DM 1062/2021) and by the Bundeswehr Technical Center for Ships and Naval Weapons, Maritime Technology and Research (WTD 71), Eckernförde, Germany, under the contract E/E71S/K1291/CF081.

resource exhaustion attacks, preventing the affected nodes from being completely excluded from the network, but does not exploit cooperation between nodes (i.e., only neighbors of a malicious node end up updating their reputation tables accordingly, keeping this knowledge for themselves).

This work expands on the described reputation mechanism (while being easily adaptable to any other reputation scheme) by proposing and implementing an alarm system that spreads information about malicious behaviors throughout the whole network, allowing for quick enactment of countermeasures such as network-level re-routing to exclude the problematic nodes from any data path. Our implementation consists of a module (from now on, `UWSentinel`) for DESERT Underwater [14], a publicly available [15] underwater network simulator developed and maintained by the SIGNET group at the University of Padova.

The rest of the paper is structured as follows. Section II reports a detailed description of `UWSentinel`'s design, while in Section III we describe the simulated topology and the simulation settings. Section IV reports and comments on the gathered results obtained in simulation and finally, Section V draws the conclusions and offers some considerations about future work.

II. Secure Underwater Protocol Stack

The DESERT Underwater Framework [14] is a complete suite of protocols for underwater acoustic and optical networks organized in a layered structure similar to the Internet Protocol (IP) suite, with some modifications regarding overhead sizes and packet headers structure that better suit the particular characteristics and efficiency requirements of the underwater environment. For example, while in the Internet billions of devices are interconnected, a real UAN usually counts no more than few tens of nodes. This allows the use of a shorter addressing mechanism and a lighter packet header. The protocol stack that is typically used is composed by 5 layers, namely:

- one or more application layers, that generate the data intended for a final destination;
- a lightweight UDP-like transport layer, that forwards the received packets to the correct application;
- a routing layer that establishes the path to be followed by the packets;
- a Medium Access Control (MAC) protocol responsible for channel access;
- a physical layer.

In simulation, the physical layer is where the bit error rate is computed, based on models of path loss, noise and interference. In contrast, in sea trials the simulated physical layer is substituted with real devices (i.e., acoustic modems) able to transmit and receive data through the acoustic channel, thereby including real-world behaviors.

In order to include security features, the DESERT protocol stack is extended by including some additional layers, `UWSentinel` and `UWWatchdog`, able to communicate with each other via cross-layer messages.

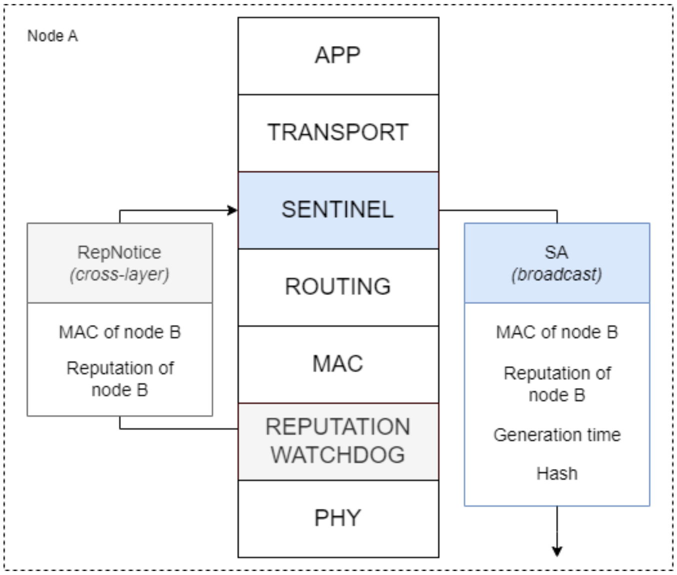

Fig. 1. Example of alarm generation: node A blacklists node B and alerts the network.

The `UWSentinel` module is placed between the transport and routing layers of the protocol stack to allow for broadcasting of alarm packets when malicious activity is detected, while being completely transparent to all other kinds of packets. `UWSentinel` is designed to be compatible with any reputation system, as long as it is equipped with the capability of sending cross-layer messages.

