Maritime-Oriented Network Slicing in O-RAN Integrated Aerial-Terrestrial Networks

Sahar Ammar[1], Wiem Abderrahim[2], and Basem Shihada[1]

Abstract—The deployment of reliable maritime communication systems integrated with terrestrial networks is challenging due to the large maritime regions, the difficulty of deploying conventional base stations at sea, and the heterogeneity of proprietary equipment. In this paper, we propose an AI-based network slicing framework for Open Radio Access Network (O-RAN) integrated aerial-terrestrial maritime networks that incorporates non-tethered and tethered unmanned aerial vehicles (UAVs) and marine buoys. The network provides ubiquitous connectivity while addressing diverse maritime user requirements. Specifically, we leverage network slicing to accommodate the needs of two distinct slices: the maritime infotainment slice, which demands high data rates, and the maritime emergency communication slice, which requires high-reliability and low-latency. Moreover, we adopt virtualization principles to enable flexible deployment approaches for virtualized network functions (VNFs), that can be dynamically scaled and/or migrated across virtualized network nodes. Then, we design a network slicing framework based on a Deep Reinforcement Learning (DRL) that takes into account the characteristics of the maritime environment, and present two algorithms using Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO). Our findings highlight the importance of the integration of aerial and terrestrial networks with network slicing to enhance the energy efficiency of maritime communications while meeting diverse Quality of Service (QoS) requirements.

Index Terms—Maritime Communication, Open-RAN, Integrated aerial-terrestrial Network, Maritime Emergency and Rescue, Network Slicing, Deep Reinforcement Learning (DRL).

1 Introduction

Maritime activities have expanded over the last few decades beyond traditional fishing and maritime transportation to include ocean exploration and climate change research. This created a global marine market size worth 4,420.7 million USD in 2022 and expected to surpass 10,000 million USD in 2033 [1]. These activities require enhanced maritime communications systems that provide ubiquitous connectivity and satisfy the various demands of the passengers, fishermen, and devices on remote ships. More critically, ships and underwater rescue operations require more reliable communication. In fact, statistics reveal that an average of 17% accidents per 100 marine vessels occurred worldwide between 2012 and 2017 [2]. However, establishing such systems remains a challenge that requires further research. This is due to the low user density, the vast ocean areas that should be covered and the difficulty of deploying typical base stations (BSs) in the seas [3]. Conventional maritime communication technologies are based on on-shore BSs providing basic services including text messaging and voice calling. Additionally, satellite-based solutions were used to expand the coverage in the sea [4]. However, they suffer from large propagation distances and restricted on-board resources leading to significant delays and limited data rates. Moreover, international regulations are required to manage the different demands and distribute the licenses; which might delay the deployment and yield political complications [3]. Also, maritime users particularly in fishing villages and small islands cannot afford the use of satellite-based systems. This is due to their high costs and large antennas which cannot be installed on fishing and small boats. Therefore, researchers are dedicating their efforts to develop comprehensive 6G communication systems that extend and complement the terrestrial networks. Specifically, integrated aerial-terrestrial networks have emerged as a promising solution to endorse the coverage and service scope of terrestrial networks. In fact, unmanned aerial vehicles (UAVs) can expand the near-shore coverage, by acting as relaying units to connect marine vessels and on-land BSs. Thanks to their cost-efficiency, simple and flexible deployment, UAVs are suitable to ensure seamless connectivity in the near-shore region. The use of these aerial platforms, including tethered and non-tethered UAVs, have been explored in [5]–[8] to extend the coverage of terrestrial networks and enhance maritime connectivity. Nonetheless, such integrated networks are usually heterogeneous and rely on proprietary equipment. This limits their flexibility and adaptability to different Quality of Service (QoS) demands, which is required to support diversified maritime use cases. For instance, marine passenger infotainment users need a high data rate connectivity, while maritime emergency communication necessitates low latency.

To overcome these inherent challenges of maritime communication, we propose an intelligent network slicing framework built on an integrated aerial-terrestrial maritime network architecture. Incorporating non-tethered UAVs, tethered UAVs, and marine buoys, the proposed architecture offers ubiquitous connectivity and satisfies the requirements of various maritime users. Additionally, we introduce Open

Sahar Ammar and Basem Shihada are with CEMSE Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Makkah Province, Saudi Arabia. Wiem Abderrahim is with University of Carthage, Ecole Supérieure des Communications de Tunis (Sup'Com), LR11TIC05, Réseaux Radio-Mobiles Multimédia (MEDIATRON) Lab and University of Gabes, Ecole Nationale d'Ingénieurs de Gabès (ENIG), Tunisia. E-mail: {sahar.ammar, wiem.abderrahim, basem.shihada}@kaust.edu.sa.

Radio Access Network (O-RAN) concepts in the integrated maritime network to offer openness, intelligence and interoperability. The O-RAN architecture is mainly based on virtualized RAN with disaggregated components and AI-powered controllers [9]–[11]. Specifically, the RAN virtualization through Network Function Virtualization (NFV) can be exploited to improve network programmability, flexibility, and agility. Additionally, to enable multi-vendor deployments, O-RAN promotes the disaggregation of RAN functions into Centralized Unit (CU), Distributed Unit (DU), and Radio Unit (RU) as well as the use of open interfaces and white-box hardware. To efficiently manage the network, O-RAN architecture introduces the RAN Intelligent Controller (RIC) based on software-defined networking (SDN) principles [10]. RIC is responsible for network management, RAN automation, and resource orchestration through the integration of AI technologies. These O-RAN features enable the deployment of the proposed intelligent network slicing framework via the creation of optimized network slices that ensure the coexistence of maritime applications with different QoS requirements. In fact, the O-RAN virtualization supports the flexible deployment of VNFs through scaling and migration in network nodes, while the O-RAN open interfaces facilitate data collection for the training and inference of the proposed DRL-based RAN slicing and VNF deployment algorithms.

1.1 Related Works

In this section, we discuss the existing related works on network slicing in maritime networks and integrated aerial-terrestrial networks. We focus on relevant studies tackling the problems of RAN slicing and virtualized network function (VNF) deployment, particularly VNF scaling and migration[1].

1.1.1 Network Slicing in Maritime Networks

The literature on network slicing in maritime networks is limited with a handful of studies focusing mainly on the design of SDN/NFV-enabled architectures as technology enablers of slicing. Specifically, SDN/NFV-based architectures are proposed for Internet of Things-based maritime transport applications and underwater networks in [16] and [17]. In addition, an SDN-enabled integrated maritime network is designed in [18] to optimize QoS requirements by leveraging a resource scheduling strategy based on deep Q-network. In [19], an SDN-based architecture for underwater acoustic networks is developed with network slicing to improve routing and resource allocation. Moreover, the authors of [20] propose a software-defined maritime fog computing architecture to provide communication and computing services. Despite such efforts, several challenges associated with the use of slicing in maritime networks, including RAN slicing, VNF deployment, and slice management, remain unexplored.

1. We note that the works included in this section employ traditional and RL-based techniques in their studies. Hence, for readers particularly interested in RL-based approaches in network slicing, we refer to the following references for more details in the context of terrestrial networks [12]–[14] and integrated networks [15].

1.1.2 RAN Slicing and VNF Deployment in Integrated Aerial-Terrestrial Networks

The problems of RAN slicing and VNF deployment in terrestrial networks are extensively examined [12], [13]. For instance, a heuristic VNF migration algorithm is proposed in [21] for system cost minimization. Additionally, the authors of [22] develop a VNF scaling scheme based on deep reinforcement learning (DRL) to optimize the latency, service acceptance rate and deployment cost. Meanwhile, other studies jointly investigating RAN slicing and VNF deployment are reported in [23], [24]. In particular, VNF placement, CPU allocation, and traffic routing are jointly considered in [23] to support vertical applications. Using heuristics and convex optimization techniques, the authors the formulated problem of delay minimization. Moreover, in [24], a VNF embedding and RAN slicing strategy is designed to maximize the number of mapped VNFs, utilizing heuristic algorithms. However, these solutions, tailored for terrestrial networks, are not suitable for integrated aerial-terrestrial networks, primarily due to their unique characteristics. In particular, the high mobility of UAVs, constrained onboard resources, and limited power supplies, introduce rapid environment dynamics and additional constraints. This increases the complexity of the RAN slicing and VNF deployment problems in integrated networks, rendering conventional methods inefficient and intractable, and necessitating more intelligent and adaptive approaches. That is why several efforts have been dedicated to tackle these challenges [15]. For instance, dynamic RAN slicing is optimized jointly with UAV positioning in [25] and [26] for improved performance. The authors of [25] target resource consumption minimization using a clique-based algorithm, while the authors of [26] develop a DRL-based method to tackle the formulated multi-objective optimization. Meanwhile, VNF deployment is considered in [27]–[29] for integrated networks. In particular, a hierarchical DRL-based scheme is proposed to jointly minimize the average delay and maximize the energy efficiency through VNF placement, scheduling and migration with UAV trajectory optimization [28]. Moreover, the joint RAN slicing and VNF deployment is examined in [30] for integrated aerial-terrestrial networks. the authors design an iterative algorithm that maximizes the computing resource utilization efficiency to support the three 5G slices. Despite these contributions, further investigations are required to develop intelligent, adaptive, and energy-efficient approaches capable of coping with the features of integrated networks.

