

PRObust: A Percolation-Based Robustness Optimization Model for Underwater Acoustic Sensor Networks

Zhaowei Zhang, Chunfeng Liu, Wenyu Qu, Zhao Zhao, and Weisi Guo, *Senior Member, IEEE*

Abstract—In Underwater Acoustic Sensor Networks (UASNs), the robustness of network is greatly affected by complex marine environments when implementing multi-hop data transmission. Factors such as the underwater acoustic channel and dynamic topological changes induced by multi-layered oceanic vortices exacerbate this influence. However, there is currently a research gap in the specific area of robustness optimization for UASN. Existing studies on robustness optimization are unsuitable for UASN as they neglect the considerations of the marine environment and node characteristics (e.g., residual energy). In this work, we propose PRObust, a percolation-based robustness optimization model for UASN. PRObust consists of two distinct phases: percolation modeling and bottleneck optimization. In the percolation modeling phase, we incorporate both node and edge features, considering the physical and topological properties, and introduce a novel approach for calculating link quality. In the bottleneck optimization phase, we devise a graph theory-based method to identify bottlenecks, leveraging the flow information recorded by nodes to improve the accuracy of bottleneck discovery. Moreover, we integrated time slots and a current movement model into the proposed model, allowing its applicability to dynamically changing UASN. Extensive simulation results demonstrate that PRObust, after optimizing the bottlenecks, significantly enhances both network robustness and performance with minimal overhead compared to other existing methods.

Index Terms—Underwater Acoustic Sensor Networks, Robustness Optimization, Percolation, Current Movement, Data Transmission, Network Traffic.

I. INTRODUCTION

UNDERWATER Acoustic Sensor Networks (UASNs) consist of nodes equipped with underwater communication and computational capabilities. With advantages of flexibility and scalability, UASN has been widely applied in various domains such as underwater surveillance, target detection, and environmental monitoring [1]. This facilitates comprehensive monitoring and research in the marine environment. However, the complex and unpredictable nature of the marine environment poses significant challenges to data transmission within

UASN, leading to a decline in network robustness [2]. As a result, ensuring the robustness of UASN becomes imperative for the provision of network services without interruptions.

The robustness of UASN faces significant challenges due to the dynamic nature of the marine environment and the unique characteristics of underwater acoustic channels. These challenges arise from a fundamental contradiction. On one hand, the dynamic variations, interference from oceanic noise, and limitations of underwater acoustic channels impose restrictions on the transmission capacity and processing capabilities of links in UASN [3]. On the other hand, UASN, as a typical convergent network, requires the aggregation of data collected by all nodes to the sink node, resulting in uneven traffic distribution across different links. When the processing capacity of a link constrains the flow of network traffic, consecutively transmitted data packets accumulate at sensor nodes that are unable to forward them. Although such abnormal states may not necessarily cause interruptions, the data packets will be unable to reach the sink node, thereby impeding the network's data collection capability and diminishing overall network robustness. Efforts aimed at minimizing the impact on traffic reaching the sink node, caused by limitations in the processing capacity of network links, represent a compelling endeavor in enhancing the robustness of UASN.

Based on the background discussed above, the concept of network robustness in this work pertains to the network's ability to maintain traffic delivery to the sink node under various environmental conditions [4]. Bottlenecks are defined as the links whose removal minimizes the traffic reaching the sink node to the greatest extent within the network [5]. In recent years, percolation theory has gained significant attention in different network domains due to its capacity to analyze network dynamics. By utilizing link quality as a foundation for network modeling, which reflects the processing capacity of the individual links, percolation theory enables the development of models that capture both topological and physical characteristics of real-world networks. Researchers have discovered that the removal of a small fraction of links can lead to abrupt changes in the overall network performance [6]. Through the optimization of a limited number of links that are susceptible to causing abrupt changes in network robustness, the overall network performance can be significantly improved.

Although a significant amount of research has been done on optimizing network robustness in scenarios such as complex networks [7] and urban transportation networks [8], these

This research was supported in part by the National Natural Science Foundation of China-Guangdong Joint Fund under Grant No. U1701263, the National Natural Science Foundation of China (NSFC) under grant No. 61871286 and No. 61672131. (Corresponding author: Wenyu Qu, E-mail: wenyu.qu@tju.edu.cn.)

Zhaowei Zhang, Chunfeng Liu, Wenyu Qu, and Zhao Zhao are with College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with Tianjin Key Laboratory of Advance Network Technology and Application, Tianjin, China.

Weisi Guo is with the School of Aerospace, Transport, and Manufacturing, Cranfield University, MK43 0AL Cranfield, U.K., and also with The Alan Turing Institute, NW1 2DB London, U.K. (e-mail: weisi.guo@cranfield.ac.uk).

Manuscript received X XX, XXXX; revised X XX, XXXX.

studies can not directly apply to UASN. To establish a systematic framework for optimizing UASN robustness, we need to address the following challenges:

- 1) **The limitation of the acoustic channel.** In UASN, data packets are transmitted through acoustic channels characterized by substantial attenuation and high delay. Moreover, the presence of environmental noise poses a challenge to ensuring reliable data transmission [9].
- 2) **The limitation of the resources.** It is challenging to provide sufficient energy to nodes in UASN. Performing intensive operations and large-scale computations on nodes significantly increase energy consumption, posing a threat to the network's lifecycle.
- 3) **The dynamic changes of UASN.** Factors such as changes in the marine environment and the displacement of moored nodes driven by currents easily lead to variations in the communication environment and network topology [10].
- 4) **The attributes of nodes and edges.** Different from other networks, UASN involves nodes and edges with distinct attributes, requiring the integration of both node and edge characteristics to capture the network's features.

To overcome the aforementioned limitations, we employ percolation theory as a modeling tool for UASN. We introduce the concept of link quality to assess the processing capacity of individual links. Additionally, we propose a novel approach for optimizing bottlenecks based on the percolation model, aiming to optimize these links and enhance the network's robustness. The proposed model incorporates the characteristics of traffic imbalance observed in real-world networks, as well as the specific physical attributes of UASN. Furthermore, our model is theoretically applicable to UASN employing any routing strategy as it directly utilizes the routing information of the network. The main contributions of this work can be summarized as follows:

- 1) In this work, we propose PRobust, a percolation theory-based robust optimization model, to improve data transmission and enhance network robustness in UASN. In order to tackle the dynamic environments in UASN, we integrate time slots and a current movement model into the proposed model. To the best of our knowledge, this work is the first research in the UASN that specifically focuses on network robustness optimization.
- 2) We introduce a novel link quality metric that combines node quality and communication quality, capturing the characteristics of both nodes and edges in the network. Additionally, we take into account the specific attributes of underwater acoustic channels in UASN and the dynamic nature of the network, ensuring a comprehensive representation of the network's physical and topological features.
- 3) A novel approach for optimizing bottlenecks using the PRobust model is presented, with the goal of enhancing network robustness through optimizing special links. This method accounts for the scenario where multiple links may have the same link quality. To address this challenge, the information recorded by nodes in multi-

hop routing is introduced, enabling a more detailed evaluation of the impact of different links on data transmission. This enhancement improves the accuracy of identifying bottlenecks with identical link quality.

- 4) Through extensive simulations, we conducted a performance evaluation of PRobust in terms of maintaining data transmission under diverse network conditions and assessing its robustness across different network scales. The results demonstrate that PRobust outperforms other state-of-the-art methods by significantly enhancing network robustness. Additionally, PRobust exhibits remarkable improvements in key network performance metrics, including packet delivery ratio, delay, and throughput.

The rest of this paper is organized as follows. In Sec. II, we conduct a literature review and discuss existing robust optimization methods. Sec. III presents the mobile model, acoustic communication model, and percolation model. Sec. IV discusses the design of the PRobust model, while Sec. V focuses on simulation and performance analysis. Finally, the conclusion, discussion, and future work are presented in Sec. VI.

II. RELATED WORK

In the field of UASN, robustness has been a long-standing concern, but existing studies in this area are mainly focused on considering robustness factors during node deployment and protocol design. To the best of our knowledge, there is currently no dedicated research specifically addressing the optimization of UASN robustness, making our work the first article to target UASN robustness optimization. Considering the limited literature on robustness optimization in the UASN field, we also investigate the latest research on robustness optimization in other related areas such as wireless sensor networks, complex networks, and urban transportation networks. In this section, a concise summary is presented for three specific directions of robustness optimization in various network domains: topology optimization, topology control, and link optimization.

A. Topology Optimization

Various techniques such as heuristic algorithms and reinforcement learning can be utilized to construct an optimal network topology in order to maximize robustness metrics such as the maximum connected component and rigidity features under the current environmental conditions.

Given the constraints of limited energy and the importance of network robustness, Zhao et al. proposed a dynamic routing algorithm based on Q-learning and introduced an optimal rigid graph approach utilizing potential games to balance energy consumption and network robustness [11]. The rigidity features were employed to assess the stability of the current network, with larger values indicating higher robustness. In a related study, Guang et al. presented the energy-balanced robust topology SFRG, which leveraged scale-free networks and rigid topology [12]. They constructed an optimal rigid graph topology structure within the designated area and treated the complete rigid graph as nodes to establish a scale-free network

to enhance the network's robustness. Nevertheless, the topology optimization techniques proposed in the aforementioned works often neglected the traffic demands of real network scenarios, relying solely on topology features to evaluate network robustness, which may not necessarily improve the network's service capacity. Furthermore, the dynamic and intricate nature of underwater environments necessitates frequent computation and adjustment of node topology whenever environmental changes occur, potentially leading to a compromised overall network lifecycle.

B. Topology Control

Topology control mechanisms exert control over nodes to enhance the network's robustness by employing strategies such as power control, sleep scheduling. This type of mechanism aims to construct a highly robust network while ensuring network connectivity.

Du et al. proposed a non-cooperative game-theoretic topology control model for Wireless Sensor Networks (WSNs) [13]. The main objective of their work is to develop an energy-efficient and fault-tolerant topology control algorithm that maintains k -connectivity in the network. This approach enhances network robustness and reliability while appropriately managing the network's lifespan. In another study, Fan et al. introduced a deep reinforcement learning framework called FINDER, which leverages training on small-scale complex networks to identify a set of critical participants, whose activation or removal maximizes or minimizes the network's connected components [14]. FINDER exhibits exceptional performance in efficiently and effectively identifying critical participants compared to existing methods. However, both approaches fail to consider dynamic network changes, which would significantly increase computational overhead on the nodes. The execution of frequent complex computations on underwater acoustic sensor nodes poses significant challenges. Additionally, large-scale topology control methods lead to increased energy consumption, contradicting the limited energy characteristics of UASN.

C. Link Optimization

Through the manipulation of network edges and the utilization of frameworks such as percolation theory, the identification and optimization of bottlenecks within the network are accomplished, leading to the ultimate enhancement of the network's robustness.