`UWWatchdog`, first introduced in [1], is a transparent layer placed between the MAC and the physical layers to overhear neighboring transmissions and keep track of which nodes are misbehaving, creating a reputation table. Nodes with a bad reputation are blacklisted and considered unreliable. In order to exclude such nodes from the network, `UWSentinel` spreads knowledge about their blacklisting to all nodes in the UAN. In fact, in our implementation, `UWSentinel`'s action is always triggered when a node gets blacklisted by the reputation system, although in future works a more advanced filtering criterion could be applied to the reputation value of the cross-layer message to decide whether or not to send an alarm. Figure 1 shows the protocol stack and the packets involved in the process of node A reporting that node B has been blacklisted.

A. Alarm generation

Alarm generation on node A is triggered when a cross-layer message (from here on, `RepNotice`) containing B's MAC address and current reputation value is received from the underlying reputation module. At this point, a `Sentinel Alarm (SA)` is generated and broadcast through the network to inform all other nodes about the event. An `SA` contains the following fields:

1) *hash_ID* - computed at the moment the packet is generated in order to uniquely identify it over multiple hops. It depends on both generation time and MAC address;

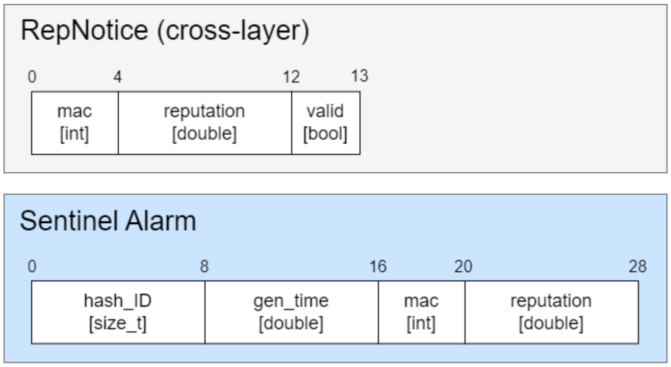

Fig. 2. RepNotice and Sentinel Alarm packet structure. Field sizes are reported in Bytes.

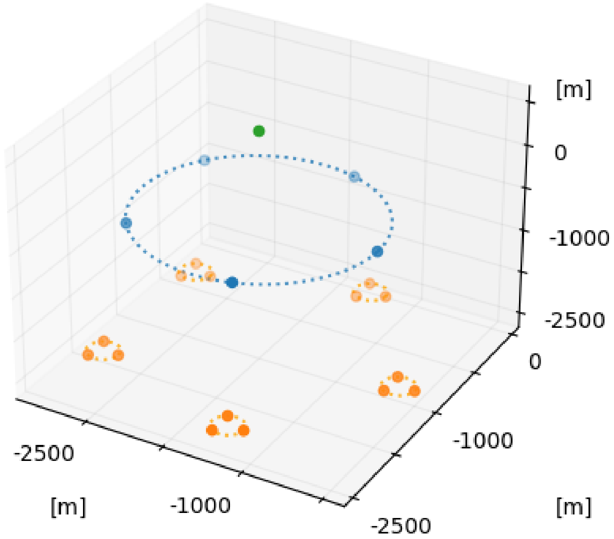

Fig. 3. Tree-like topology used for running simulations. Orange nodes (*sources*) generate data, while blue nodes (*relays*) forward data towards the green node (*sink*).

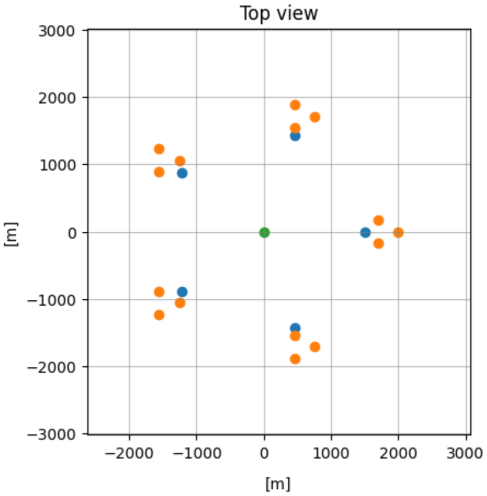

Fig. 4. 2D top view of the tree-like topology used for running simulations.