To address these research gaps, our work investigates the joint RAN slicing and VNF deployment in integrated aerial-terrestrial maritime networks. Specifically, we consider the characteristics of maritime integrated networks such as the low user density and the clustered distribution, high mobility of UAVs, and their limited resources. Additionally, we focus on energy-efficiency maximization, a critical aspect of such networks, and we take into account the distinct QoS requirements, in terms of throughput, delay and reliability, of each network slice. We highlight that the concurrent satisfaction of these heterogeneous requirements is necessary, presenting significant challenges.

1.2 Main Contributions

In this work, we propose an intelligent network slicing framework for integrated aerial-terrestrial maritime networks. Leveraging O-RAN principles, the proposed architecture offers ubiquitous connectivity and satisfies various maritime user demands. We exploit the concept of RAN virtualization to dynamically deploy network functions in virtualized network elements, specifically UAVs, tethered UAVs and marine buoys. The VNFs can be scaled and/or migrated in these virtualized nodes, featuring different hardware characteristics and energy constraints. In addition, we adopt RAN slicing to serve the desired requirements of two slices. Specifically, we focus on serving the maritime infotainment slice with high data rates needs and the maritime emergency communication slice; which requires high reliability and low delays. In particular, we employ resource slicing to properly allocate the computing and communication resources to serve the two maritime slices. The main contributions of this paper can be summarized as follows:

- We design a network slicing framework for integrated aerial-terrestrial networks, that takes into account the characteristics and requirements of maritime users.
- We leverage RAN virtualization, a key concept of the O-RAN architecture, to flexibly deploy VNFs through scaling and migration in network nodes that offer different resource and energy constraints.
- We use RAN slicing to properly allocate the resources to serve two types of slices namely the high data rate connectivity slice for marine passenger infotainment, and the high reliability emergency communication slice for ships rescue. We perform inter-slice and intra-slice resource management where the computing resources are allocated to each slice (inter-slice allocation) and the communication resources are allocated to each user belonging to each slice (intra-slice allocation).
- We formulate the joint RAN slicing and VNF deployment with UAV trajectory optimization problem to improve the performance of the integrated network. We maximize the overall energy efficiency, which is a key aspect in these networks, and meet the requirements of the slices in terms of data rates, reliability and delay.
- We propose a Deep Reinforcement Learning-based Maritime Network Slicing Framework to tackle the formulated problem exploiting the characteristics of the maritime environment and using two policy gradient algorithms, namely Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO).

2 System Model

The proposed integrated aerial-terrestrial maritime network is illustrated in Fig.1. The architecture is composed of tethered and non-tethered UAVs as well as marine buoys that act as RAN nodes where different VNFs can be deployed. These nodes provide connectivity to the maritime end-users, i.e. user equipment (UEs), on the ships belonging to the marine infotainment slice or emergency slice. Moreover, Fig.1 illustrates how the major components and interfaces of the O-RAN architecture map onto the proposed integrated network, based on the O-RAN Alliance reference architecture [10], [11], to support maritime-oriented slicing. Specifically, the open RU (O-RU) and open DU (O-DU), hosting low-level functionalities, can be deployed on the UAVs and buoys to reduce network latency, by eliminating the open-fronthaul link, fulfilling the low latency requirement of the emergency slice. Meanwhile, the open CU (O-CU) can be deployed on on-land edge/cloud servers since it manages higher-level functions requiring larger computing resources. The O-CU connects to the O-DUs through the F1 interface, which carries user and control planes traffic. In addition, the O-RAN architecture includes two types of RICs, namely the Near-Real-Time (Near-RT) RIC, and the Non-Real-Time (Non-RT) RIC. On the one hand, the Near-RT RIC deals with real-time RAN control and management by enforcing policies provided by the Non-RT RIC, using trained AI models. It can be deployed on edge/cloud servers and it communicates with the O-CU and O-DU nodes through the E2 interface, which allows the Near-RT RIC to send control commands to the O-CU/O-DU and collect network data from them. On the other hand, the Non-RT RIC is responsible for RAN analytics, policy management, and network optimization by training AI algorithms. It can be deployed on regional or national cloud servers and it connects to the Near-RT RIC via the A1 interface, which enables the Non-RT RIC to transfer AI-enabled policies and models, and receive updated network information. Consequently, the RICs enable the O-RAN intelligence required to support the proposed intelligent network slicing framework. Table 1 summarizes the main notations used throughout the paper.

2.1 Integrated-Aerial-Maritime Channel Modeling

In the proposed integrated-aerial-maritime network, four types of wireless channels can be distinguished. This includes (i) the Air-to-Air (AA) channel for the communication between UAVs, (ii) the Air-to-Sea (AS) channel for UAVs to maritime end-users and buoys links, (iii) the Sea-to-Air (SA) channel for UAVs to buoys communication, and (iv) the Sea-to-Sea (SS) channel in for marine buoys and end-users links. The modeling of each channel includes the large-scale fading characterized by the path loss of the dominant Line-of-Sight (LOS) component, and the small-scale fading modeled as a Rician fading [31]. First, the path loss for the AA, AS and SA channels can be expressed using the Free-space path loss model, as follows [32],

$$L_{\mathrm{s_c}}[t] = 10\alpha_{\mathrm{s_c}} \log_{10}\left(\frac{4\pi d_{A,B}[t]}{\lambda}\right) \tag{1}$$

For the SS channel, given that three types of rays can co-exist in this environment, the path loss is modeled using the two-ray model and the three-ray model to account for different link ranges, and it is expressed as [31],

$$L_{\mathrm{SS}}[t] = \begin{cases} 20\log_{10}\left(2L_0 \sin\left(\frac{2\pi z_A z_B}{\lambda d_{A,B}[t]}\right)\right), d_{A,B}[t] \leq d_{\mathrm{b}} \\ 20\log_{10}\left(2L_0\left(1+\Delta_{SS}\right)\right), d_{A,B}[t] \geq d_{\mathrm{b}} \end{cases} \tag{2}$$

where $\Delta_{SS} = 2\sin\left(\frac{2\pi z_A z_B}{\lambda d_{A,B}[t]}\right)\sin\left(\frac{2\pi(z_d - z_A)(z_d - z_B)}{\lambda d_{A,B}[t]}\right)$, $L_0 = \frac{4\pi d_{A,B}[t]}{\lambda}$, z_d is the duct layer height, and $d_{\mathrm{b}} = \frac{4 z_A z_B}{\lambda}$ is the boundary distance the model to employ. Additionally, z_A and z_B are the heights of the transmitter and receiver, $\alpha_{\mathrm{s_c}}$ denotes the path loss exponent and $\lambda = c/f$, with f

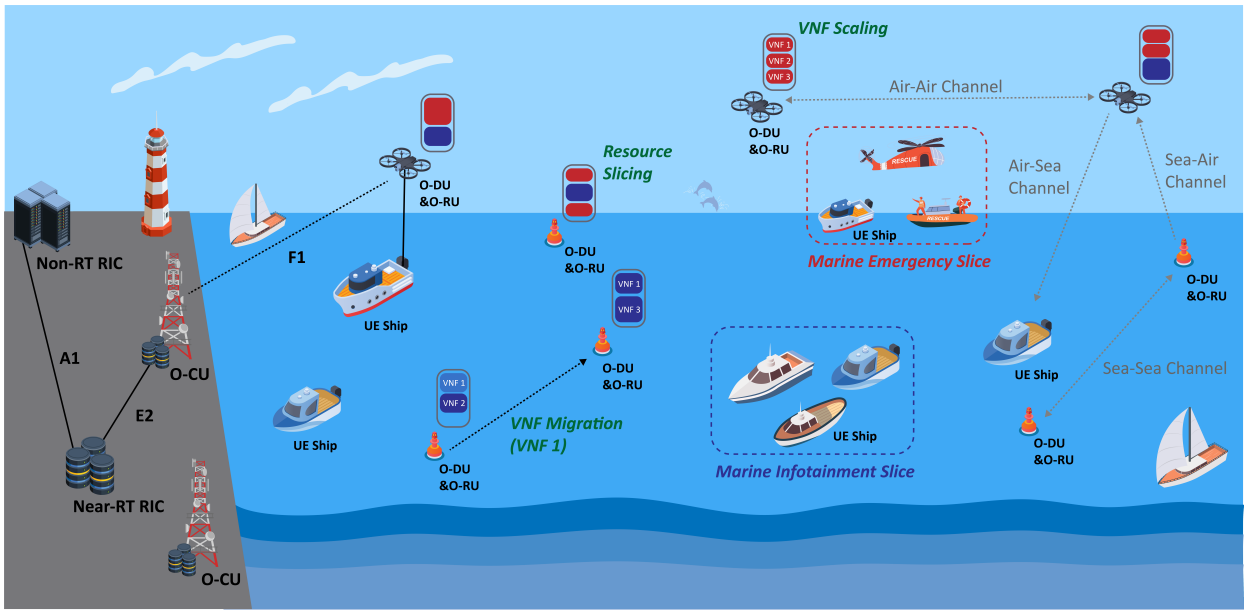


Fig. 1: Illustration of the O-RAN Integrated Maritime Network Architecture.