Li et al. conducted a study on urban traffic networks and discovered that the overall network consists of dynamically formed local traffic clusters interconnected by bottlenecks [15]. It is observed that targeted improvements on key bottlenecks at a smaller scale can have a significant impact on enhancing global traffic conditions. The research utilized percolation models to represent the urban traffic network and proposed a method based on the percolation threshold to identify bottlenecks. Experimental findings indicate that addressing bottlenecks identified through the percolation threshold offers a cost-effective strategy for substantially improving the overall network traffic. Hamedmoghadam et al. observed that traffic

demand experiences non-uniform fluctuations in networks and is often hindered by congestion or overload [16]. To address this issue, they introduced a percolation-based network analysis framework that theoretically identifies bottlenecks exerting a decisive influence on traffic flow within the network. Through experiments conducted on large-scale real traffic networks, they demonstrated that mitigating congestion on a small fraction of identified bottlenecks can lead to significant network enhancements. However, it is important to note that the aforementioned studies primarily focused on the attributes of network edges while neglecting the impact of node attributes on network performance. Moreover, considering the dynamic underwater environment in UASN, it is necessary to develop a comprehensive framework capable of analyzing network robustness in the context of dynamically changing networks.

D. Discussion and Innovation

The design of network robustness optimization for different scenarios has shown diverse trends in the literature. However, as far as we know, there is a current research gap in addressing the specific optimization of robustness in UASN. Therefore, it is advocated to develop a dedicated framework for optimizing robustness, specifically designed for UASN. In light of the existing challenges identified in related research, including the lack of consideration for UASN's underwater environment characteristics, network dynamics, and limited node energy, we propose a percolation model framework that is well-suited for UASN networks. To address the challenges mentioned above, corresponding solutions are proposed as follows:

- 1) To overcome the challenges arising from the underwater environment in UASN, a percolation model PRobust has been developed, explicitly tailored to accommodate the unique characteristics of UASN. This refined model comprehensively incorporates the physical and topological aspects of UASN while introducing weight coefficients to effectively balance the significance of node attributes and edge attributes.
- 2) To tackle the issue of dynamic changes in UASN, we adopt network segmentation. Within each time slot, the network is analyzed as a static entity, taking into consideration node mobility and environmental fluctuations between different time slots. This approach enables the extension of the analysis from a static network to a dynamic network.
- 3) To mitigate the limited energy problem in UASN, it is essential to minimize energy consumption while optimizing the network's robustness. Accordingly, we adopt a strategy of selectively optimizing and protecting a small subset of bottlenecks within UASN to enhance the overall network robustness. This approach circumvents the energy expenditure incurred by manipulating a large number of nodes.

III. SYSTEM MODEL

A. Network Model And Assumptions

In this work, the considered UASN comprises a sink node s and a set of sensor nodes S_n . As shown in Fig. 1, the base

station is deployed on land, the sink node s is deployed on the water surface, and the sensor nodes $n_i \in S_n$ are randomly deployed underwater. Each node n_i can sense ocean data over time to generate data packets, which are transmitted via acoustic communication to nodes within the communication range and via multi-hop routing to nodes outside the communication range, and ultimately transmitted to the sink node.

From the perspective of graph theory, the UASN can be represented as a directed graph $G(V, E)$, where $V = S_n \cup s$ represents the set of nodes and $E = \{(i, j) | v_i \in S_n, v_j \in S_n \cup s, v_i \neq v_j, d_{ij} \leq d_r\}$ represents the set of directed links between nodes. Specifically, (i, j) denotes a directed link from node v_i to v_j , where v_i is a sensor node in S_n and v_j belongs to either S_n or s . Moreover, d_{ij} denotes the distance between node v_i and v_j , while d_r represents the maximum communication range of the sensor nodes.

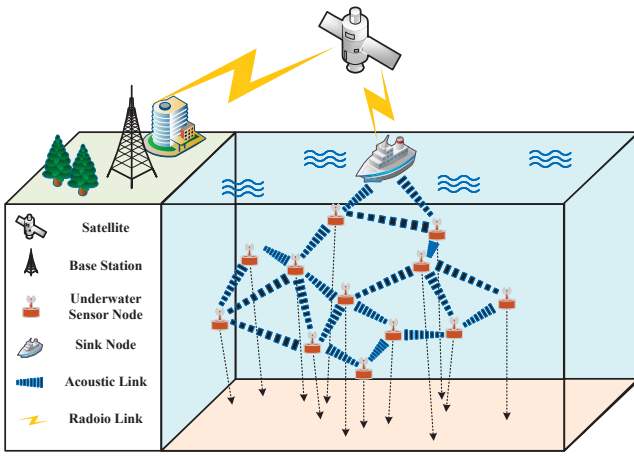


Fig. 1. The UASN system model

To ensure the data transmission between itself and neighboring nodes, each node requires a buffer to store data packets that have not yet been sent due to the limited capacity of underwater acoustic communication. The sink node s transfers the collected ocean information to the base station for further analysis and processing.

Given that the sink node is deployed on the water surface, allowing for easy interaction and energy replenishment, it is assumed that its communication, storage, and energy capabilities are unlimited. In contrast, as the sensor nodes are deployed at random locations in the marine environment, direct interaction with them becomes challenging. Therefore, it is assumed that all sensor nodes have limited communication, storage, and energy capabilities. As a result, it becomes crucial to consider the limited processing capacity of links since the processing capacity of a link can constrain data transmission. Unsent data packets will accumulate in the buffer of the current node, impeding further transmission. The accumulation of data packets may cause congestion, leading to a decrease in data transmission across the network, which could ultimately impact the UASN's robustness [17]. This critical issue has not been adequately addressed in prior research. Therefore, when assessing the processing capacity of network links, it is essential to consider the potential impact of data packet

congestion resulting from the limited transmission rates, in addition to factors such as the underwater acoustic channel. We provide some reasonable assumptions below for facilitating analysis.

- 1) For the analysis of factors impacting data transmission in UASN, it is assumed that the initial deployment locations of each node are predetermined. Furthermore, the UASN is assumed to be a connected network without any node additions or removals during the network evolution process.
- 2) We partition dynamic and evolving networks into multiple static networks across different time slots. For each network in a given time slot, it is assumed that the external environment is relatively stable, nodes are stationary, and every link in the network experiences the same environmental effects.
- 3) As the time slot changes, it is assumed that there can be variations in external environmental factors and node positions. Consequently, the network analysis is performed within a coherent time slot, and the network is reanalyzed only when network changes exceed a predefined threshold.

B. Current Movement Model

In this work, in order to achieve a more realistic simulation of ocean current dynamics, we adopt a hybrid approach combining Lamb vortices [18] and directed water flow to model the currents in each layer. Additionally, we incorporate the velocity-depth relationship obtained in the Gulf Stream [19], where the velocity decreases with increasing depth in a multilayer current model [20]. The velocity of the ocean current at point $i(x_i, y_i, z_i)$ in layer h at time t can be expressed as:

$$\begin{cases} V_{x_i}^c(t) = v_x^h(t) - \Gamma(t) \frac{y_i - y_v^h}{2\pi d_{iv}^2} [1 - e^{-\frac{d_{iv}^2}{(\delta^h)^2}}] \\ V_{y_i}^c(t) = v_y^h(t) + \Gamma(t) \frac{x_i - x_v^h}{2\pi d_{iv}^2} [1 - e^{-\frac{d_{iv}^2}{(\delta^h)^2}}] \\ V_{z_i}^c(t) = v_z^h(t). \end{cases} \quad (1)$$

In (1), $V_{x_i}^c(t)$, $V_{y_i}^c(t)$, and $V_{z_i}^c(t)$ represent the velocity components along the x , y , and z axes, respectively, of the ocean current at time t , at location i in layer h . Meanwhile, $v_x^h(t)$, $v_y^h(t)$, and $v_z^h(t)$ represent the velocity components along the x , y , and z axes, of the water flow at time t in layer h . Consistent with the approach described in [21], [22], our focus is primarily on the horizontal motion of the current within the current layer, disregarding the vertical motion. The variables x_v^h and y_v^h denote the positions of the vortex center along the x and y axes. Additionally, d_{iv} represents the distance between point i and the vortex center, while $\Gamma(t)$ and δ^h correspond to the radius and intensity of the vortex.

By employing the velocity of the ocean current as given in (1), the ocean current force at point $i(x_i, y_i, z_i)$ in layer h at time t can be determined as follows [23]:

$$\vec{F}_{v_i}^c(t) = -0.5 A_r \rho_w c_o (\vec{V}_{v_i}^c(t))^2. \quad (2)$$

In (2), A_r represents the cross-sectional area of node i , ρ_w denotes the average density of seawater, c_o represents the viscosity coefficient of seawater, and $\vec{V}_{v_i}^c(t)$ denotes the ocean current velocity at point i at time t , expressed as $\vec{V}_{v_i}^c(t) = [V_{x_i}^c(t), V_{y_i}^c(t), V_{z_i}^c(t)]$.

In this work, it is assumed that changes in the marine environment occur exclusively during time slot transitions, involving variations in factors such as vortex position, radius, intensity, wind speed, and ship coefficients. During a given time slot, it is reasonable to assume that the node reaches a state of equilibrium, experiencing gravity force $\vec{F}^g$, buoyancy force $\vec{F}^b$, current force $\vec{F}^c$, and tension force $\vec{F}^r$. Consequently, by simultaneously considering the aforementioned equations, we can ascertain the forces exerted on the node.

$$\begin{cases} \vec{F}_{n_i}^r + \vec{F}_{n_i}^g + \vec{F}_{n_i}^b + \vec{F}_{n_i}^c = 0 \\ \alpha = \arctan\left(\frac{F_{n_i}^c}{F_{n_i}^b - F_{n_i}^g}\right). \end{cases} \quad (3)$$

Considering the nodes are securely anchored to the seabed, it is ensured that the buoyancy force F^b exceeds the gravity force F^g for every node. This guarantees that the tether remains under tension, maintaining a constant length. In Fig. 2, the node's current position can be determined by using its previous position, as well as the length and angular deviation of the tether [24].

$$\begin{cases} x_i^t = x_i^0 + z_i^0 \sin \alpha_i^t \cos \varphi_i^t \\ y_i^t = y_i^0 + z_i^0 \sin \alpha_i^t \sin \varphi_i^t \\ z_i^t = z_i^0 \cos \alpha_i. \end{cases} \quad (4)$$

In (4), x_i^t , y_i^t , and z_i^t represent the current displacement of the node i at time slot t , while x_i^0 , y_i^0 , and z_i^0 represent the initial deployment position of the node i . α_i^t and φ_i^t denote the angular deviations of the tether at the current time t . By employing the Jacobi iterative method and solving equations (1) to (4) with the previous time slot's node positions as initial values, the current node displacement can be calculated.

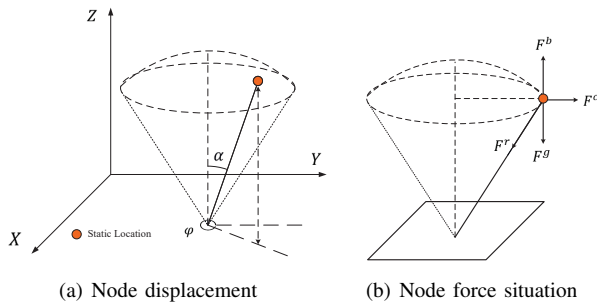


Fig. 2. The movement of the node

C. Percolation Model

The percolation model can be formulated as a mathematical problem as follows. Consider a three-dimensional grid model with $n \times n \times n$ vertices, where neighboring vertices are connected by edges with a probability $p \in [0, 1]$. The connectivity of each edge is statistically independent of others. As the

size of the grid model approaches infinity (i.e., n tends to infinity), we aim to determine the critical range of p , denoted as ρ_c , for which there exists at least one percolating flow that spans the entire grid. In the context of network theory, the percolating flow spanning the entire grid corresponds to the largest connected component in the current network.