2) *gen_time_* - the generation time of the packet;
3) *mac* - the MAC address of the misbehaving node;
4) *reputation* - the reputation of the misbehaving node for possible filtering.

The RepNotice and SA packet structures are depicted in Figure 2.

B. Alarm forwarding

When node C receives the SA, it computes its hash code and attempts packet validation. If the process is successful, the hash is searched in an internal table that stores unique copies of previously received alarm packets along with the number of times they were collected. If the hash was previously seen, the SA packet is dropped and its counter incremented. Otherwise, it is added to the table and re-broadcast in the network.

By not forwarding the same alarm more than once, UWSentinel manages to alert every node in the network without overloading it, thus preventing additional destructive interference which is already a big downside of the underwater acoustic medium. Under such premises, in a connected network where any two partitions are linked by at least one correctly functioning link (i.e., two nodes that can hear each other and behave well), the SA will eventually reach all nodes in the network.

C. The watchlist mechanism

UWSentinel can be tuned by specifying which nodes are to be reported on, adding them to its watchlist. This prevents node A from decreasing the reputation of node B if its overhearing range is not long enough to check whether or not B's transmissions are successful. It is a sort of neighbor list that can be either set in advance at deployment time given the predefined knowledge of the network topology, or built by the network protocol: discussion on how the watchlist is created is out of the scope of this paper and left as future work.

III. Topology and Simulation Setup

We proved our module to be effective and lightweight through a series of DESERT simulations. We evaluate the system over a UAN architecture with the tree-like topology depicted in Figures 3 and 4. This topology is purposefully designed to verify the correct propagation of SAs and test the limits of UWSentinel. Three types of nodes are defined:

- *sources* (depicted in orange in Figures 3 and 4), generating data on the seafloor at a depth of 2500 m;
- *relays* (depicted in blue), situated at a depth of 1000 m and forwarding *source* data to the surface;
- a *sink* (depicted in green) collecting data at sea level.

Each *source* was set to only have the nearest *relay* in its watchlist. Nodes were modeled on the EvoLogics S2C R 18/34 D [16] Underwater Acoustic Modem, operating in the 18-34 kHz frequency band, and distances were carefully chosen to prevent *sources* from being able to reach any other *relay* except their nearest one, while allowing communication