Notation	Description
T	Set of time slots
S	Set of network slices
F	Set of VNF types
N	Set of non-tethered UAVs
M	Set of tethered UAVs
K	Set of marine buoys
U_s	Set of end-users belonging to slice s
$X_{u_s}[t]$	3D position vector of end-user u_s at time slot t
$X_\zeta[t]$	3D position vector of node $\zeta \in \{N, M, K\}$ at time slot t
C_ζ^{total}	Total CPU capacity at node $\zeta \in \{N, M, K\}$
$P_\zeta^{\text{transmit}}$	Total transmission power at node $\zeta \in \{N, M, K\}$
B_ζ	Bandwidth at node ζ
P_ζ^{flight}	Power needed for the flight of node $\zeta \in \{N, M\}$
$C_{f,i}^{\text{req}}$	CPU capacity required to deploy one VNF instance i of type $f \in F$
$\Lambda_{f,s}^{\zeta}[t]$	Number of VNF instances of type $f \in F$ at node $\zeta \in \{N, M, K\}$ serving slice $s \in S$ at time slot t
$C_{f,s}^{\text{req}}$	Resource requirement of CPU capacity for deploying the VNF of type f to serve slice s
$Q_{P,f,i,s}$	CPU capacity needed by VNF instance i of type f to serve slice s
$Q_{T,f,i,s}$	Data size transmitted by VNF instance i of type f to serve slice s
R_{min}^s	Minimum throughput requirement for slice s
W_{min}^s	Minimum reliability requirement for slice s
D_{max}^s	Maximum delay requirement for slice s
$\eta_{f,s}^{\zeta}[t]$	Integer variable indicating the scaling of VNF instances of type f at node ζ to serve slice s at time slot t
$\mu_{f,i,s}^{\zeta,\zeta'}[t]$	Binary variable indicating the migration of VNF instance i of type f from node ζ to ζ' at time slot t to serve slice s
$c_{f,i,s}^{\zeta}[t]$	CPU capacity allocated to VNF instance i of type f at node ζ to serve slice s at time slot t
$p_{f,i,u_s}^{\zeta}[t]$	Transmission power allocated to VNF instance i of type f at node ζ to serve user u_s at time slot t

TABLE 1: Main Notations.

and c are the frequency and the light velocity. Also, $d_{A,B}[t]$ represents the distance between the transmitter node A and receiver node B, which can be tethered UAVs, non-tethered UAVs, buoys or maritime end-users. Thus, the path gain of the $A-B$ link is:

$$h_{A,B}[t] = 10^{\frac{G_A+G_B-L_{s_c}[t]}{10}}, \tag{3}$$

where $s_c \in \{SS, AS, SA, AA\}$, G_A and G_B are the transmitter and receiver antenna gains. Moreover, to capture the characteristics of the maritime channel, the small-scale channel fading caused by the weak paths resulting from the multiple sea surface reflections, especially in rough sea situations, is modeled as Rician fading with the following distribution [31],

$$f_{\chi_{A,B}[t]}(x) = \frac{x}{\sigma^2} \exp\left(\frac{-\left(x^2+\nu_{A,B}^2\right)}{2\sigma^2}\right) I_0\left(\frac{x\nu_{A,B}}{\sigma^2}\right) \tag{4}$$

where $\nu_{A,B}^2[t] = P_A[t]\left(\frac{\lambda}{4\pi d_{A,B}[t]}\right)^{\alpha_{s_c}} G_A G_B$ and $2\sigma^2$ represent the average received power of the LOS component and the multipath components, respectively. Additionally, $P_A[t]$ is the transmit power and $I_0(.)$ denotes the first kind of modified Bessel function of the 0^{th} order.

2.2 UAV Mobility and Flight Power Consumption

To ensure optimized and efficient network performance, we examine the joint RAN slicing and VNF deployment in conjunction with the non-tethered UAV trajectory design. We assume that the non-tethered UAVs have equal maximum velocity $V = V_{\text{UAV}}$, then the UAVs can travel a maximum distance $d_{\text{UAV}} = \tau V$ between two consecutive time slots. In addition, to avoid collisions between the UAVs including both tethered and non-tethered types, a safety minimum distance d_{safe} should be guaranteed [28]. Thus, two mobility constraints should be fulfilled in the UAV trajectory optimization:

$$C_1: \quad ||X_n[t] - X_n[t-1]|| \leq d_{\text{UAV}}, \quad \forall n \in N. \tag{5}$$

$$C_2: \quad ||X_\zeta[t] - X_{\zeta'}[t]|| > d_{\text{safe}}, \quad \forall \zeta \neq \zeta' \in \{N, M\} \tag{6}$$

where $X_\zeta[t]$ denotes the position vector of node $\zeta = n, m$ corresponding to the n^{th} UAVs and m^{th} T-UAVs. Moreover, we assume that both types of UAVs are rotary-wing UAVs. While the tethered UAVs only hover over their position, the non-tethered UAVs travel at a maximum constant velocity V. Their hovering P_m^{hover} and flight P_n^{flight} powers are [33]:

$$P_m^{\text{hover}} = P_m^{bp} + P_m^{ip}, \tag{7}$$

$$P_n^{\text{flight}} = P_n^{bp}\left(1 + \frac{3V^2}{U_{\text{tip}}^2}\right) + P_n^{ip}\left(\sqrt{1 + \frac{V^4}{4V_0^4}} - \frac{V^2}{2V_0^2}\right)^{1/2} + \frac{1}{2}D_R\rho_{\text{air}}S_{\text{rotor}}A_{\text{rotor}}V^3, \tag{8}$$

where $P_\zeta^{bp} = \frac{P_{\text{drag}}}{8}\rho_{\text{air}}S_{\text{rotor}}A_{\text{rotor}}v_{\text{blade}}^3R_{\text{rotor}}^3$ and $P_\zeta^{ip} = (1 + c_{ip})\frac{(w_\zeta^N)^{3/2}}{\sqrt{2\rho_{\text{air}}A_{\text{rotor}}}}$ denote the blade profile and induced powers of the UAV hovering. P_{drag} and ρ_{air} are the profile drag coefficient and the air density. Also, R_{rotor}, A_{rotor}, and S_{rotor} represent the rotor radius, disc area, and solidity, respectively. v_{blade}, w_ζ^N, and c_{ip} are the blade angular velocity, the aircraft weight, and the incremental correction factor to induced power. U_{tip}, V_0, D_R denote the tip speed of the rotor blade, the mean hovering rotor-induced velocity, and the fuselage drag ratio.

2.3 Quality of Service (QoS) Metrics

To serve the desired slices, we consider three slice QoS metrics, namely the throughput, reliability and delay. First, the overall throughput $R_{f,i,s}^\zeta[t]$ of slice s, derived by summing over the per-user throughput provided by the VNF instance i of type f at node ζ at time slot t, is given by,

$$R_{f,i,s}^\zeta[t] = \sum_{u_s \in U_s} B_\zeta log_2(1 + \gamma_{f,i,u_s}^\zeta[t]) \tag{9}$$

where B_ζ is the bandwidth of a single resource block allocated to one user at node ζ, U_s denotes the set of end-users belonging to slice s, and $\gamma_{f,i,u_s}^\zeta[t]$ is the SNR per user expressed as,

$$\gamma_{f,i,u_s}^\zeta[t] = \frac{h_{\zeta,u_s}[t]\chi_{\zeta,u_s}^2[t]p_{f,i,u_s}^\zeta[t]}{B_\zeta N_0} \tag{10}$$

where $\chi_{\zeta,u_s}[t]$ and N_0 are the Rician fading factor and the noise power spectral density, respectively. Also, $p_{f,i,u_s}^\zeta[t]$ denotes the transmission power allocated to VNF instance i of type f at node ζ to serve user u_s at time slot t. In this work, we assume that intra-cell interference can be overlook given the unique features of maritime environment including the low number of network nodes and the sparsely distributed users. Moreover, we assume that frequency reuse is implemented in our system to ensure that intra-cell interference remains minimal. Moreover, the transmission reliability is obtained using the outage probability defined as,

$$P_{out} = Pr[\gamma_{f,i,u_s}^\zeta[t] \leq \gamma_{f,i,u_s}^{\zeta,\min}] \tag{11}$$

where $\gamma_{f,i,u_s}^{\zeta,\min}$ denotes the minimum SNR value guaranteeing minimal link quality. Given that the channel modeling includes the Rician fading factor $\chi_{\zeta,u_s}[t]$, following the distribution in Eq (4) and assuming that $\sigma = 1$, the P_{out} can be written in terms of the cumulative distribution function (CDF) of a noncentral chi-squared distribution with two degrees of freedom and noncentrality parameter $\nu_{\zeta,u_s}^2[t]$. Hence, the transmission reliability $W_{f,i,s}^\zeta[t]$ supported by the VNF instance i of type f at node ζ to serve slice s at time slot t is,

$$W_{f,i,s}^\zeta[t] = \frac{1}{|U_s|}\sum_{u_s \in U_s} Q_1\left(\nu_{\zeta,u_s}[t], \frac{\sqrt{B_\zeta N_0 \gamma_{f,i,u_s}^{\zeta,\min}}}{\nu_{\zeta,u_s}[t]}\right) \tag{12}$$

where $Q_1(\alpha, \beta)$ denotes the Marcum Q-function of first order, given by,

$$Q_1(\alpha, \beta) = \int_\beta^\infty x \exp\left(-\frac{x^2 + \alpha^2}{2}\right) I_0(\alpha x)dx. \tag{13}$$