For a time-varying network G , at each time slot, the network's current state can be characterized by its topology structure and link quality. By associating a link quality $q_{ij} \in [0, 1]$ with each link e_{ij} between node i and node j , we establish a percolation model in network G . The link quality q_{ij} represents the relative performance of network links compared to a predefined standard. In communication networks, for instance, q_{ij} can be defined as the instantaneous rate ratio of data packets flowing along the link [25].

By gradually reducing the link quality of selected links to zero through simulations, which is equivalent to their removal from the network, the links that significantly impact network robustness can be identified. It is important to note that assigning a link quality value of zero does not imply the physical removal of these links. Rather, it signifies that the processing capabilities of these links are inadequate to sufficiently meet the demands of traffic transmission. This process primarily impacts the size of the network's largest connected component, thereby influencing its overall performance. Within percolation theory, the percolation threshold ρ_c holds particular significance as a metric of interest. Close to the percolation threshold, the network's largest connected component undergoes a sudden fragmentation into multiple smaller node clusters [26]. This observation highlights the importance of link qualities surpassing the percolation threshold in maintaining the network's global connectivity. Consequently, researchers frequently employ the percolation threshold ρ_c as an indicator of network robustness in numerous relevant investigations. While this critical phenomenon indeed indicates a sudden shift in the topological structure, it is crucial to recognize that these changes only capture a single aspect of the broader network dynamics observed in real-world scenarios.

Given the fundamental objective of an UASN to transmit data collected by sensor nodes to the sink node, our attention shifts from the number of nodes capable of maintaining communication with the sink node to the quantity of data that the sink node can accumulate in the current time slot. Therefore, if the sink node can still accumulate a significant portion of the data packets, we consider the current network highly functional, even in cases where the overall connectivity is compromised due to link failures. Consequently, it becomes imperative to employ a more appropriate metric than the percolation threshold ρ_c to assess the network's robustness within the present context.

To characterize the network's ability to guarantee services, Hamedmoghadam et al. [16] proposed a key metric, denoted as α , for assessing the reliability of a demand service network. In a concise form, it can be expressed as:

$$\alpha = \int_0^1 UD_\rho d\rho = \int_0^1 \frac{\text{tr}(R_\rho)F^T}{1_n^T F 1_n} d\rho. \quad (5)$$

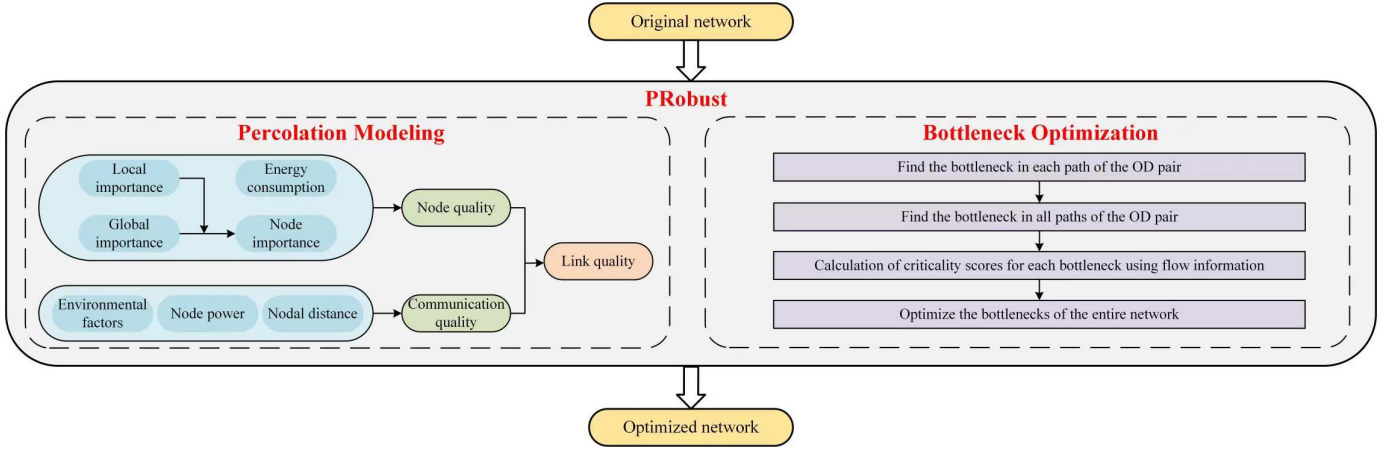


Fig. 3. Conceptual diagram depicting the proposed PRobust architecture

In (5), UD_ρ represents the proportion of total flow unaffected between all pairs of nodes under the threshold ρ in the percolation process, $tr(\cdot)$ denotes the trace of an $n \times n$ square matrix. It can be computed using the reachability matrix R and the flow demand matrix F of the network. The value of α is equivalent to the area under the UD_ρ curve in the ρ domain. α effectively captures the flow demand between all pairs of nodes during the percolation process and quantifies the unaffected flow in network variations, thereby reflecting the robustness of the current network. In contrast to conventional approaches that rely on percolation thresholds and similar metrics to evaluate network robustness, Hamedmoghadam et al. emphasize the importance of prioritizing the network's primary objective, which is to satisfy the flow demand. By simultaneously considering both the network's flow and its topological structure, the α metric provides a more comprehensive assessment of the current network's robustness.

IV. DESIGN OF THE PROBUST OPTIMIZATION MODEL

The overall workflow of the proposed PRobust model architecture is depicted in Fig. 3. PRobust comprises two principal phases: the percolation modeling phase and the bottleneck optimization phase. In the first phase, we present a design methodology for link quality, q_{ij} . Expanding on prior research [25], we take into account the distinctive characteristics of UASN and introduce a communication quality metric, q_{com} , specifically tailored for UASN environments. Furthermore, recognizing the impact of node properties on network performance in UASN, we propose a node quality metric, q_{node} , to capture node characteristics and utilize node importance to assess relative significance among nodes. This leads to the integration of communication quality q_{com} with node quality q_{node} , resulting in a novel approach for link quality design, q_{ij} . In the second phase, we leverage the percolation model established based on the current network and employ graph theory to identify bottlenecks. To overcome the limitation of handling identical link qualities encountered in previous works, we introduce the flow information recorded by nodes to optimize bottlenecks, even when link qualities q_{ij} are equal.

Finally, we calculate a significance score s_{ij} for each link in order to optimize bottlenecks throughout the entire network.

A. Percolation Modeling

1) *Node quality*: Multiple factors may influence link quality. In most of existing works, link quality is commonly obtained based on the transmission capacity of the links. However, in UASN, the state of the nodes can also impact the service capacity of the links since the sensor nodes lack reliability in underwater. For example, low node energy will elevate the probability of node disruptions, consequently leading to lower link quality q_{ij} associated with such nodes.

Given the limited energy replenishment capability in UASN, factors like energy depletion and failures can result in node disruptions, affecting the overall network structure and potentially leading to network disconnections. To quantitatively evaluate the impact of a node's disruption on the entire network, we integrate local and global topological considerations to assess the collective influence of nodes. The node importance of node n_i is calculated as the product of its local importance, $node_l^i$, and its global importance, $node_g^i$, which is expressed as:

$$node_{imp}^i = node_l^i \times node_g^i. \quad (6)$$

The local importance, $node_l^i$, of node n_i is primarily determined based on its degree, representing the local connectivity of the node. To differentiate the relative importance among nodes with the same degree, we incorporate the characteristics of the network topology and define the local importance as follows [27]:

$$node_l^i = d(i) + \frac{d_N(i)}{D} = d(i) + \frac{\sum_{j \in N(i)} d(j)}{d(i) \cdot D}. \quad (7)$$

Here, $d(i)$ represents the degree of the current node n_i , $d_N(i)$ represents the average degree of neighboring nodes of the node n_i , and D denotes the maximum degree among all nodes in the UASN. The expression $\frac{d_N(i)}{D}$ is equivalent to the normalized average degree of neighboring nodes of the current node. As depicted in Fig. 4, we consider a hypothetical

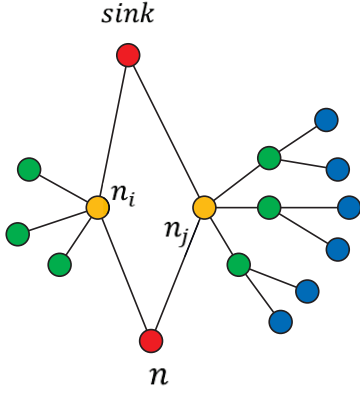


Fig. 4. Illustration of node local importance

scenario where the path from node n to the sink node includes intermediate nodes n_i and n_j , both having the same degrees, denoted as $d(i)$ and $d(j)$ respectively. Our objective is to assess the relative impact of n_i and n_j on the local network. Notably, it is observed that a node n_j with a higher average degree among its neighbors is more likely to be situated within a densely connected node cluster. In the event of node failures, disruptions to nodes n_j in dense node clusters may potentially impact a larger number of links compared to sparse node clusters. Consequently, we conclude that n_j exerts a more substantial influence on the local network than n_i . Even in the case of n_i 's failure, it will impact a fewer number of links compared to the potential impact resulting from the failure of n_j . When evaluating the node quality, denoted as q_{node}^i , we tend to attribute a higher level of reliability to n_i , as it belongs to a sparse node cluster.

We define the global importance as the advanced node betweenness centrality proposed by ourself, which quantifies the interconnectivity between associated nodes. The node betweenness centrality is commonly used to measure the proportion of shortest paths passing through a particular node in the current network, and a higher node betweenness centrality implies a more influential role in network communication [28]. Considering the specific characteristics of UASN network transmission, we focus solely on the shortest paths between the sink node and each individual node when calculating the advanced node betweenness centrality:

$$node_g^i = \sum \frac{n_{st}^i}{g_{st}}. \quad (8)$$

Here, g_{st} represents the number of shortest paths from the sink node to the target node, and n_{st}^i represents the number of shortest paths from the sink node to the target node that pass through node n_i . A higher value of the advanced node betweenness centrality for node n_i indicates that the node plays a more significant role in forwarding data packets to the sink node.