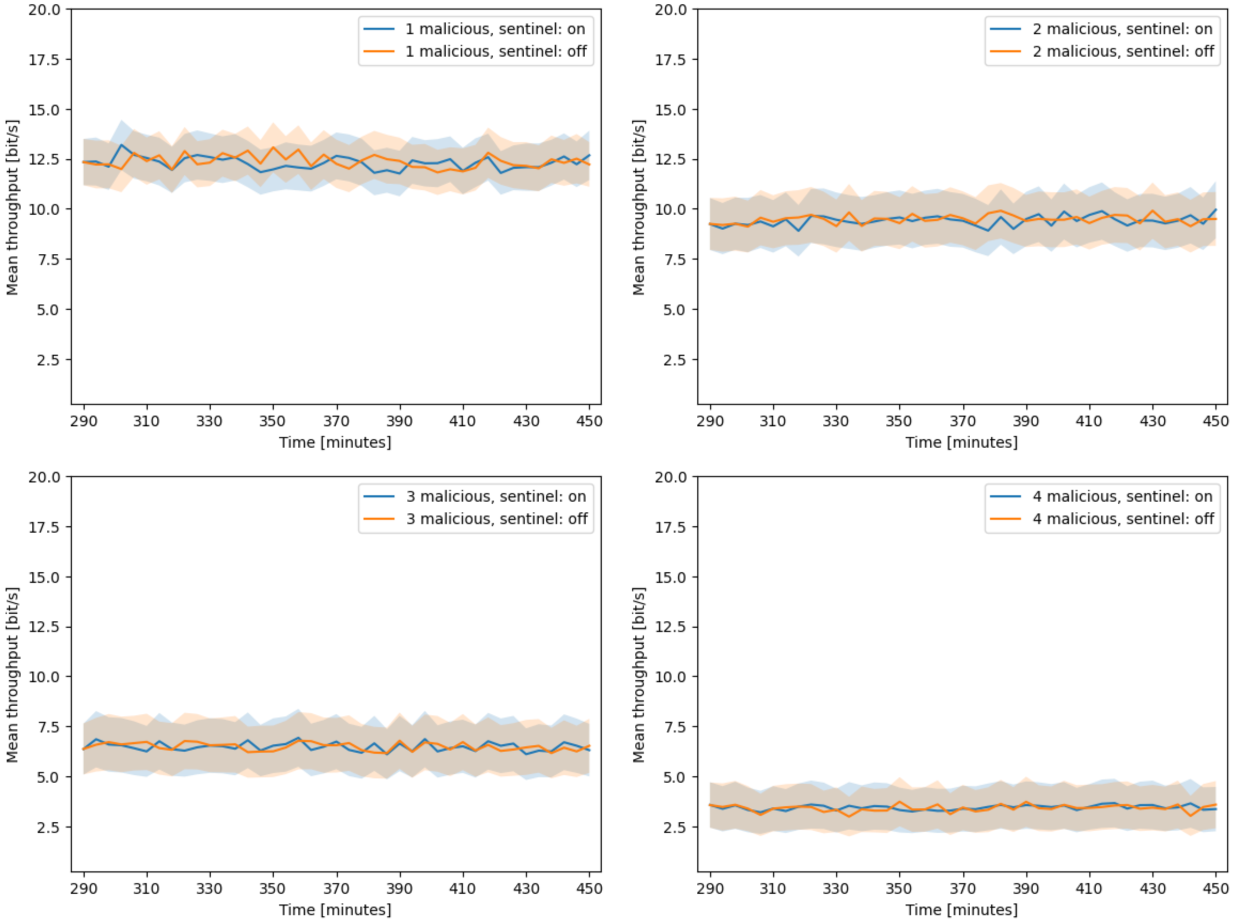

Fig. 5. Mean throughput (solid lines) to the *sink* measured over intervals of 1200 s with respectively 1, 2, 3 and 4 *relays* simultaneously deactivated at time 15000 s after reputation establishment. The semi-transparent area indicates the 95% Confidence Interval for the mean. For each graph, both scenarios with UWSentinel on (in blue) and off (in orange) are depicted for visual comparison.

between *sources* that belong to adjacent clusters. The node-to-node maximum communication range was found to be of approximately 2000 m, with a Packet Delivery Ratio (PDR) of 90% decreasing to 60% at a distance of 2100 m.

To achieve a more realistic and meaningful simulation environment, ambient noise was introduced through wind and shipping activity models, as described in [17]. Specifically, the shipping factor was set to 1, while the wind speed to 10 m/s. Furthermore, wave propagation was set to circular due to the depth of the topology. Malicious activity was simulated by simultaneously deactivating a certain number of *relays* after some time to ensure reputation establishment, causing the underlying *sources* to generate and broadcast an alarm packet each. These SAs would then travel across the lower ring up until finding a viable route to the *sink*, alerting all other nodes in the process.

100000 s of deployment were simulated, with each *source* generating a Constant Bitrate (CBR) traffic with fixed packet size of 180 B and a fixed period of 80 s. The acoustic bitrate was set to 2500 bps. All main simulation parameters are summarized in Table I.

TABLE I
SIMULATION PARAMETERS

Simulation duration	100000 s
CBR period	80 s
Packet size	180 B
Bitrate	2500 bps
Transmission power	175 dB re µPa
Central frequency	26 kHz
Bandwidth	16 kHz
Shipping activity	1
Wind speed	10 m/s
Propagation geometry	circular

IV. Results

Simulations were aimed at testing UWSentinel's capability of efficiently alerting as many nodes as possible, with the *sink* being the most important node to be alerted. To obtain more statistically accurate results, 10 runs for each scenario were simulated. Scenarios differ by the number of malicious *relays* (varying from 1 to 4) and the state of UWSentinel