Furthermore, the total delay $D_{f,i,s}^\zeta[t]$ of VNF instance i of type f at node ζ serving slice s at time slot t is given by,

$$D_{f,i,s}^\zeta[t] = D_{P,f,i,s}^\zeta[t] + D_{T,f,i,s}^\zeta[t] \tag{14}$$

where $D_{P,f,i,s}^\zeta[t]$ and $D_{T,f,i,s}^\zeta[t]$ are the processing and transmission delays expressed as follows:

$$D_{P,f,i,s}^\zeta[t] = \frac{Q_{P,f,i,s}}{c_{f,i,s}^\zeta[t] + C_{f,i}^{\text{req}}}, \quad D_{T,f,i,s}^\zeta[t] = \frac{Q_{T,f,i,s}}{R_{f,i,s}^\zeta[t]} \tag{15}$$

where $c_{f,i,s}^\zeta[t]$ is the CPU capacity allocated to VNF instance i of type f at node ζ to serve slice s at time slot t. $C_{f,i}^{\text{req}}$ and $Q_{P,f,i,s}$ are the CPU capacity required to deploy one VNF instance i of type f and the CPU capacity needed to serve slice s. Also, $Q_{T,f,i,s}$ denotes the data size transmitted by i to serve slice s. In case of VNF migration, the migration delay $D_{\text{M},f,i}^{\zeta,\zeta'}[t]$ is added to the total delay and it is given by,

$$D_{\text{M},f,i}^{\zeta,\zeta'}[t] = \frac{Q_{M,f,i}}{R_{f,i}^{\zeta,\zeta'}[t]} \tag{16}$$

where $Q_{M,f,i}$ is the data size of VNF instance i of type f and $R_{f,i}^{\zeta,\zeta'}[t] = B_\zeta log_2(1 + \gamma_{f,i}^{\zeta,\zeta'}[t])$ denotes the migration throughput with $\gamma_{f,i}^{\zeta,\zeta'}[t] = \frac{h_{\zeta,\zeta'}[t]\chi_{\zeta,\zeta'}^2[t]p_{f,i}^{\zeta,\zeta'}}{B_\zeta N_0}$ and $p_{f,i}^{\zeta,\zeta'}$ is the transmit power needed for the migration of VNF instance i of type f from ζ to ζ'.

3 Joint RAN slicing and VNF deployment Problem

To optimize the performance of the proposed maritime network, we jointly consider RAN slicing and VNF deployment. In particular, we make a VNF scaling and/or a VNF migration decision while simultaneously allocating the computing and communication resources to serve the desired slices. Additionally, we design the non-tethered UAV trajectory to optimize network operation. Consequently, the optimization variables are defined as follows:

- $\eta_{f,s}^\zeta[t]$: Integer variable indicating the number of VNF instances of type f to add or remove at node ζ to serve slice s at time slot t.
- $\mu_{f,i,s}^{\zeta,\zeta'}[t]$: Binary variable equal to 1 VNF instance i of type f deployed at node ζ at time slot $t-1$ migrates to node ζ' at at time slot t to serve slice s, and 0 otherwise.

- $c^{\zeta}_{f,i,s}[t]$: CPU capacity allocated to VNF instance i of type f at node ζ to serve slice s at time slot t.
- $p^{\zeta}_{f,i,u_s}[t]$: Transmission power allocated to VNF instance i of type f at node ζ to serve user u_s belonging to slice s at time slot t.
- $X_n[t]$: Position vector of the n^{th} non-tethered UAVs at time slot t.

To properly scale and/or migrate the VNF instances to serve slice s, multiple constraints should be satisfied. On the one hand, the computing and communication resource constraints are defined as follows $\forall \zeta, \zeta',\ \forall t$:

$$C_3: \sum_{s\in S}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta}_{f,s}[t]}(1-\mu^{\zeta,\zeta'}_{f,i,s}[t])\cdot(c^{\zeta}_{f,i,s}[t]+C^{\text{req}}_{f,i}) \le C^{\text{total}}_{\zeta} \tag{17}$$

$$C_4: \sum_{s\in S}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta'}_{f,s}[t]}\mu^{\zeta,\zeta'}_{f,i,s}[t]\cdot(c^{\zeta'}_{f,i,s}[t]+C^{\text{req}}_{f,i}) \le C^{\text{total}}_{\zeta'} \tag{18}$$

$$C_5: \sum_{s\in S}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta}_{f,s}[t]}\sum_{u_s\in U_s}(1-\mu^{\zeta,\zeta'}_{f,i,s}[t])\cdot p^{\zeta}_{f,i,u_s}[t] \le P^{\text{transmit}}_{\zeta} \tag{19}$$

$$C_6: \sum_{s\in S}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta'}_{f,s}[t]}\sum_{u_s\in U_s}\mu^{\zeta,\zeta'}_{f,i,s}[t]\cdot p^{\zeta'}_{f,i,u_s}[t] \le P^{\text{transmit}}_{\zeta'} \tag{20}$$

where S and F are the sets of network slices and VNF types. C^{total}_{ζ} and $P^{\text{transmit}}_{\zeta}$ are the total CPU capacity and transmission power at node ζ, respectively, and $\Lambda^{\zeta}_{f,s}[t]$ is the number of VNF instances of type f at node ζ serving slice s at time slot t. The constraints $C_3 - C_6$ ensure that the consumed resources do not exceed the available resources at nodes ζ and ζ'. It is worth noting that C_3 and C_4 guarantee that the consumed CPU capacity remains less or equal than $C^{\text{total}}_{\zeta'}$ in case of the migration of VNF instance i of type f and C^{total}_{ζ} in case of no migration. Equivalently, C_5 and C_6 ensure the same conditions for the transmission power. Moreover, since multiple VNF instance i of type f can be deployed in different nodes, the total allocated computing resources should meet the needs of the served slice s. This is ensured by the following constraint $\forall f \in F, \quad \forall s \in S, \quad \forall t$:

$$C_7: \sum_{\zeta\in\{N,M,K\}}\sum_{i\in\Lambda^{\zeta'}_{f,s}[t]}\mu^{\zeta,\zeta'}_{f,i,s}[t]\cdot c^{\zeta'}_{f,i,s}[t] + \sum_{\zeta\in\{N,M,K\}}\sum_{i\in\Lambda^{\zeta}_{f,s}[t]}(1-\mu^{\zeta,\zeta'}_{f,i,s}[t])\cdot c^{\zeta}_{f,i,s}[t] = C^{\text{req}}_{f,s}, \tag{21}$$

where N, M, and K represent the set of non-tethered UAVs, tethered UAVs, and marine buoys. $C^{\text{req}}_{f,s}$ denotes the resource requirement of CPU capacity for deploying the VNF of type f to serve slice s. On the other hand, the following slice constraints should be satisfied to fulfill the QoS requirements of the infotainment and emergency slices in terms of throughput, reliability and delay $\forall s \in S, \quad \forall t$:

$$C_8: \sum_{\zeta\in\{N,M,K\}}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta'}_{f,s}[t]}\mu^{\zeta,\zeta'}_{f,i,s}[t]\cdot R^{\zeta'}_{f,i,s}[t] + \sum_{\zeta\in\{N,M,K\}}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta}_{f,s}[t]}(1-\mu^{\zeta,\zeta'}_{f,i,s}[t])\cdot R^{\zeta}_{f,i,s}[t] \ge R^{s}_{\min}, \tag{22}$$

$$C_9: \sum_{\zeta\in\{N,M,K\}}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta'}_{f,s}[t]}\mu^{\zeta,\zeta'}_{f,i,s}[t]\cdot W^{\zeta'}_{f,i,s}[t] + \sum_{\zeta\in\{N,M,K\}}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta}_{f,s}[t]}(1-\mu^{\zeta,\zeta'}_{f,i,s}[t])\cdot W^{\zeta}_{f,i,s}[t] \ge W^{s}_{\min}, \tag{23}$$