The residual energy of a node is a crucial factor influencing its reliability, as it reflects the node's current lifespan and its capacity to carry out tasks. We posit that a higher residual energy value indicates a more dependable node. Let E_{re}^i denote the current residual energy of sensor node n_i , and E_{ini}^i

represent its initial energy. We define the node energy factor F_{energy}^i of node n_i as follows:

$$F_{energy}^i = \frac{E_{re}^i}{E_{ini}^i}. \quad (9)$$

Ultimately, we undertake a comprehensive assessment of the potential influence of node on the overall network and define the node quality q_{node}^i for node n_i . Subsequently, a normalization procedure is applied to standardize the node quality values across all nodes:

$$q_{node}^i = F_{energy}^i \times (1 - node_{imp}^i). \quad (10)$$

2) *Communication quality*: When analyzing link quality, prior studies predominantly assumed successful packet transmission, considering only data packet congestion resulting from limitations in link transmission rate. However, in UASN, the communication capability of underwater acoustic channels introduces the possibility of transmission errors or failures during packet delivery. Consequently, relying solely on transmission rate to assess the processing capability of the current link is inadequate. In networking, packet delivery ratio (PDR) commonly represents the current link's ability to correctly transmit data packets. Nonetheless, evaluating the processing capability of the current link solely based on PDR disregards the impact of transmission rate limitations on link processing capability. Thus, when evaluating link processing capability, it is essential to comprehensively consider both aspects.

For the sake of convenience, we assume that the transmission rate varies when the transmission power p_{ij} differs across nodes. We adopt an underwater acoustic channel model that is based on the Urick model [29]. The attenuation of a sound signal with a frequency of f kHz transmitted over a distance of d m can be estimated as follows:

$$\frac{10 \log A(d, f)}{A_0} = 10k \log d + 10d \log \alpha(f). \quad (11)$$

In (11), A_0 represents a unit normalization constant, while k denotes an expansion factor utilized to characterize the geometric profile of sound wave propagation. Unless explicitly stated otherwise, a typical value of 1.5 is adopted for k . The absorption coefficient $\alpha(f)$ in dB/km can be estimated according to the subsequent formulation [30]:

$$10d \log \alpha(f) = \frac{0.11f^2}{1 + f^2} + \frac{44f^2}{4100 + f^2} + 2.75 \times 10^{-4} f^2 + 0.003. \quad (12)$$

Under the given conditions, when considering solely path loss, the signal-to-noise ratio (SNR) of a signal with frequency f transmitted over a distance d can be estimated as follows:

$$SNR(d, f) = \frac{P/A(d, f)}{N(f)\Delta f}. \quad (13)$$

In (13), P denotes the transmission power, $A(d, f)$ represents the given path loss, $N(f)$ represents the environmental noise, and Δf denotes the receiver noise bandwidth. The environmental noise $N(f)$ is computed by considering the

contributions from turbulence (N_t), shipping (N_s), waves (N_w), and thermal (N_{th}) noise sources.

$$\begin{aligned}
10 \log N_t(f) &= 17 - 30 \log f \\
10 \log N_s(f) &= 40 + 20(s - 0.5) + 26 \log f \\
&\quad - 60 \log(f + 0.03) \\
10 \log N_w(f) &= 50 + 7.52^{\frac{1}{2}} + 20 \log f \\
&\quad - 40 \log(f + 0.4) \\
10 \log N_{th}(f) &= -15 + 20 \log f.
\end{aligned} \tag{14}$$

To facilitate the estimation of the packet delivery rate between two underwater acoustic sensor nodes, binary phase-shift keying (BPSK) modulation is employed, with each symbol carrying one bit. In a non-fading additive white Gaussian noise (AWGN) channel, the bit error rate (BER) can be determined as follows [31]:

$$BER(d, f) = \frac{1}{2} \left(1 - \sqrt{\frac{SNR(d, f)}{1 + SNR(d, f)}} \right). \tag{15}$$

Considering the given expression for $SNR(d, f)$, the probability of successfully transmitting m bit data packets between two sensor nodes at a distance of d in an acoustic link with frequency f can be formulated as follows:

$$p(d, f, m) = (1 - BER(d, f))^m. \tag{16}$$

In conclusion, the communication quality q_{com}^{ij} is defined, and a normalization operation is applied to the communication quality values of all links:

$$q_{com}^{ij} = \frac{v_t}{v_{max}} \times p(d, f, m). \tag{17}$$

3) *Link quality*: Given the unique characteristics of UASN compared to other network scenarios, the processing capability of the current link is influenced by both nodes and edges. To account for both node attributes and edge attributes in the network, weight coefficients are introduced to balance node quality (q_{node}) and communication quality (q_{com}). We define λ_1 and λ_2 as the weight coefficients for node quality and communication quality, respectively, with the constraint that $\lambda_1 + \lambda_2 = 1$. The final link quality (q_{ij}) is defined as follows:

$$q_{ij} = \lambda_1 \times \frac{q_{node}^i + q_{node}^j}{2} + \lambda_2 \times q_{com}^{ij}. \tag{18}$$

In the aforementioned equation, q_{ij} denotes the link quality of the link e_{ij} connecting node n_i and node n_j . The variables q_{node}^i and q_{node}^j represent the node qualities of nodes n_i and n_j , respectively. Additionally, q_{com}^{ij} represents the communication quality of the link e_{ij} . Taking into account the previous normalization of node quality q_{node} and communication quality q_{com} , we ensure that the current link quality $q_{ij} \in [0, 1]$. Through dynamic adjustment of the weight coefficients according to different scenarios, we can identify a set of relatively optimal weight coefficients that achieve a balanced trade-off between node quality and communication quality.

B. Bottleneck Optimization

Once the percolation model is established, we can identify the bottlenecks that impede network traffic transmission within the current model. Considering the relative reflection of link quality q_{ij} on the processing capability of the current link in the percolation model, it can be assumed that the impact on links with small q_{ij} is inevitable when links with larger q_{ij} are affected. Moreover, as the robust optimization in UASN aims to minimize the impact on traffic reaching the sink node, more attention is given to the internal traffic of the network. In real-world network, the constraints imposed by link service capability on data transmission may not lead to link disruptions, but they significantly impact network performance. To facilitate the observation of changes in network performance within the percolation model, we assume that the aforementioned traffic-constraining links with q_{ij} become 0 and investigate the resulting performance changes. The steps of our proposed bottleneck optimization algorithm are outlined as follows:

Step 1: For each target node $n_t \in S_n$ and the sink node s , forming a node pair $S - T$, a directed path R_{st} is established from sensor node n_t to sink node s , consisting of links $e_{ij} \in E$ that connect different nodes $n_i, n_j \in V$ in the same direction, with n_t as the first node and s as the last node. Considering the scenario of multipath routing, a set of directed paths R_{st} is utilized to store all routing paths from sensor node n_t to sink node s . Subsequently, the set of directed paths R is employed to record paths from all nodes to the sink node, while the link quality matrix Q is utilized to store the link quality q_{ij} of each link e_{ij} in the network. Furthermore, we utilize the flow matrix F to record the flow information collected by nodes in multi-hop routing, where f_{ij} represents the flow carried by each link e_{ij} . The set of bottlenecks is denoted as E_b , initially initialized as empty ($\emptyset$).

Step 2: For each node pair $S - T$, we establish an auxiliary link set E_t initialized as empty ($\emptyset$). A directed path R_{st} from sensor node n_t to sink node s consists of multiple links e_{ij} . We create a temporary link set E_m , also initialized as empty ($\emptyset$). By traversing the entire routing path R_{st} and utilizing its associated link quality set Q_{st} , we identify the link e_{ij} with the minimum link quality q_{ij} . This link is deemed the bottleneck on the current path R_{st} and is included in the link set E_m due to its restrictive impact on traffic flow. When the link set E_m contains only one link ($|E_m| = 1$), it signifies the existence of a sole bottleneck on the current routing path, and we add the link e_{ij} to the auxiliary link set E_t . However, when the link set E_m contains more than one link ($|E_m| > 1$), implying the presence of multiple bottlenecks on the current routing path, we must select the relatively more significant one. In scenarios where multiple links possess the minimum q_{ij} value, we designate the link e_{ij} with the highest flow from the set F_{st} as the bottleneck on the current path, subsequently appending it to the auxiliary link set E_t . This is because when these links have limited capacity to handle the traffic, the disconnection of the link carrying the highest flow will have the most significant impact on the network's data transmission. Therefore, it should be selected as the bottleneck.

Step 3: For all routing paths between node pairs $S - T$,

Algorithm 1 Discovering The Bottlenecks On Each Path**Input:** S_n , s , the routing set R , the link quality matrix Q , the flow matrix F **Output:** the set of bottlenecks between $S - T$ E_t **Initialization:** $E_t = \emptyset$

```

1: for  $R_{st} \in R$  do
2:   Create  $E_m = \emptyset$ 
3:   for  $e_{ij} \in R_{st}$  do
4:     if  $q_{ij} = \min(Q_{st})$  then
5:        $E_m = E_m \cup e_{ij}$ 
6:     end if
7:   end for
8:   if  $|E_m| = 1$  then
9:      $E_t = E_t \cup e_{ij}$ 
10:  else
11:    for  $e_{ij} \in E_m$  do
12:      if  $f_{ij} = \max(F_{st})$  then
13:         $E_t = E_t \cup e_{ij}$ 
14:      end if
15:    end for
16:  end if
17: end for

```

we iterate through the auxiliary link set E_t and create an empty temporary link set E_n . The link e_{ij} with the highest link quality q_{ij} from E_t is selected as the bottleneck in the communication between node pairs $S - T$ and added to the link set E_n . When a link with q_{ij} less than the max value among E_n in Q_{st} (in Fig. 5 it equals 0.6) is disconnected, it is considered that the data between the target node 1 and the sink node 8 remains unaffected as communication can still be maintained between node 2 and node 5. However, when all links with q_{ij} less than 0.7 (or any value greater than the max value in Q_{st} and less than 1) are disconnected, the link between node 2 and node 5 is severed, resulting in the inability to establish communication between node 1 and node 8. In this scenario, the link between node 2 and node 5 is deemed to be the actual link that maintains communication between the target node 1 and the sink node 8, thus selected as the bottleneck in the current node pair 1 – 8. If the link set E_n contains only one link, it is added to the set of bottlenecks E_b . If the link set E_n contains more than one link, the link with the highest flow among the edges is similarly selected as the bottleneck on the current path and added to the set of bottlenecks E_b .

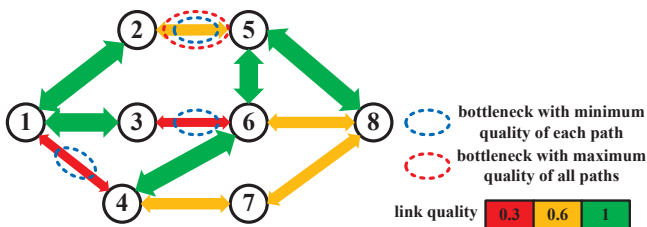


Fig. 5. Illustration of bottleneck optimization

Algorithm 2 Discovering The Bottlenecks Between $S - T$ **Input:** S_n , s , the routing set R , the link quality matrix Q , the flow matrix F **Output:** the set of bottlenecks E_b **Initialization:** $E_b = \emptyset$

```

1: for  $n_t \in S_n$  do
2:   Create  $E_t = \emptyset$ 
3:   Discovering the bottlenecks on each path
4:   Create  $E_n = \emptyset$ 
5:   for  $e_{ij} \in E_t$  do
6:     if  $q_{ij} = \max(Q_{st})$  then
7:        $E_n = E_n \cup e_{ij}$ 
8:     end if
9:   end for
10:  if  $|E_n| = 1$  then
11:     $E_b = E_b \cup e_{ij}$ 
12:  else
13:    for  $e_{ij} \in E_n$  do
14:      if  $f_{ij} = \max(F_{st})$  then
15:         $E_b = E_b \cup e_{ij}$ 
16:      end if
17:    end for
18:  end if
19: end for

```

Step 4: In order to assess the influence of each bottleneck in the set E_b on the global network traffic, a significance score s_{ij} is defined to indicate the proportion of its traffic demand relative to the total traffic demand:

$$s_{ij} = \frac{f_{st}}{1_n^T F 1_n} = \sum_{s,t \in V, e_{st}^* = e_{ij}} \frac{f_{st}}{1_n^T F 1_n}. \quad (19)$$

In (19), f_{st} represents the flow between the current node pair $S - T$, and F represents the flow distribution matrix of the current network. When a link e_{ij} may serve as a bottleneck for multiple node pairs comprising nodes n_t and the sink node in $S - T$ (in (19) it equals to $e_{st}^* = e_{ij}$), we accumulate the ratio of flow between each node pair in $S - T$ to the overall flow. By relying solely on the network's flow matrix for computation, our framework remains suitable for networks with non-uniform flow distribution. Finally, we rank all links in E_b according to their significance score s_{ij} . The higher the score, the more pronounced the constraining impact on data transmission in the network, and the greater the influence on network robustness.