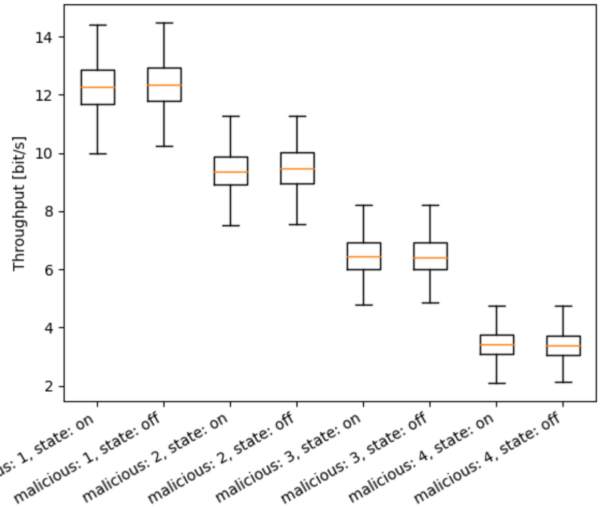

Fig. 6. Box plot of the throughput to the *sink* from *relay* deactivation at 15000 s to the end of the simulation for any combination of number of malicious nodes and state (on/off) of `UWSentinel`. Orange horizontal lines represent the throughput median across the whole simulation, bottom and top of the box indicate the first and third quartile respectively, and whiskers depict minimum and maximum sampled value.

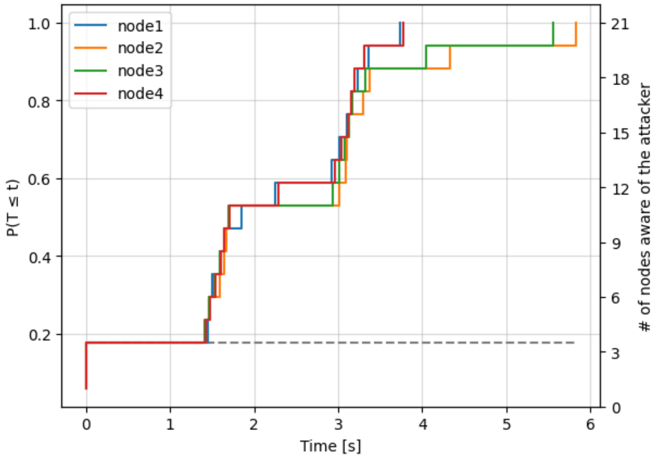

Fig. 7. Probability that a generic node in the network becomes aware of a misbehaving node before the time on the x-axis in the scenario with 4 malicious *relays* and `UWSentinel` activated. The right y-axis represents the number of nodes aware of the malicious node, while the dotted grey line indicates the probability that a node is alerted of a misbehaving *relay* different from the one in its `watchlist`, in case `UWSentinel` is deactivated.

(on and off). Any result presented in the rest of this section is thus obtained for each scenario by averaging on all runs.

A. *Impact on network throughput*

We proved that `UWSentinel` has little to no impact on the network throughput by plotting the mean CBR throughput in bit/s against time, starting from *relay* deactivation until the end of the simulation, with `UWSentinel` on and off for each possible number of malicious *relays* (Figure 5). Solid lines indicate the mean throughput measured over intervals of 1200 s while semi-transparent areas show the 95% Confidence Interval (CI) for the mean. For each scenario, both the mean throughput and its CI mostly overlap, showing no relevant difference between the two cases with `UWSentinel` on and off. This result was further confirmed by comparing the number of `SA`s received by each *source* in the network with the overall number of packets generated by the CBR modules throughout the whole simulation. In the worst-case scenario, with 4 *relays* being deactivated at time 15000 s, each *source* received 12 `SA`s replicated a maximum of 6 times each while 18750 packets were generated by CBR. Because of the large difference between such amounts, it was expected that our module did not impact the network throughput significantly.

Figure 6 shows the same results differently, where for each scenario the orange horizontal bar represents the throughput median across the whole simulation, bottom and top of the box indicate the first and third quartile respectively, and whiskers depict minimum and maximum sampled value. Again, no significant difference in data traffic can be noticed between having `UWSentinel` active or not. Given that tests were performed under heavy traffic conditions (maximizing throughput while keeping the mean PDR over 80%) we can infer that `UWSentinel`'s ability to not impact on network traffic would extend to most scenarios.