$$C_{10}: \sum_{\zeta\in\{N,M,K\}}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta'}_{f,s}[t]}\mu^{\zeta,\zeta'}_{f,i,s}[t]\cdot(D^{\zeta'}_{f,i,s}[t]+D^{\zeta,\zeta'}_{\text{M},f,i}[t]) + \sum_{\zeta\in\{N,M,K\}}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta}_{f,s}[t]}(1-\mu^{\zeta,\zeta'}_{f,i,s}[t])\cdot D^{\zeta}_{f,i,s}[t] \le D^{s}_{\max}, \tag{24}$$

where $R^{s}_{\min}$, $W^{s}_{\min}$ and $D^{s}_{\max}$ denote the minimum throughput, reliability, and delay requirements for slice s. While meeting the QoS requirements of both slices, our goal is to maximize the energy efficiency $\Phi_{\text{EE}}[t]$ of the network, defined as follows,

$$\Phi_{\text{EE}}[t] = \sum_{\zeta\in\{N,M,K\}}\sum_{s\in S}\sum_{f\in F}\Big[\sum_{i\in\Lambda^{\zeta'}_{f,s}[t]}\mu^{\zeta,\zeta'}_{f,i,s}[t]\cdot\frac{R^{\zeta'}_{f,i,s}[t]}{P_{\zeta'}[t]} + \sum_{i\in\Lambda^{\zeta}_{f,s}[t]}(1-\mu^{\zeta,\zeta'}_{f,i,s}[t])\cdot\frac{R^{\zeta}_{f,i,s}[t]}{P_{\zeta}[t]}\Big] \tag{25}$$

where $P_{\zeta}[t]$ is the power consumption (flight and service) at node ζ expressed as,

$$P_{\zeta}[t] = P^{\text{flight}}_{\zeta}\cdot\mathbb{1}_{\{N,M\}}(\zeta) + \sum_{s\in S}\sum_{f\in F}\sum_{i\in\Lambda^{\zeta}_{f,s}[t]}\left(\sum_{u_s\in U_s}p^{\zeta}_{f,i,u_s}[t] + \Omega_1\left(c^{\zeta}_{f,i,s}[t]+C^{\text{req}}_{f,i}\right)^{\Omega_2}\right) \tag{26}$$

where $\Omega_1 = 10^{-28}$ and $\Omega_2 = 3$ are parameters related to the CPU model [34]–[36]. We formulate the optimization problem of the joint resource slicing, VNF scaling and migration with UAV trajectory design, as follows,

$$(\mathbf{P}): \max_{\eta^{\zeta}_{f,s}[t],\mu^{\zeta,\zeta'}_{f,i,s}[t],c^{\zeta}_{f,i,s}[t],p^{\zeta}_{f,i,u_s}[t],X_n[t]} \frac{1}{T}\sum_{t\in T}\Phi_{\text{EE}}[t] \quad s.t \quad C_1 - C_{10}, \quad \mu^{\zeta,\zeta'}_{f,i,s}[t]\in\{0,1\}, \quad \eta^{\zeta}_{f,s}[t]\in\mathbb{Z}. \tag{27}$$

where T is the set of time slots. The optimization problem $(\mathbf{P})$ can be simplified to consider either the VNF scaling or the VNF migration with the resource slicing and the UAV trajectory design. This yields two special cases with problems $(\mathbf{P_S})$ and $(\mathbf{P_M})$,

$$(\mathbf{P_S}): \max_{\eta^{\zeta}_{f,s}[t],c^{\zeta}_{f,i,s}[t],p^{\zeta}_{f,i,u_s}[t],X_n[t]} \frac{1}{T}\sum_{t\in T}\Phi_{\text{EE}}[t] \quad s.t \quad C_1 - C_3, C_5, C_7 - C_{10}, \quad \eta^{\zeta}_{f,s}[t]\in\mathbb{Z}. \tag{28}$$

$$(\mathbf{P_M}): \max_{\mu^{\zeta,\zeta'}_{f,i,s}[t],c^{\zeta}_{f,i,s}[t],p^{\zeta}_{f,i,u_s}[t],X_n[t]} \frac{1}{T}\sum_{t\in T}\Phi_{\text{EE}}[t] \quad s.t \quad C_1 - C_{10}, \quad \mu^{\zeta,\zeta'}_{f,i,s}[t]\in\{0,1\}. \tag{29}$$

4 Deep Reinforcement Learning-based Maritime Network Slicing Framework

Since the formulated problem is a mixed-integer nonlinear optimization problem with multiple constraints, it is classified as NP-hard. In addition, the proposed integrated maritime network is characterized by its large-scale topology, dynamic environment, and time-variant traffic demand. This increases the complexity of the joint optimization problem. Consequently, conventional model-based optimization methods can no longer provide the necessary efficiency and optimality. Hence, we develop a DRL-based framework to tackle the problem of dynamic joint RAN slicing and VNF deployment with UAV trajectory optimization.

4.1 Background on Reinforcement Learning

RL is a branch of machine learning that is based on sequential learning where an agent learns to make decision by interacting with an environment, with the goal of maximizing a cumulative reward [37]. RL algorithms can be categories into value-based methods and policy gradient methods. On the one hand, value-based approaches, including Q-learning and Deep Q-Networks (DQNs), utilize value functions to implicitly optimize their policies. In fact, they select the actions that maximizes the value function, which is an estimation of the expected cumulative reward that the agent can obtain from a specific state (or state-action pair), under a particular policy. On the other hand, policy gradient methods, such as REINFORCE and Advantage Actor-Critic (A2C), explicitly optimize a policy by representing it as a parameterized function. The policy's parameters are updated it through gradient ascent in the direction of increased expected reward. Traditional RL algorithms (e.g. Q-learning and REINFORCE) struggle with high-dimensional state-action spaces and dynamic environments. Thus, Deep RL (DRL) was developed to overcome these limitations and broaden the application range of RL [38]. DRL algorithms, such as DQN, Deep Deterministic Policy Gradient (DDPG), A2C, and Proximal Policy Optimization (PPO), combine the concepts of RL with deep neural networks (DNNs). In particular, DQN [39] was introduced as an extension to Q-learning to handle high-dimensional state spaces by employing DNNs as value function estimators. Additionally, multiple variants of DQN were developed to enhance its performance, including Double DQN (DDQN) and Dueling DQN. For instance, DDQN uses an online network for action selection and a target network for Q-value evaluation, which addressed the overestimation bias issue of DQN. Meanwhile, DDPG [40] was designed to deal with continuous actions using a deterministic policy gradient and an actor-critic architecture. DDPG also suffers from overestimation bias which was handled by the Twin Delayed DDPG (TD3) algorithm using two critic networks for Q-value estimation. In addition, Soft Actor–Critic (SAC) was designed for continuous actions using the TD3 twin architecture combined with stochastic policies and entropy regularization.

Although these off-policy algorithms alleviate the limitations of traditional RL, they still face multiple challenges that are better addressed by on-policy approaches. Table 2 compares the properties of these methods. In particular, while they offer improved sample efficiency, off-policy algorithms, such as DQN and DDPG, suffer from multiple issues, in terms of stability, adaptability, and exploration, especially when dealing with complex environments such as in the considered problem. In fact, their reliance on experience replay buffers in the training process makes them more prone to instability and prevents them from adapting to complex dynamic environments. Specifically, the samples, stored in the buffers, include data collected under both new and old policies, which can destabilize the network's updates. This causes oscillations in the value estimates and increases the algorithms sensitivity to hyper-parameters. Moreover, they select actions based on deterministic or ϵ-greedy mechanisms that require hyper-parameter tuning, leading to limited exploration. Meanwhile, on-policy approaches, including A2C [41] and PPO [42], address these challenges by learning using data generated by the current policy, resulting in lower sample efficiency compared to off-policy agents. These actor–critic algorithms employ stochastic policies and entropy regularization, which improve their exploration capabilities, allowing them to handle discrete and continuous action spaces, as well as complex stochastic environments. In addition, they update their policies with newly collected rollouts, by running multiple environment instances (workers) in parallel to accelerate sample collection and ensure data decorrelation, which improves their sample efficiency. This also enhances their learning stability, adaptability, and robustness to hyper-parameter tuning. Consequently, on-policy algorithms are more suitable for complex highly dimensional and dynamic environments which is why they are commonly utilized in wireless communication applications. Therefore, we propose two DRL algorithms based on A2C and PPO in this work to tackle the joint RAN slicing and VNF deployment problem.