Upon obtaining the set of bottleneck links ranked by their significance scores s_{ij} , our strategy is to selectively optimize the link quality q_{ij} in order to enhance network robustness. It is evident that optimizing a larger number of links leads to a greater enhancement in network robustness. However, it is important to acknowledge that this improvement is accompanied by an increase in additional overhead. Thus, it is crucial to strike a balance between enhancing network robustness and minimizing costs. In practical scenarios, we prioritize the optimization of links with higher significance scores. By specifically targeting these bottlenecks, our objective is to

Algorithm 3 Optimizing The Bottlenecks

Input: S_n , s , the routing set R , the link quality matrix Q , the flow matrix F , the number of the optimized bottlenecks n

Output: the sorted set of bottlenecks E_b

Initialization: $s_{ij} = 0$

```

1: for  $e_{ij} \in E_b$  do
2:   Calculate  $s_{ij}$ 
3: end for
4: Sort  $\forall e_{ij} \in E_b$  in descending order according to  $s_{ij}$ 
5: Set the link quality  $q_{ij}$  of top  $n$  bottlenecks according to  $s_{ij}$  to 1

```

elevate their link quality to the optimal state ($q_{ij} = 1$). This targeted approach enables efficient resource allocation and prioritization of improvements, leading to an overall enhancement in network robustness and performance.

In UASN, there are several approaches that can be employed to achieve the improvement in link quality. One such approach involves leveraging the capabilities of Autonomous Underwater Vehicles (AUVs) for inter-node data transmission. By utilizing the mobility and communication capabilities of AUVs, we can establish reliable and efficient data transfer between nodes, thereby enhancing the link quality and improving overall network performance. Another avenue for improvement is through the modification of inter-node communication methods. By optimizing the communication protocols and techniques employed among nodes, we can ensure the optimal processing capability of the existing links, resulting in improved link quality and enhanced network robustness.

Through the implementation of the optimization techniques mentioned before, we can assess the final effects on the network's performance and robustness. By comparing key metrics such as packet delivery ratio, delay, throughput, and network robustness metrics α before and after the optimization, we can quantitatively evaluate the effectiveness of the PRobust. These evaluations provide insights into the extent to which the link quality improvements contribute to overall network performance enhancement, validating the efficacy of our optimization strategies in enhancing network robustness.

V. SIMULATION RESULTS AND ANALYSIS

In this section, we provide empirical evidence of the efficacy of the PRobust network robustness optimization strategy, which is grounded in percolation theory. We conduct a comparative performance evaluation between PRobust and three state-of-the-art network robustness optimization approaches, namely Criticality Score [16], Edge Betweenness Centrality [28], and Percolation Criticality [15].

A. Simulation Settings

To demonstrate its applicability to various networks, we begin by generating a random network and comparing the performance variations resulting from different robustness optimization strategies. The performance evaluation is conducted through 2000 simulation runs. In practical sensor networks,

nodes with higher power generally exhibit superior communication capabilities. To capture the diversity in communication abilities among nodes, we assume a proportional relationship between node power and transmission rates [32]. Nodes with higher power exhibit faster transmission rates and are less prone to packet congestion. Furthermore, to simulate a dynamic marine environment, we assume the stability of the marine environment within each time slot, with changes occurring in subsequent slots. The remaining parameters required for the simulation evaluation are provided in Table I [33].

TABLE I
SYSTEM PARAMETERS

Parameter	Value
Sensor field	1000 m $\times$ 1000 m $\times$ 1000 m
Node transmission power	[6,12,18] W
Node receiving power	1.1 W
Node transmission rate	[4600, 5750, 6900] bits/s
Node initial energy	20000 J
Carrier frequency	48 KHz
Bandwidth	5 KHz
Communication range	400 m
Packet length	1 KB
Time slot	5 s
Wind speed	[0,10] m/s
Ship coefficient	[0,1]
Maximum number of retransmissions	5

TABLE II
CURRENT MOVEMENT MODEL PARAMETERS

Layer	Depth (m)	v_x^h, v_y^h, v_z^h (m/s)
1	0 - 200	0.25, 0.25, 0
2	200 - 400	0.20, 0.20, 0
3	400 - 600	0.15, 0.15, 0
4	600 - 800	0.10, 0.10, 0
5	800 -1000	0.05, 0.05, 0

TABLE III
CALCULATION PARAMETERS OF CURRENT MOVEMENT

Parameter	Value
Sea water density	1.05 g/cm ³
USN density	0.8 g/cm ³
USN volume	0.025 m ³
USN cross-sectional area	0.16 m ²
Viscosity coefficient	0.5
Gravitational acceleration	9.8 m/s ²
Vortex radius	[50,100] m
Vortex intensity	[0,50]

The ocean current movement model utilizes a five-layer ocean current model, and the primary parameters for each layer are outlined in Table II. In recognition of the dynamic characteristics of the marine environment, we consider the possibility of varying positions for vortex centers. The crucial parameters for computing ocean current motion are presented in Table III.

B. Baselines

The PRobust approach proposed in this work is compared with three other baseline methods as follows:

- 1) Criticality Score (CS): In this methodology, the percolation model is constructed by evaluating the link quality as the ratio of the link data transmission rate to the maximum rate. The bottleneck is determined as the maximum link among all identified bottlenecks.
- 2) Edge Betweenness Centrality (EB): In this approach, the edge betweenness centrality is calculated as the proportion of times each link occurs on the shortest paths between other node pairs. The link that has a higher occurrence on the shortest paths of node pairs involved in communication is considered as the bottleneck.
- 3) Percolation Criticality (PC): In this scheme, the percolation model is constructed using the link quality defined as the ratio between the data transmission rate and the maximum rate. The bottleneck is determined by identifying the link that induces the collapse of the largest connected component in the network when the threshold is modified.

C. Evaluation Metrics

To conduct a comprehensive evaluation of the performance differences between the network before and after optimization, we conduct comparisons across various dimensions, including network robustness metric α , packet delivery ratio, delay, and throughput. For each of these indicators, we provide the calculation methods to ensure transparency and reproducibility.

- 1) Network robustness metric α : In the percolation theory, with the increase of the threshold ρ , the links in the network with a link quality below ρ are gradually removed, leading to a gradual decrease in the proportion of nodes capable of maintaining communication with the sink node until it reaches 0. During the link removal process, the integral value of the curve representing the unaffected nodes' communication is denoted as α and can be expressed as (5) mentioned earlier. Compared to directly employing the percolation threshold ρ as an indicator of network robustness, α offers a more accurate representation of the network's capacity to sustain its functionality. Given that all data gathered by UASN nodes must be conveyed to the sink node, our assessment of network traffic focuses solely on the potential impact on the flow from each node to the sink node.
- 2) Package delivery ratio (PDR): PDR is defined as the ratio of the number of successfully delivered packets from all nodes to the sink node to the total number of packets sent during a data transmission session. In our simulation, we consider both congestion and theoretical packet delivery rates to calculate the communication quality in terms of PDR, which is expressed as follows:

$$P = \frac{\sum_{i=1}^n \prod_{j=1}^h 1 - (1 - q_{com}^{j,j+1})^r}{n}. \quad (20)$$

In (20), P denotes the average PDR from all nodes to the sink node, n represents the count of nodes in the present network, h signifies the number of hops along the current path, $q_{com}^{j,j+1}$ denotes the communication quality

from node j to node $j+1$, and r represents the maximum number of retransmissions within the current network.

- 3) Delay: The average end-to-end delay is defined as the mean time required for all nodes in the network to transmit a data packet to the sink node and for the packet to be successfully received. In our simulation, we focus on the packet transmission delay and propagation delay as the primary factors contributing to the end-to-end delay, and they can be mathematically expressed as follows:

$$D = \frac{1}{n} \times \sum_{i=1}^n \sum_{j=2}^h \left\{ \frac{L}{b_m} + \sum_{k=1}^r \left(\frac{L}{b_m} + \frac{R}{v} \right) \times (1 - (1 - q_{com}^{j,j+1})^r) \right\}. \quad (21)$$

In the above equation, D denotes the average delay in the current network, n represents the number of nodes in the network, h represents the number of hops along the current path, L represents the length of the data packet, b_m represents the transmission rate of the current node, R represents the distance between the current node j and the next hop node $j+1$, v represents the speed of sound propagation underwater, $q_{com}^{j,j+1}$ represents the communication quality from node j to node $j+1$, and r represents the maximum number of retransmissions in the current network.

- 4) Throughput: The cumulative number of packets successfully received by the sink node within a single time slot. In our experimental analysis, we compute the average number of packets received by the sink node per time slot throughout the entire network lifespan. This metric can be formulated as follows:

$$T = \frac{\sum_{i=1}^t \sum_{j=1}^n L \times P}{t \times n}. \quad (22)$$

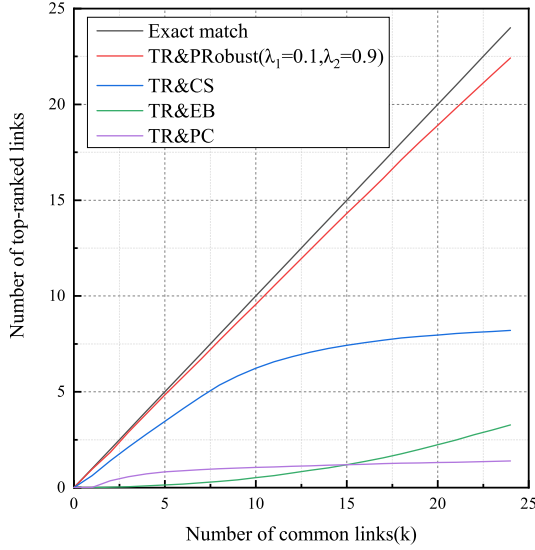
In (22), T represents the throughput of the current network, t represents the network's lifetime, n represents the number of nodes in the current network, L represents the length of the data packet, and P represents the packet delivery ratio from the current node n to the sink node.