B. *Speed of `SA` propagation*

Besides proving `UWSentinel`'s transparency towards the network throughput, we also tested our module's efficiency in quickly alerting all nodes in the network when malicious activity is detected. In particular, we measured how quickly nodes in the network can be notified about the presence of a possible malicious node by the protocol's `SA` packets. Figure 7 depicts a node-averaged Cumulative Distribution Function (CDF), defined as:

$$F_i(t) = \sum_{j\in \boldsymbol{N},\ j\neq i} \frac{P(T_{ij} \leq t)}{|\boldsymbol{N}|-1}, \quad i \in \boldsymbol{A}. \tag{1}$$

The random variable T_{ij} is the time from when any node $\neq i$ becomes aware of the presence of malicious node i through overhearing and sends the first `SA` packet, to when node j receives such notification. $\boldsymbol{N}$ is the total set of nodes in the network, while $\boldsymbol{A} \subset \boldsymbol{N}$ is the subset of malicious nodes. The grey dashed horizontal line represents the percentage of nodes in the network with this knowledge when `UWSentinel` is not used. In this case only few nodes close to the malicious one know of its existence and are informed of its behavior by the reputation system, which means that, independently of time, if a random node is chosen, it will be always made aware with fixed probability

$$P = \frac{\#\ nodes\ in\ range}{|\boldsymbol{N}|}. \tag{2}$$

We point out that the average propagation time is around 1 s between *sources* and *relays* and around 2 s between two groups of *sources*. The plot in Figure 7 shows that using `UWSentinel` all the nodes in the network (including the *sink*) are made aware of all the misbehaving nodes and their MAC addresses in less than 6 seconds. Such a short diffusion time is likely due to the lightness of the `SA` packets that despite the multi hop topology of the network are able to reach the *sink* quickly. Furthermore, as depicted through the grey dotted line, without using `UWSentinel` just ∼20% of the nodes become aware of a malicious *relay*, i.e., the three *sources* belonging to the same cluster and the network *sink*.

C. Discussion

The negligible requirements of the `UWSentinel` protocol in terms of traffic and the rapidity of the alert messages in reaching all the nodes in the network set the ground for the development of algorithms that allow nodes to share trust evaluations, and to combine these values with their own evaluations. This is especially important for networks with a sparse node distribution, in the presence of mobile nodes or in the case nodes deployed in a certain area want to know the reputation of nodes deployed in a different area of the network. In fact, as presented in Figure 7, in the scenario considered in this paper only the three nodes close to the malicious *relay* are able to discover its presence if the `UWSentinel` mechanism is not available. While this first step already provides a valuable solution to detect anomalies, a complete distributed trust mechanism should be built in order to assess whether a mobile node can be trusted or not. The integration of trust evaluations performed by the node itself, usually called *first-hand evaluations*, and evaluations received from other nodes, called *second-hand evaluations*, is a topic of research that is currently under active investigation [18]. The first thing to consider is to avoid blindly accepting all second-hand evaluations, as this would allow an easy attack on the trust system; the second thing is how the second-hand evaluations are integrated: their value is usually lower than the first hand ones but they may be helpful in making decisions in uncertain scenarios or in case of lack of information. Taking the mean value would be the first step, but it is usually better to weigh the values according to how much the node providing second-hand evaluations is trusted.

V. Conclusions

In this work we built on a deployed reputation mechanism, to provide an underwater acoustic communication network with a protocol to spread reputation information among the nodes of the network. Our solution, called `UWSentinel`, is a protocol stack layer that, receiving information from the reputation layer via cross-layer messages, crafts a packet able to alert all the nodes in the network about a node acting in a non-compliant way. This layer relies on information provided by a reputation layer, thus being transparent to the particular reputation evaluation mechanism deployed. Furthermore, the resulting packets generated and forwarded by the network, called `Sentinel Alarms`, are lightweight and have little impact on the throughput of the network. Finally, in the configuration of our simulations, where the propagation time between *sources* and *relays* is around 1 s, the efficiency of the reputation data diffusion allows to alert all the nodes in the network in less than 6 s. We point out that our proposed solution is built with the aim of being general enough to adapt to most protocols and algorithms, thus, future work will focus on a decision policy that, making use of the propagated `Sentinel Alarms`, is able to make effective decisions to exclude suspect nodes or recover network connectivity.