Property	On-Policy Algorithms	Off-Policy Algorithms
Policy update	Learning through data generated by the current policy.	Learning through data stored in the replay buffer and generated by different policies.
Training stability	Higher stability because policy updates rely on freshly collected samples.	Susceptible to instability because policy updates rely on data collected under both new and old policies.
Adaptability to dynamic environments	Higher adaptability thanks to the reliance on recent interactions with the environment.	Lower adaptability because of the use of old transitions from the buffer.
Exploration capability	Inherent exploration thanks to the use of stochastic policies and entropy regularization.	Limited exploration using mechanisms, such as $\epsilon-$greedy and additive noise, relying on external hyper-parameters.[2]
Sample efficiency	Lower sample efficiency because only fresh data is used.	Higher sample efficiency because data stored in the buffer can be reused.
Sensitivity to hyper-parameters	Lower sensitivity to hyper-parameter tuning.	Higher sensitivity to hyper-parameter tuning.
Examples	A2C, PPO, TRPO	DQN, DDPG, SAC

TABLE 2: Comparison of off-policy and on-policy algorithms [37], [39]–[43].

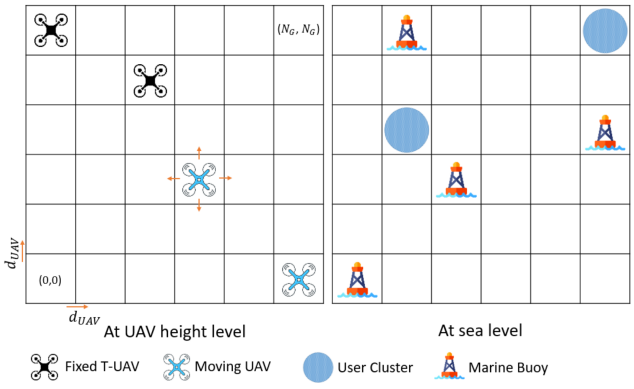


Fig. 2: Illustration of the maritime area in a 2D grid.

4.2 Proposed DRL-based framework

In this section, we present our DRL-framework for joint RAN slicing and VNF deployment with UAV trajectory optimization. First, the formulated problem in (3) considers the individuals on the ships as the end-users. However, this results in extremely large state and action spaces. Meanwhile, these end-users are sparsely distributed in the vast marine region, but they are also grouped in small areas, forming dispersed clusters [44]. Hence, we exploit the cluster property of the maritime environment and consider the ships as the end-users in the Markov Decision Process (MDP) formulation. Then, we formulate the proposed optimization problem as an RL problem by defining the state and action spaces as well as the reward function. We also explain the training process for the proposed A2C- and PPO-based RAN slicing and VNF deployment algorithms.

4.2.1 The State space

At timeslot t, the state $S[t]$ describing the environment is composed of the 3D position vectors $X_\zeta[t]$ of the UAVs, T-UAVs, and buoys, the VNF instances placement $\delta^\zeta_{f,i,s}[t]$ and the number of VNF instances $\Lambda^\zeta_{f,s}[t]$ at each node. In addition, the state includes the ships information consisting of a 3D position vector $X_{u_s}[t]$ and a binary indicator $I_s[t]$ associating the ship with the corresponding slice s. Thus, the state is given by $S[t] = \{X_\zeta[t], \delta^\zeta_{f,i,s}[t], \Lambda^\zeta_{f,s}[t], X_{u_s}[t], I_s[t]\}$. Moreover, we quantize the large maritime area into squares forming a N_G by N_G grid, as illustrated in Fig.2. Then, we represent the position vectors as a tuple (x, h) where x is an integer indicating the square to which the ship u_s or the node ζ belong and h is a binary indicating the sea and air levels. This further simplifies the state representation.

4.2.2 The Action space

The actions of the RL agent involve the combinations of the VNF deployment decisions, including the discrete scaling actions $\eta^\zeta_{f,s}[t]$ and the binary migration actions $\mu^{\zeta,\zeta'}_{f,i,s}[t]$, the RAN slicing decisions consisting of the continuous actions $c^\zeta_{f,i,s}[t]$ representing the CPU capacity and $p^\zeta_{f,i,u_s}[t]$ indicating the transmission power. Additionally, the action space include the continuous actions $\tilde{X}_n[t]$ used for the trajectory optimization of the n^{th} non-tethered UAVs. Thus, at timeslot t, the actions of the RL agent are defined as $A[t] = \{\eta^\zeta_{f,s}[t], \mu^{\zeta,\zeta'}_{f,i,s}[t], c^\zeta_{f,i,s}[t], p^\zeta_{f,i,u_s}[t], \tilde{X}_n[t]\}$. The actions $A[t]$ include both continuous and discrete components, adding complexity when applying RL algorithms. Thus, we discretize the continuous actions to address this issue and simplify the action space representation. First, we exploit the grid in Fig.2 simplifying the maritime area to convert $\tilde{X}_n[t]$ to a discrete action describing the UAV movement from one square to another. This results into five actions $\tilde{X}_n[t] = \{$left, right, up, down, none$\}$ where the UAV can travel d_{UAV} at each timesolt t. Then, we quantize the RAN slicing actions into two levels in an on/off fashion. So, the RL agent can either select a minimum or a maximum value for the computing and communication resource slicing. This quantization step unifies the action space and facilitates the use of RL algorithms.

4.2.3 The Reward function

We design the reward function $R[t]$ to maximize the energy efficiency at timeslot t while satisfying the constraints, using a weighted penalty method [45]. The reward is defined as,

$$R[t] = \omega_{\text{obj}} \cdot \Phi_{\text{EE}}[t] - \sum_{C \in \Gamma} \omega_c \cdot max\{0, C[t] - C_{max}\} \quad (30)$$

where ω_{obj} and ω_c are the weights balancing between the objective function and the constraints. Also, $C[t]$ and C_{max} represent the constraints when written in form of $C[t] \leq C_{\text{max}}$, and Γ is the set of constraints defined in Section 3.

4.2.4 Training process

A2C [41] and PPO [42] simultaneously train two neural networks; an actor network, which selects actions based on the learned policy using probability distribution, and a critic network, which evaluates these actions by estimating the value function. The two algorithms use advantage functions $A_t(s_t, a_t)$ to measure how beneficial an action a_t is given a state s_t. A2C utilizes a short-horizon n-step trajectory to compute A_t, while synchronously averaging the gradients from the parallel workers for the policy update. This reduces the gradients variance, accelerates the training, and allows the policy to quickly adapt to the environment dynamics. Meanwhile, PPO employs the Generalized Advantage Estimation (GAE) to compute advantages through long-horizon trajectories and introduces a clipping mechanism that constrains policy updates. This prevents excessively large steps improving learning stability and robustness at the cost of longer convergence. These key differences in the training process allow A2C to converge faster and present better overall performance compared to PPO in our setting, as demonstrated by the simulation results in the following section.

- *Training process for A2C:*

The A2C-based approach is presented in Algorithm 1. At each timestep t, the agent selects an action a_t according to its policy π_α, given the current state s_t, as shown in lines 5–10. Specifically, the action selection of A2C is based on stochastic policies where the agent samples an action a_t from a discrete probability distribution $\pi_\alpha(a_t|s_t)$ given by,

$$\pi_\alpha(a_t|s_t) = \frac{e^{x_{a_t}}}{\sum_a e^{x_a}} \quad (31)$$

2. SAC is an exception among off-policy algorithms since it uses stochastic policies and entropy regularization for exploration.

where x_{a_t} denotes the logit for action a_t. The logits are the output of the actor network and represent non-normalized scores for each action, which are transformed into probabilities using the softmax function defined in (31). Then, the agent receives a reward r_t, and the next state s_{t+1}. These trajectories (a_t, s_t, r_t, s_{t+1}) are stored into batches for updating the actor and critic networks, as shown in lines 11–16 in Algorithm 1. Then, the return G_t^{A2C}, defined in line 12, and the state-value function $V_\psi(s_t)$, estimated by the critic, are used to derive the advantage function A_t^{A2C}, given by,

$$A_t^{A2C}(s_t, a_t) = G_t^{A2C} - V_\psi(s_t) \tag{32}$$

Then, using gradient ascent, the actor network is updated in line 14 by maximizing the policy objective given by,

$$\mathcal{L}_{\text{actor}}^{A2C}(\alpha) = \mathbb{E}_t\left[A_t^{A2C}(s_t, a_t)\log \pi_\alpha(a_t|s_t)\right] \tag{33}$$

where $\log \pi_\alpha(a_t|s_t)$ are the log probability of action a_t given state s_t under the policy π_α. Simultaneously, using gradient descent, the critic network is updated in line 15 by minimizing the critic loss expressed as,

$$\mathcal{L}_{\text{critic}}^{A2C}(\psi) = \mathbb{E}_t\left[\left(G_t^{A2C} - V_\psi(s_t)\right)^2\right] \tag{34}$$

- *Training process for PPO:*

The PPO-based approach is presented in Algorithm 2. Following similar training process as A2C, the PPO agent interacts with the environment as shown in lines 5-10. Then, the algorithm updates its actor and critic networks using the collected batches of trajectories as shown in lines 11-16. On the one hand, the PPO actor is updated in line 14 by maximizing the policy objective, which includes the clipped surrogate objective and the entropy loss term and given by,

$$\begin{aligned}\mathcal{L}_{\text{actor}}^{PPO}(\theta) = \mathbb{E}_t[\min(\rho_t(\theta)A_t^{PPO}, \text{clip}(\rho_t(\theta), 1-\upsilon,\\ 1+\upsilon)A_t^{PPO})] + \tau_{\text{exp}}\,\mathbb{E}_{\pi_\theta}[\log \pi_\theta(a|s_t)]\end{aligned} \tag{35}$$

where $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta\text{old}}(a_t|s_t)}$ represents the probability ratio between the old and the new policies, τ_{exp} denotes the entropy coefficient and υ is the clipping hyperparameter. Moreover, A_t^{PPO} is derived using GAE and given by [46],

$$A_t^{PPO}(s_t, a_t) = \sum_{k=0}^{\infty}(\gamma_{PPO}\lambda_{GAE})^k\delta_{t+k} \tag{36}$$

where $\delta_t = r_t + \gamma_{PPO}V_\phi(s_{t+1}) - V_\phi(s_t)$. On the other hand, the critic is updated in line 15 via the minimization of the value loss $\mathcal{L}_{\text{critic}}^{PPO}(\phi)$, which has similar expression to A2C, defined in (34), with $G_t^{PPO} = A_t^{PPO} + V_\phi(s_t)$. These synchronous updates allow the actor to improve its action selection and the critic to better estimate the value function.