D. Simulation Results

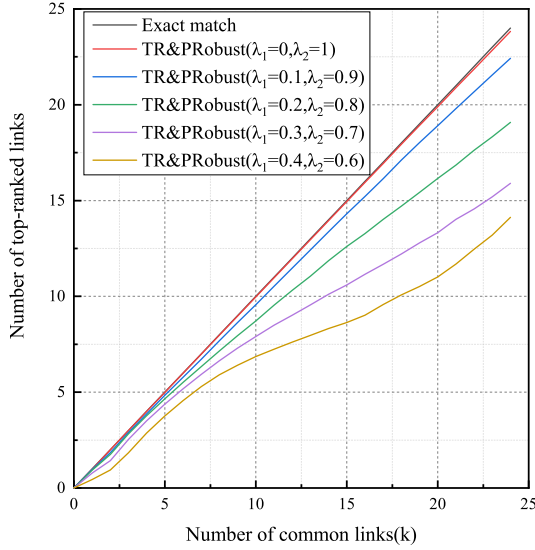
In this section, we commence by conducting a comparative analysis between the bottlenecks identified by our proposed PROBUST method and the ground truth bottlenecks, thereby validating the effectiveness of our approach. Subsequently, we perform an in-depth analysis of the network optimized using PROBUST. Finally, we conduct a comprehensive comparison of PROBUST with other baselines.

- 1) *Reliability Validation of PROBUST*: To assess the validity of the bottlenecks identified by PROBUST in the network, we employ a numerical simulation approach for verification. We introduce a small perturbation ϵ (in our simulations, $\epsilon = 0.00001$ is chosen) to the link quality q_{ij} of each link. Subsequently, we rank the perturbation magnitude of the robustness metric α for the resultant network. The link that demonstrates the highest increase in the robustness metric α

is considered the most critical bottleneck in the network. By employing this numerical simulation approach, we obtain the true ranking (TR) of each link in the network.



(a) PRobust and other methods

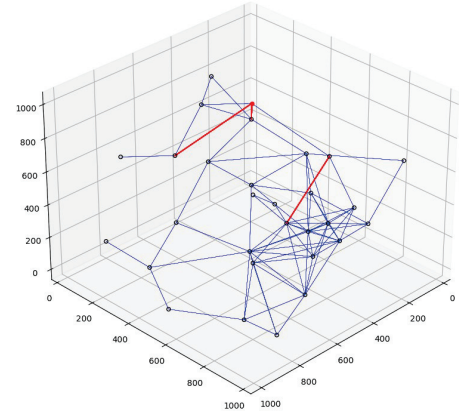


(b) PRobust with different weights

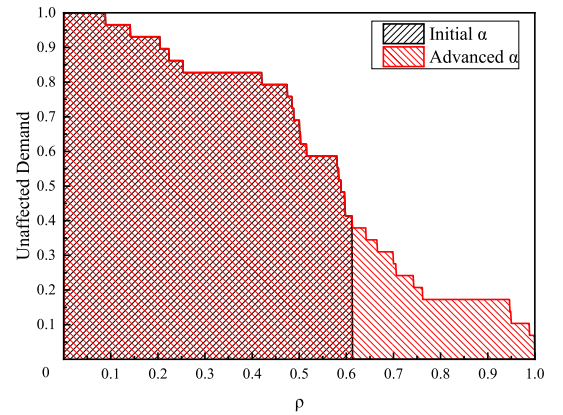
Fig. 6. Assessing the accuracy of PRobust and other methods in bottleneck optimization.

In our simulation of bottleneck optimization, multiple approaches were applied in a network with 100 randomly generated nodes, yielding respective sets of bottlenecks. To evaluate the accuracy of these findings across different approaches, we examined the top k bottlenecks obtained by each method and assessed the number of links that coincided with the true set of bottlenecks. Given the limited scale of our network, the number of bottlenecks by various methods was constrained, and we restricted k to a maximum value of 25. Fig. 6(a) demonstrates the correspondence between the bottlenecks discovered by PRobust ($\lambda_1=0.1$, $\lambda_2=0.9$), CS, EB, and PC, and the actual bottlenecks in the network, for values of k ranging from 1 to 25. Notably, PRobust ($\lambda_1=0.1$, $\lambda_2=0.9$) exhibited a

close alignment with the true bottlenecks in the network. CS demonstrated some capacity to identify bottlenecks when the link count was small, but its accuracy was limited due to the inadequate considerations in its percolation model construction. Conversely, significant disparities were observed between the bottlenecks identified by EB and PC and the true bottlenecks. In situations with a low number of identified bottlenecks, PC demonstrated faster discovery. However, as the link count increased, EB outperformed PC in bottleneck optimization. In Fig. 6(b), a comparison was made between the bottlenecks identified by PRobust using different weight coefficients and the true bottlenecks. When solely considering communication quality and disregarding node quality, PRobust ($\lambda_1=0$, $\lambda_2=1$) exhibited a close correspondence with the true bottlenecks in the network. Nevertheless, as the value of λ_1 increased, the optimized bottlenecks by PRobust gradually deviated from the true bottlenecks due to the growing significance of node quality. This deviation could impact the accuracy of bottleneck discovery. Thus, appropriate weight coefficients should be chosen in accordance with the network size.



(a) Location of bottlenecks



(b) Comparison of network robustness index α

Fig. 7. An example of network optimization leveraging PRobust.

2) *Optimization Results of PRobust:* In Fig. 7, we present an analysis of the network's changes before and after optimization using a network example comprising 40 nodes. Fig. 7(a) depicts the network, where the sink node is highlighted in red, and the sensor nodes within the UASN are represented by black hollow nodes. The links between nodes are shown as

blue lines, while the bottlenecks optimized by our proposed approach are indicated by red lines. During the optimization process, we focused on optimizing only three bottlenecks in the current network. From the observation of Fig. 7(a), we note that the bottlenecks tend to be located near the vicinity of the sink node or between two larger node clusters. This behavior can be attributed to the nature of our robustness metric α , which takes into account the traffic flow between all sensor nodes and the sink node. Consequently, the links surrounding the sink node experience higher traffic loads. When a link exhibits poor quality and bears a substantial amount of traffic, it can become a limiting link for the overall flow within the network. Additionally, in order to facilitate the transmission of data packets to the sink node, the links between larger node clusters also endure significant traffic loads, which further contribute to link-constrained flow patterns, consistent with our expectations.

In Fig. 7(b), we present the variation in the proportion of unaffected traffic in the network as the percolation threshold undergoes changes. The black line depicts this variation, and the size of the black region corresponds to the integral value of the curve between the range $[0, 1]$, representing the network's initial robustness metric α . As the percolation threshold increases, the proportion of unaffected traffic gradually diminishes until it reaches zero. However, through link optimization, we can mitigate this rate of decrease. Notably, when the percolation threshold reaches approximately 0.61, all traffic is impacted, rendering transmission to the sink node infeasible. This observation underscores the challenge faced by the initial network in ensuring robust network service in complex and dynamic environments.

Conversely, the red line illustrates the variation in the proportion of unaffected traffic in the network after optimization using PRobust. The area under the red curve corresponds to the robustness metric α of the optimized network. In our simulation, we represent the optimization of bottlenecks by assigning a link quality q_{ij} value of 1. In practical scenarios, real-time link optimization can be achieved by employing Autonomous Underwater Vehicles (AUVs) for node-to-node transmission or by modifying communication techniques. It is evident that by optimizing the three links depicted in Figure 7(a), we guarantee the preservation of unaffected traffic even in the most adverse environments (i.e., when the percolation threshold reaches 1), thereby significantly enhancing the overall network robustness.

3) *Comparison with Baselines:* Fig. 8 demonstrates the variations in the network's robustness metric α with different numbers of optimized links. It is evident that increasing the number of optimized links enhances the overall network robustness across all schemes. PRobust exhibits the most substantial improvement in network robustness, followed by CS, PC, and EB, which exhibit comparatively lower performance. In PRobust, as the value of λ_1 increases, the enhancement in network robustness α gradually diminishes. This can be attributed to the adverse impact of excessive consideration of node quality on the accuracy of bottleneck discovery. Furthermore, we observe that while the network's robustness continues to improve with an increasing number of optimized

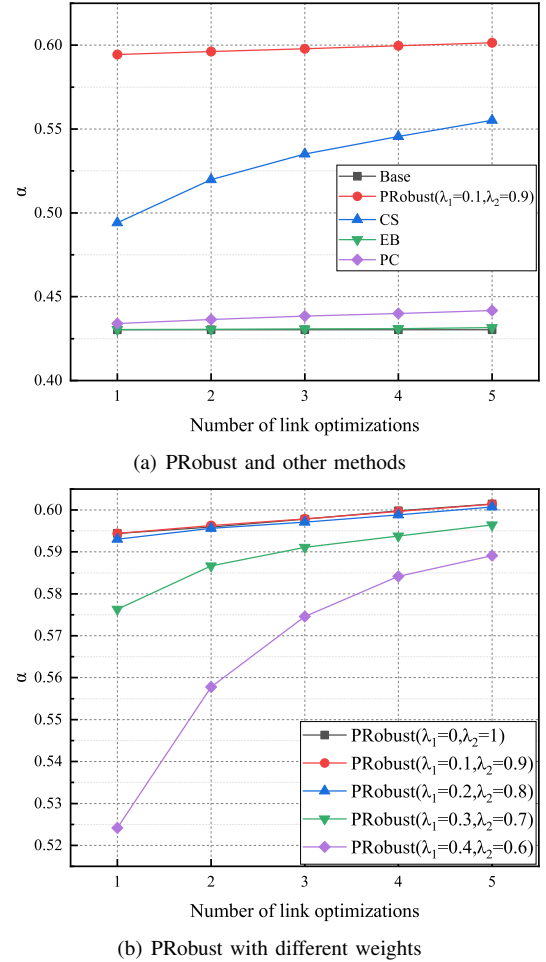


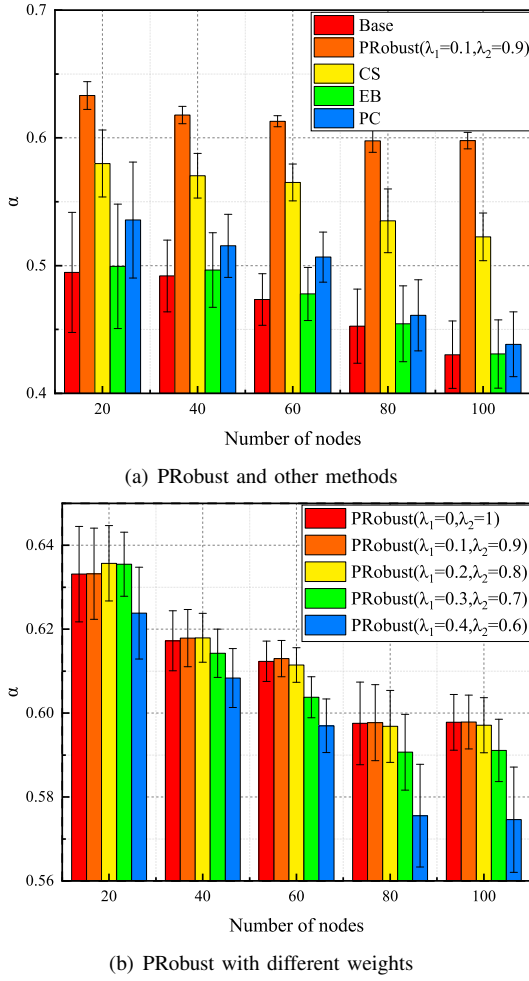
Fig. 8. The impact of optimized link numbers on α .

links, the magnitude of improvement becomes less significant beyond three optimized links for most schemes. Taking into account the associated overhead of link optimization, we uniformly optimize three links in the subsequent simulation processes.