References

[1] D. Tronchin, R. Francescon, F. Campagnaro, A. Signori, R. Petroccia, K. Pelekanakis, P. Paglierani, J. Alves, and M. Zorzi, "A secure cross-layer communication stack for underwater acoustic networks," in *IEEE OCEANS 2021*, San Diego – Porto, San Diego, CA, USA, Sep. 2021.

[2] A. Signori, F. Campagnaro, I. Nissen, and M. Zorzi, "Channel-based trust model for security in underwater acoustic networks," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 479–20 491, Oct. 2022.

[3] A. Pal, F. Campagnaro, K. Ahraf, R. Rahman, A. Ashok and H. Guo, "Communication for underwater sensor networks: A comprehensive summary," *ACM Transactions on Sensor Networks (TOSN)*, vol. 19, no. 1, pp. 1–44, Feb. 2023.

[4] K. Casey, A. Lim, and G. Dozier, "A sensor network architecture for tsunami detection and response," *International Journal of Distributed Sensor Networks*, vol. 4, no. 1, pp. 27–42, Jan. 2008.

[5] P. Kumar, P. Kumar, and P. Priyadarshini, "Underwater acoustic sensor network for early warning generation," in *IEEE Oceans*, Hampton Roads, VA, USA, Oct. 2012.

[6] M. Waldmeyer, H.-P. Tan, and W. K. Seah, "Multi-stage AUV-aided localization for underwater wireless sensor networks," in *IEEE Workshops of International Conference on Advanced Information Networking and Applications*, Biopolis, Singapore, Mar. 2011, pp. 908–913.

[7] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless sensor networks for underwater surveillance systems," *Ad Hoc Networks*, vol. 4, no. 4, pp. 431–446, Jul. 2006.

[8] E. Felemban, F. K. Shaikh, U. M. Qureshi, A. A. Sheikh, and S. B. Qaisar, "Underwater sensor network applications: a comprehensive survey," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, Nov. 2015.

[9] M. Murad, A. A. Sheikh, M. A. Manzoor, E. Felemban, and S. Qaisar, "A survey on current underwater acoustic sensor network applications," *International Journal of Computer Theory and Engineering*, vol. 7, no. 1, Feb. 2015.

[10] G. Yang, L. Dai, and Z. Wei, "Challenges, threats, security issues and new trends of underwater wireless sensor networks," *Sensors*, vol. 18, no. 11, p. 3907, Nov. 2018.

[11] S. Jiang, "On securing underwater acoustic networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 719–752, Firstquarter 2019.

[12] A. Stefanov and M. Stojanovic, "Design and Performance Analysis of Underwater Acoustic Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 10, pp. 2012–2021, Dec. 2011.

[13] C. Lal, R. Petroccia, K. Pelekanakis, M. Conti, and J. Alves, "Toward the Development of Secure Underwater Acoustic Networks," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 1075–1087, Oct. 2017.

[14] F. Campagnaro, R. Francescon, F. Guerra, F. Favaro, P. Casari, R. Diamant, and M. Zorzi, "The DESERT underwater framework v2: Improved capabilities and extension tools," in *Proc. Ucomms*, Lerici, Italy, Sep. 2016.

[15] "DESERT: Design, simulate, emulate and realize test-beds for underwater network protocols," Last time accessed: Jan. 2024. [Online]. Available: https://desert-underwater.dei.unipd.it/

[16] "EvoLogics S2C R 18/34D Underwater Acoustic Modem," Last time accessed: Jan. 2024. [Online]. Available: https://evologics.de/product/s2c-r-18-34d-27

[17] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 34–43, Oct. 2007.

[18] S. Chinnaswamy and K. Annapurani, "Trust aggregation authentication protocol using machine learning for IoT wireless sensor networks," *Computers & Electrical Engineering*, vol. 91, p. 107130, Mar. 2021.