Algorithm 1: A2C-based RAN slicing and VNF deployment algorithm

1: **Input** the environment and the hyperparameters including the number of episodes N_{ep}, the learning rates ϵ_α^{A2C} and ϵ_ψ^{A2C}, and the discount factor γ_{A2C}.
2: **Initialize** the actor network π_α, the critic network V_ψ.
3: **for** episode = 1 to N_{ep} **do**
4: Observe the initial state s_1
5: **for** t = 1 to n **do**
6: Compute action probabilities $\pi_\alpha(a_t|s_t)$ using (31).
7: Select the action $a_t \sim \pi_\alpha(\cdot|s_t)$
8: Receive the reward r_t, and the next state s_{t+1}
9: Store the trajectory (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$.
10: **end for**
11: **for** each trajectory in $\mathcal{D}$ **do**
12: Compute the return $G_t^{A2C} = \sum_{k=0}^{n-1}\gamma_{A2C}^k r_{t+k} + \gamma_{A2C}^n V_\psi(s_{t+n})$
13: Compute the advantage A_t^{A2C} using (32).
14: Update the actor network by maximizing the policy objective $\mathcal{L}_{\text{actor}}^{A2C}(\alpha)$ in (33), using gradient ascent:
$$\alpha \leftarrow \alpha + \epsilon_\alpha^{A2C}\nabla_\alpha\mathcal{L}_{\text{actor}}^{A2C}(\alpha)$$
15: Update the critic network by minimizing the critic loss $\mathcal{L}_{\text{critic}}^{A2C}(\psi)$ in (34), using gradient descent:
$$\psi \leftarrow \psi - \epsilon_\psi^{A2C}\nabla_\psi\mathcal{L}_{\text{critic}}^{A2C}(\psi)$$
16: **end for**
17: **end for**

4.3 Convergence and Complexity Analysis

As policy gradient algorithms, the theoretical foundation of A2C and PPO is built on the Policy Gradient Theorem [37]. These methods learn a parameterized policy $\pi_\beta(a \mid s)$ by maximizing an objective $\mathcal{J}(\beta)$, which is a performance measure generally defined as the expected discounted return. They update their policies through gradient ascent, which requires the computation of the policy gradient $\nabla_\beta\mathcal{J}(\beta)$. As proven in [37], the Policy Gradient Theorem offers an analytic expression for $\nabla_\beta\mathcal{J}(\beta)$ that is independent of the specifics of the environment and it is given by,

$$\nabla_\beta\mathcal{J}(\beta) = \mathbb{E}_{s,a\sim\pi_\beta}[\nabla_\beta \log \pi_\beta(a|s)Q^{\pi_\beta}(a|s)] \tag{37}$$

where $Q^{\pi_\beta}(a|s)$ is the action-value function under policy π_β. In practice, the Q-values $Q^{\pi_\beta}(a|s)$ are estimated and different algorithms employ different estimation techniques such as using the critic network for actor-critic methods. In fact, the Policy Gradient Theorem allows the replacement of the $Q^{\pi_\beta}(a|s)$ with the advantage function $A_t(s,a)$ in the policy gradient (**??**) without changes in the expected gradient. This substitution reduces the gradients variance and improves training stability. Furthermore, this theorem ensures that the policy updates are in the ascent direction of the objective $\mathcal{J}(\beta)$. Consequently, under the standard assumptions of the step size α_t (i.e. $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$), the stochastic approximation theory guarantees that these algorithms converge almost surely to the stationary points of $\mathcal{J}(\beta)$. Therefore, policy gradient algorithms, including A2C and PPO, are guaranteed to converge to locally optimal policies [37], [47], [48]. However, global optimality of these methods, particularly when deep neural networks are used, remain an open research issue.

The computational complexity of PPO and A2C is dominated by the value function evaluation and the policy update steps, which involve forward passes through the critic and actor networks. Hence, the complexity depends on their network architectures which are defined by the state and action spaces dimensionality and the structure

Algorithm 2: PPO-based RAN slicing and VNF deployment algorithm

Input the environment and the hyperparameters including the number of episodes N_{ep}, the learning rates ϵ_θ^{PPO} and ϵ_ϕ^{PPO}, the discount factor γ_{PPO}, the GAE parameter λ_{GAE}, the entropy coefficient τ_{exp}, and the clipping hyperparameter υ.
Initialize the actor network π_θ, the critic network V_ϕ.
for episode = 1 to N_{ep} **do**
  Observe the initial state s_1
  for t = 1 to T **do**
    Compute action probabilities $\pi_\theta(a_t|s_t)$ using (31).
    Select the action $a_t \sim \pi_\theta(\cdot|s_t)$
    Receive the reward r_t, and the next state s_{t+1}
    Store the trajectory (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$.
  end for
  for each trajectory in $\mathcal{B}$ **do**
    Compute the advantage A_t^{PPO} using (36).
    Compute probability ratio $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta\text{old}}(a_t|s_t)}$.
    Update the actor network by maximizing the policy objective $\mathcal{L}_{\text{actor}}^{PPO}(\theta)$ in (35), using gradient ascent:
$$\theta \leftarrow \theta + \epsilon_\theta^{PPO} \nabla_\theta \mathcal{L}_{\text{actor}}^{PPO}(\theta)$$
    Update the critic network by minimizing the critic loss $\mathcal{L}_{\text{critic}}^{PPO}(\phi)$ in (34), using gradient descent:
$$\phi \leftarrow \phi - \epsilon_\phi^{PPO} \nabla_\phi \mathcal{L}_{\text{critic}}^{PPO}(\phi)$$
  end for
end for

of the hidden layers. The state size D_S is given by $(N + M + K)(2 + SF + \Lambda_{max}) + 3U_s$ and the action size is D_A is given by $(N + M + K)(2SF + 2\Lambda_{max}(1 + U_s)) + 5N$, where Λ_{max} is the maximum number of VNF instances per network node. We assume that the two algorithms have the same architecture where the actor and critic networks include H_a and H_c hidden layers with $N_{a,h}, h = 1..H_a$ and $N_{c,h}, h = 1..H_c$ neurons, respectively. The complexity of one forward pass of the actor and critic network is given by $O_{actor} = O(D_S N_{a,1} + \sum_{h=1}^{H_a} N_{a,h-1} N_{a,h} + D_A N_{a,H_a})$ and $O_{critic} = O(D_S N_{c,1} + \sum_{h=1}^{H_c} N_{c,h-1} N_{c,h} + N_{c,H_c})$, respectively. Therefore, the computational complexities of the proposed A2C and PPO approaches are $O(T_{A2C}(O_{actor}^{A2C} + O_{critic}^{A2C}))$ and $O(T_{PPO}(O_{actor}^{PPO} + O_{critic}^{PPO}))$, respectively. T_{A2C} and T_PPO are the number of timesteps per rollout for A2C and PPO. Once the training is completed, only a single forward pass through the trained models is needed for action selection, given the network information. This computation allows the algorithm to be executed with a negligible latency that is compatible with the RIC control intervals.