Considering that the networks utilized in our simulations are generated randomly, we have considered the errors arising from these stochastic networks in the subsequent comparisons. The error bars depicted in the comparisons encapsulate 95% of the random networks.

Fig. 9 presents the variation in the network's α for different schemes as the network scale changes. In Fig. 9(a), PRobust demonstrates the most substantial enhancement in network robustness, followed by CS, while PC and EB exhibit comparatively lower improvements. Furthermore, PRobust exhibits lower variance compared to other schemes, indicating its superior robustness across diverse network instances. This superiority stems from PRobust's comprehensive consideration of both the topological and physical characteristics of the network when designing link quality, incorporating the influence of nodes and edges. In contrast, CS, which solely accounts for transmission rates, offers a less precise depiction of the current network conditions.

Additionally, we observe that as the network scale expands,

Fig. 9. Comparison of PRobust and other methods on α .

all schemes experience a decrease in the magnitude of robustness improvement. Nonetheless, PRobust displays a significantly higher enhancement ratio compared to other schemes. This can be attributed to PRobust's ability to effectively leverage the redesigned link quality, considering a larger number of edges within the network, thereby capturing the distinctions among various link qualities more accurately. Moreover, by taking into account the convergence characteristics of UASN, our proposed bottleneck optimization method accounts for traffic distribution within the network and addresses scenarios involving identical link qualities. The incorporation of the flow information in our approach facilitates improved ranking of these links, resulting in better alignment between the bottlenecks and the actual ones present in the network.

Fig. 9(b) illustrates the dependence of weight coefficients on network size for achieving maximum optimization of network robustness. PRobust ($\lambda_1=0.2, \lambda_2=0.8$) demonstrates the most notable enhancement in network robustness when the network comprises 20 nodes. However, as the network size increases, resulting in increased complexity, the weight coefficient λ_1 , which leads to the most significant improvement in network robustness, decreases. For a network containing 100 nodes, PRobust ($\lambda_1=0.1, \lambda_2=0.9$) exhibits the most substantial improvement in network robustness.

This behavior can be attributed to the observation that in smaller networks, a majority of the traffic tends to concentrate on a few nodes, rendering the quality of those nodes critical for traffic flow. Consequently, greater emphasis is placed on node quality. However, as the number of nodes expands, traffic is distributed across different nodes, diminishing the likelihood of a few nodes significantly impacting the overall network, as observed in smaller networks. Hence, it becomes justifiable to decrease the weight coefficient associated with node quality, λ_1 , while increasing the weight coefficient linked to communication quality, λ_2 , to more accurately reflect the processing capability of links.

As the network size continues to vary, PRobust ($\lambda_1=0.1, \lambda_2=0.9$) consistently manifests substantial improvements in network robustness. Thus, it can be concluded that, in the majority of network scenarios, selecting weight coefficients of $\lambda_1=0.1$ and $\lambda_2=0.9$ achieves a balanced consideration of node quality and communication quality.

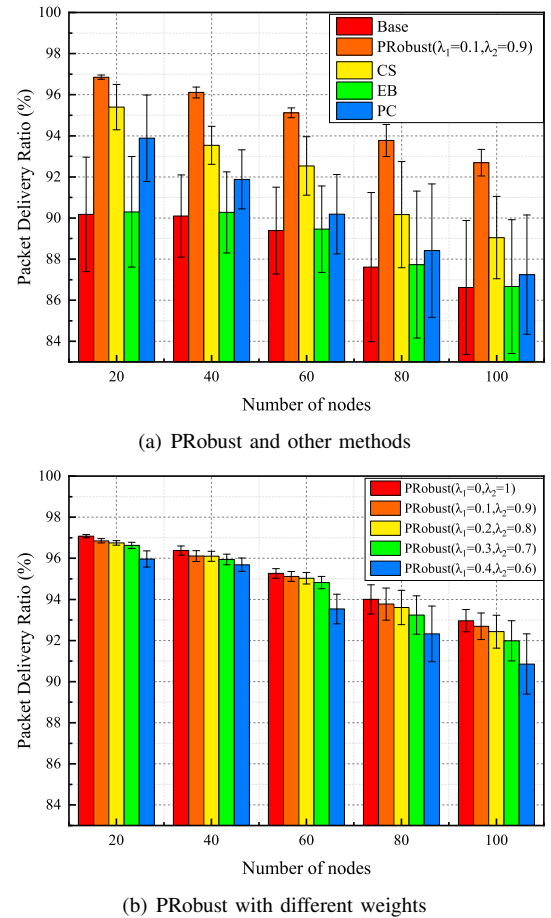


Fig. 10. Comparison of PRobust and other methods on PDR.

Fig. 10 illustrates the impact of different schemes on the packet delivery ratio in networks with varying sizes. As depicted in Fig. 10(a), PRobust demonstrates the most substantial improvement in packet delivery ratio, followed by CS, while PC and EB exhibit relatively poorer performance. Furthermore, PRobust exhibits the lowest variance among all schemes regarding the enhancement of the packet delivery ratio. This can be attributed to the fact that other schemes

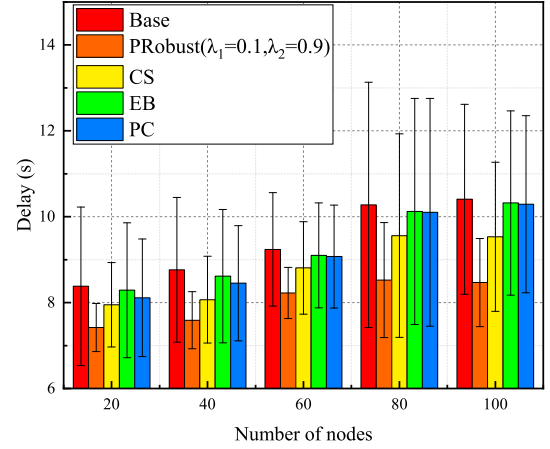
predominantly consider transmission rate and network topology as the limiting factors for packet transmission, lacking consideration of the realistic transmission characteristics in network scenarios.

Moreover, as the network size increases, all schemes experience a decline in the improvement of the packet delivery ratio. However, PRobust maintains a relatively stable enhancement proportion. Several factors contribute to this observation. The EB scheme solely considers the frequency of link usage without considering whether these frequently used links encounter transmission rate constraints that hinder traffic flow. The PC scheme optimized bottlenecks based solely on link quality constraints, neglecting the frequency of link usage. While the CS scheme partially considers both link quality and usage frequency, its design of link quality only incorporates transmission rate restrictions, failing to reflect the impact of the environment on link quality. Furthermore, its bottleneck optimization method cannot handle special cases where link qualities are equal, and optimizing less critical bottlenecks leads to a lower improvement in packet delivery ratio compared to PRobust.

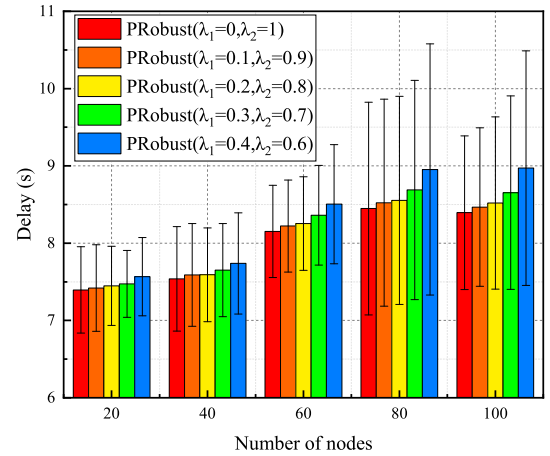
In Fig. 10(b), we compare the effects of PRobust schemes with different weight coefficients on the packet delivery ratio. As λ_1 increases, the improvement in packet delivery ratio diminishes. However, PRobust still maintains a high level of improvement compared to schemes such as CS. This is because excessively large values of λ_1 excessively emphasize the influence of nodes on link quality evaluation, thereby impeding the comprehensive reflection of communication quality's impact on link capacity.

Fig. 11 presents the impact of different schemes on network latency across varying network sizes. As depicted in Fig. 11(a), PRobust yields the most substantial reduction in network latency, followed by CS, while PC and EB exhibit comparatively poorer performance in latency reduction. Furthermore, PRobust exhibits the lowest variance among all schemes in terms of reducing network latency. Fig. 11(b) illustrates the influence of PRobust with different weight coefficients on network latency. Similarly, due to its superior consideration of link communication quality, PRobust demonstrates the most pronounced reduction in latency under the configuration of PRobust ($\lambda_1=0, \lambda_2=1$).

Fig. 12 portrays the influence of different schemes on network throughput in networks of varying sizes. Fig. 12(a) demonstrates that PRobust exhibits the most prominent enhancement in network throughput compared to all other schemes, with its effectiveness increasing notably as the network size expands. This phenomenon can be ascribed to PRobust's ability to identify bottlenecks that consider both link quality and network traffic, thereby optimizing traffic flow for a greater number of nodes. Fig. 12(b) presents the impact of different weight coefficients on network throughput improvement within the PRobust framework. Evidently, when $\lambda_1=0$ and $\lambda_2=1$, the overall network throughput experiences the most substantial enhancement, aligning precisely with our anticipated outcomes.



(a) PRobust and other methods



(b) PRobust with different weights

Fig. 11. Comparison of PRobust and other methods on delay.

VI. CONCLUSIONS

This paper investigates the problem of optimizing the robustness of UASN in ocean current environments. We propose PRobust, a robustness optimization model for UASN based on percolation theory, which incorporates both node and edge characteristics into link quality. To enhance network robustness, we introduce a dedicated bottleneck optimization method tailored for the percolation model network. The problem of bottleneck optimization is effectively addressed by introducing the flow information recorded by nodes. Extensive simulation results validate the reliability of our proposed PRobust approach, which yields significant improvements in network robustness while enhancing other key metrics. In essence, our PRobust method achieves enhanced network service reliability and performance with minimal overhead.

Our future work will primarily focus on the design of a robust prediction and control algorithm. This algorithm aims to maintain network stability and ensure a certain level of service capability in uncertain environments by leveraging historical environmental information. Moreover, we plan to integrate AUV scheduling and path planning to enable end-to-end optimization spanning from bottleneck discovery to AUV-optimized links.

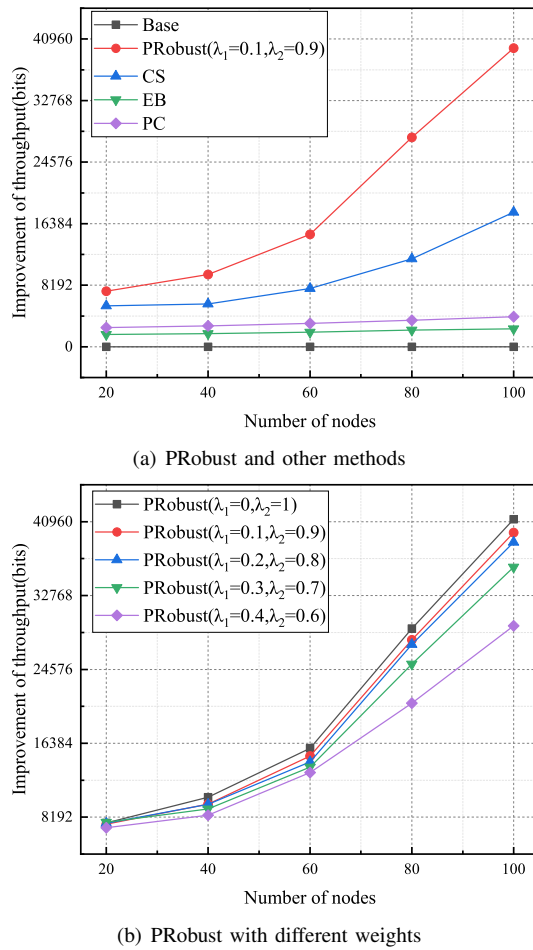


Fig. 12. Comparison of PRObust and other methods on throughput.

REFERENCES

- [1] Tie Qiu, Zhao Zhao, Tong Zhang, Chen Chen, and CL Philip Chen. Underwater internet of things in smart ocean: System architecture and open issues. *IEEE transactions on industrial informatics*, 16(7):4297–4307, 2019.
- [2] Shengming Jiang. On reliable data transfer in underwater acoustic networks: A survey from networking perspective. *IEEE Communications Surveys & Tutorials*, 20(2):1036–1055, 2018.
- [3] Junhai Luo, Yanping Chen, Man Wu, and Yang Yang. A survey of routing protocols for underwater wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 23(1):137–160, 2021.
- [4] XiaoLong Ren, Niels Gleinig, Dirk Helbing, and Nino Antulov-Fantulin. Generalized network dismantling. *Proceedings of the national academy of sciences*, 116(14):6554–6559, 2019.
- [5] Yilin Shen, Nam P Nguyen, Ying Xuan, and My T Thai. On the discovery of critical links and nodes for assessing network vulnerability. *IEEE/ACM Transactions on Networking*, 21(3):963–973, 2012.
- [6] Flaviano Morone and Hernán A Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65–68, 2015.
- [7] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulnerability of complex networks. *Physical review E*, 65(5):056109, 2002.
- [8] Alexander A Ganin, Maksim Kitsak, Dayton Marchese, Jeffrey M Keisler, Thomas Seager, and Igor Linkov. Resilience and efficiency in transportation networks. *Science advances*, 3(12):e1701079, 2017.
- [9] Milica Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):34–43, 2007.
- [10] Yougan Chen, Jianying Zhu, Lei Wan, Xing Fang, Feng Tong, and Xiaomei Xu. Routing failure prediction and repairing for auv-assisted underwater acoustic sensor networks in uncertain ocean environments. *Applied Acoustics*, 186:108479, 2022.
- [11] Haiyan Zhao, Jing Yan, Tao Wu, Aihong Li, and Xiaoyuan Luo. Energy-efficient underwater data collection: A q-learning based approach. *Journal of Marine Science and Application*, 21(3):204–218, 2022.
- [12] Xiaoyun Guang, Wenyu Qu, Chunfeng Liu, Tie Qiu, and Mohammed Atiqzaman. A subregional monitoring-oriented topology control strategy in uwsns. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 731–738. IEEE, 2019.
- [13] Yongwen Du, Jinzong Xia, Junhui Gong, and Xiaohui Hu. An energy-efficient and fault-tolerant topology control game algorithm for wireless sensor network. *Electronics*, 8(9):1009, 2019.
- [14] Changjun Fan, Li Zeng, Yizhou Sun, and Yang-Yu Liu. Finding key players in complex networks through deep reinforcement learning. *Nature machine intelligence*, 2(6):317–324, 2020.
- [15] Daqing Li, Bowen Fu, Yunpeng Wang, Guangquan Lu, Yehiel Berezin, H Eugene Stanley, and Shlomo Havlin. Percolation transition in dynamical traffic network with evolving critical bottlenecks. *Proceedings of the National Academy of Sciences*, 112(3):669–672, 2015.
- [16] Homayoun Hamedmoghadam, Mahdi Jalili, Hai L Vu, and Lewi Stone. Percolation of heterogeneous flows uncovers the bottlenecks of infrastructure networks. *Nature communications*, 12(1):1254, 2021.
- [17] Alessio Sacco, Matteo Flocco, Flavio Esposito, and Guido Marchetto. Owl: Congestion control with partially invisible networks via reinforcement learning. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.
- [18] Bartolome Garau, Alberto Alvarez, and Gabriel Oliver. Auv navigation through turbulent ocean environments supported by onboard h-adcp. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3556–3561. IEEE, 2006.
- [19] Daniel Halkin and T Rossby. The structure and transport of the gulf stream at 73 w. *Journal of Physical Oceanography*, 15(11):1439–1452, 1985.
- [20] Chunfeng Liu, Zhao Zhao, Wenyu Qu, Tie Qiu, and Arun Kumar Sangaiah. A distributed node deployment algorithm for underwater wireless sensor networks based on virtual forces. *Journal of Systems Architecture*, 97:9–19, 2019.
- [21] Guangjie Han, Songjie Shen, Houbing Song, Tao Yang, and Wenbo Zhang. A stratification-based data collection scheme in underwater acoustic sensor networks. *IEEE Transactions on Vehicular Technology*, 67(11):10671–10682, 2018.
- [22] Dario Pompili, Tommaso Melodia, and Ian F Akyildiz. Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks. *Ad Hoc Networks*, 7(4):778–790, 2009.
- [23] Jinmo Park and Nakwan Kim. Dynamics modeling of a semi-submersible autonomous underwater vehicle with a towfish towed by a cable. *International Journal of Naval Architecture and Ocean Engineering*, 7(2):409–425, 2015.
- [24] Linfeng Liu, Ruchuan Wang, and Fu Xiao. Topology control algorithm for underwater wireless sensor networks using gps-free mobile sensor nodes. *Journal of Network and Computer Applications*, 35(6):1953–1963, 2012.
- [25] Pablo Echenique, Jesús Gómez-Gardenes, and Yamir Moreno. Dynamics of jamming transitions in complex networks. *Europhysics letters*, 71(2):325, 2005.
- [26] Ming Li, Run-Ran Liu, Linyuan Lü, Mao-Bin Hu, Shuqi Xu, and Yi-Cheng Zhang. Percolation on complex networks: Theory and application. *Physics Reports*, 907:1–68, 2021.
- [27] Jiaxin Du, Guangjie Han, Chuan Lin, and Miguel Martínez-García. Ltrust: an adaptive trust model based on lstm for underwater acoustic sensor networks. *IEEE Transactions on Wireless Communications*, 21(9):7314–7328, 2022.
- [28] Jeongah Yoon, Anselm Blumer, and Kyongbum Lee. An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. *Bioinformatics*, 22(24):3106–3108, 10 2006.
- [29] Petrika Gjanci, Chiara Petrioli, Stefano Basagni, Cynthia A Phillips, Ladislau Bölöni, and Damla Turgut. Path finding for maximum value of information in multi-modal underwater wireless sensor networks. *IEEE Transactions on Mobile Computing*, 17(2):404–418, 2017.
- [30] Leonid Maksimovich Brekhovskikh, Yu P Lysanov, and Jurij P Lysanov. *Fundamentals of ocean acoustics*. Springer Science & Business Media, 2003.
- [31] Theodore S Rappaport. *Wireless communications: Principles and practice, 2/E*. Pearson Education India, 2010.

- [32] Yishan Su, Yibo Zhu, Haining Mo, Jun-Hong Cui, and Zhigang Jin. A joint power control and rate adaptation mac protocol for underwater sensor networks. *Ad Hoc Networks*, 26:36–49, 2015.
- [33] Evologics S2c Acoustic Modems. Website, 2023. <https://evologics.de/acoustic-modems>.



Zhaowei Zhang received the bachelor's degree from Hebei University of Technology, China, in 2021. He is working toward the master's degree in College of Intelligence and Computing, Tianjin University, China. His research interests cover underwater Internet of things and underwater wireless sensor networks.



research interests include wireless network, traffic modeling, protocol design, performance evaluation and QoS, and truthful computing in non-cooperative wireless networks.

Chunfeng Liu received the bachelor's degree in mathematics from Shandong Normal University, China, in 2000, and the master's degree in mathematics and the Ph.D. degree in computer science from Tianjin University, China, in 2003 and 2009, respectively. Since 2003, she has been an Assistant Lecturer, Lecturer, and Associate Professor with College of Intelligence and Computing, Tianjin University. She was a Visiting Scholar with the School of Information Technology and Engineering, University of Ottawa, Canada, from 2007 to 2008. Her



computing, computer networks, and information retrieval. She has authored over 80 technical papers in international journals and conferences. She is on the committee board for a couple of international conferences.

Wenyu Qu received the bachelor's and master's degrees from the Dalian University of Technology, China, in 1994 and 1997, respectively, and the Ph.D. degree from the Japan Advanced Institute of Science and Technology, Japan, in 2006. She is currently a Professor with College of Intelligence and Computing, Tianjin University. She was a Professor with Dalian Maritime University, China, from 2007 to 2015. She was an Assistant Professor with the Dalian University of Technology, China, from 1997 to 2003. Her research interests include cloud



2015 and Bell Labs Prize Finalist in 2014 and Semi-Finalist in 2016 and 2019.

Weisi Guo (Senior Member, IEEE) received the M.Eng., M.A., and Ph.D. degrees from the University of Cambridge. He was an Associate Professor with the University of Warwick. He is currently the Chair Professor of human-machine intelligence with Cranfield University. He is also a Turing Fellow with The Alan Turing Institute. He has published more than 130 articles and is PI on over \$3.5m of research grants from EPSRC, H2020, Royal Society, Innovate U.K., and DSTL. His research has won several international awards, including IET Innovation in



Zhao Zhao received the bachelor's degree from Hebei University of Technology City College, China, in 2015, and the master's degree from Hebei University of Engineering, China, in 2018. He is working toward the Ph.D degree in College of Intelligence and Computing, Tianjin University, China. His research interests cover underwater Internet of things and underwater wireless sensor networks.



DSPACE

<https://dspace.org/>

PRObust: a percolation-based robustness optimization model for underwater acoustic sensor networks

Zhang, Zhaowei; Liu, Chunfeng; Qu, Wenyu; Zhao, Zhao; Guo, Weisi

2025-02

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/tnsm.2024.3487956>

Zhang Z, Liu C, Qu W, et al., (2025) PRObust: a percolation-based robustness optimization model for underwater acoustic sensor networks. IEEE Transactions on Network and Service Management, Volume 22, Issue 1, February 2025, pp. 702-717