5 Results and Analysis

In this section, we present the simulation results evaluating the performance of the proposed DRL-based network slicing framework. First, we consider a 25 Km^2 maritime area having 5 marine buoys that can be used for communication purposes. We assume that the ships and the buoys have the same height $z_{u_s} = z_k = 2m$, $k \in K$ while the heights of the tethered and non-tethered UAVs are $z_m = 112m$, $m \in M$ and $z_n = 115m$, $n \in N$, respectively. Also, we set the safety distance between the UAVs to $d_{\text{safe}} = 3m$. Additionally, the bandwidth, CPU capacity and transmit power of the buoys are $B_k = 2$ MHz, $C_k^{\text{total}} = 10^9$ (cycles/s), and $P_k^{\text{transmit}} = 30$ dBm [49]. Moreover, the CPU capacity required to serve one slice s and to deploy one VNF instance of type f are $C_{f,s}^{\text{req}} = 10^9$ cycles/s and $C_{f,i}^{\text{req}} = 2.5\ 10^8$ cycles/s. The data sizes required to serve the infotainment and the emergency slice are 2 Mbits and 2 Kbits. The CPU capacity required by one VNF instance i to serve one slice s is $Q_{P,f,i,s} = 10^7$ cycles/s. In case of migration, the data size of the migrated VNF instance and the transmit power needed for it migration are $Q_{M,f,i} = 100$ Kbits and $p_{f,i}^{\zeta,\zeta'} = 16$ dBm. The main simulation settings are summarized in Table 3.

Parameter	Value
Path loss exponents $\alpha_{AA}, \alpha_{AS}, \alpha_{SA}, \alpha_{SS}$	1.9, 2.2, 2.2, 2
Carrier frequency f	2 GHz
Noise power spectral density N_0	-124 dBm/Hz
UAVs antenna gain G_n, G_m	20 dB
Buoys and ships antenna gain G_k, G_{u_s}	10 dB
UAV velocity V	$10m/s$
UAV weight w^N	$20N$
Profile drag coefficient P_{drag}	0.012
Blade angular velocity v_{blade}	300 rad/s
incremental correction factor c_{ip}	0.1
Air density ρ_{air}	1.225 Kg/m^3
Tip speed of the rotor blade U_{tip}	$120m/s$
mean hovering rotor-induced velocity V_0	$4.03m/s$
fuselage drag ratio D_R	0.6
Rotor radius R_{rotor}	$0.4m$
Disc area A_{rotor}	$0.503m$
Solidity S_{rotor}	0.05

TABLE 3: Main simulation Parameters [32], [33], [50], [51].

Furthermore, we consider a decision making timeslot of length $t = 10s$, we train the DRL agents using $N_{env} = 8$ parallel environments, and we evaluate their performance by averaging over $N_{ep} = 20$ episodes with $T = 100$ steps. This captures the stochasticity of the environment while offering accelerated training and effective lightweight evaluation. Regarding the DRL parameters, we fine-tune the hyperparameters of the proposed A2C- and PPO-based RAN slicing and VNF deployment algorithms through extensive experimentation. First, we adopt the same actor and critic networks architecture for the two algorithms to ensure a fair comparison. Specifically, both actor and critic networks are designed using four layers with 128, 64, 64, and 128 neurons. This architecture allows the RL agent to handle high-dimensional state-action spaces and deal with the non-linear dependency between the different actions. The learning rates for the A2C actor and critic are $\epsilon_\alpha^{A2C} = \epsilon_\psi^{A2C} = 0.0007$. Meanwhile, we adopt an adaptive learning rate for PPO-based algorithm, balancing between convergence and training speed. The learning rates for the PPO actor and critic are $\epsilon_\theta^{PPO} = \epsilon_\phi^{PPO} = \epsilon_{inital} \cdot \exp(-d_{rate}(1 - E))$ where $\epsilon_{inital} = 0.0005$ is the initial learning rate, $d_{rate} = 0.99$ is the exponential decay and E training progress. Moreover, we tune the discount factor $\gamma_{PPO} = \gamma_{A2C} = 0.99$, the PPO entropy coefficient $\tau_{\text{exp}} = 0.005$, and the PPO clipping hyperparameter $\upsilon = 0.3$. In our simulations, we solve the

three problems of joint RAN slicing and VNF deployment as discussed in Section 3. Throughout this section, we refer to these cases respectively as Migration-based deployment ($\mathbf{P_M}$), Scaling-based deployment ($\mathbf{P_S}$) and Hybrid deployment ($\mathbf{P}$).

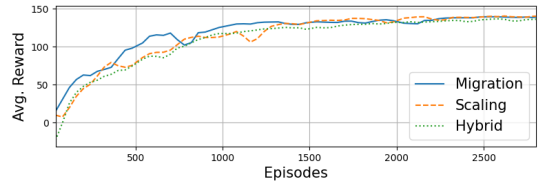
5.1 Convergence Performance

To understand the convergence of the proposed DRL-based algorithms, we examine the average training reward for the three problems, illustrated in Fig. 3 (a) and (b). First, we observe that the A2C agent converges faster than the PPO agent requiring about 2700 episodes. This aligns with the training process of A2C that adopts short-horizon policy update, as discussed in previous section. In addition, A2C achieves a slightly higher total cumulative reward compared to PPO agent. This is due to A2C's policy updates which are more rapid and aggressive than PPO, allowing it to improve its policy in early training, take advantage of the initial rewards, and adapt to the environment dynamics. Nonetheless, as A2C explores various policies, the reward exhibits more fluctuations suggesting lower learning stability particularly in the Migration case. Meanwhile, the PPO agent shows slower convergence taking up to 20000 episodes, which is due to the use of long-horizon trajectories for policy updates. In addition, PPO presents improved stability compared to A2C, which is caused by the clipping mechanism that ensures the changes in the policy remain conservative. This reduces the reward fluctuations and stabilizes the learning at the cost of training time. Furthermore, we compare the reward performance of the proposed DRL-based algorithms with four benchmarks, as demonstrated in Fig. 3 (c), (d), (e) and (f). In fact, since the optimization problem formulated in Section 3 is NP-hard, deriving a global optimal solution can be untractable and inefficient. Thus, we consider a metaheuristic method based on the genetic algorithm, a static baseline with one VNF instance per slice and equal resource slicing, as well as the greedy and random benchmarks. The DRL agents, after convergence, achieve significantly higher reward compared to all benchmarks. Specifically, the A2C-based and the PPO-based approaches achieve an average reward of approximately $+150$ and around $+100$, respectively, whereas the metaheuristic benchmark yields a highly fluctuating performance with rewards oscillating around $+50$. Meanwhile, the static baseline shows a consistently negative reward around -70, while the greedy and random methods present substantial variability and severely negative rewards, frequently dropping below -50. This indicates persistent energy inefficiencies and constraints violations including resources and QoS requirements. Consequently, these results show the superiority of DRL-based solutions over relevant benchmarks in solving complex dynamic optimization problems such as joint RAN slicing and VNF deployment in integrated aerial-terrestrial maritime networks.

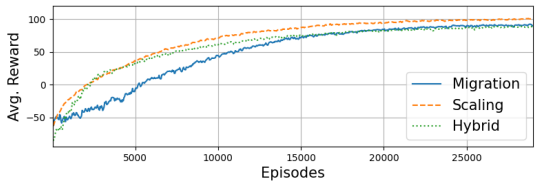
5.2 Impact of Aerial Network Settings

After training the DRL agents, we investigate the performance of our model from a communication perspective. Extensive simulations were conducted while varying the types of UAVs (i.e. tethered, non-tethered) and their ratio within a fixed total number of UAVs. For each simulation, we focus on analyzing four network performance indicators; energy efficiency, achievable throughput of the infotainment slice, delay of the emergency slice and guaranteed reliability to the emergency slice. To establish a fair comparison and obtain accurate insights, we maintain the same total power for the three scenarios, independently of the number of network nodes. We begin by exploring the energy efficiency of our integrated aerial-terrestrial maritime network as illustrated in Fig.4. We notice that the A2C-based algorithms offers superior energy efficiency gains compared to the PPO agent. This is due to the rapid policy updates of the A2C that allows it to better adapt to the environment dynamics when the three approaches are considered across different UAV deployments. In contrast, while it offers stable learning, the clipping mechanism restricts the PPO-based algorithms leading to lower efficiency. Moreover, when comparing deployment strategies, the Scaling-based approach outperforms the Hybrid and the Migration-based schemes. Specifically, the PPO's long-horizon updates can effectively handle the gradual changes in the environment introduced by the scaling actions, but struggle to track the more abrupt shifts caused by the migration actions.

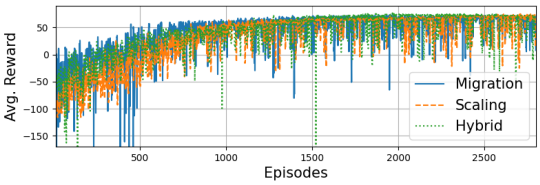
Furthermore, we investigate the performance of the DRL agents in terms of the slice QoS requirements as illustrated in Fig.5, Fig.6, and Fig.7. First, we examine the throughput of the infotainment slice supported by the proposed integrated maritime network. As shown in Fig.5, we notice that increasing the ratio of tethered UAVs substantially enhances the throughput for both DRL agents thanks to the increased proximity of the UAVs to the end-users. Additionally, we observe that the three deployment schemes can fulfill the infotainment slice needs, as long as one tethered UAV is deployed, by providing throughput higher than the minimum requirement R^s_{min}. Moreover, when the A2C agent is used, the Scaling approach offers increased infotainment slice throughput, as depicted in Fig.5.(a). Meanwhile, when the PPO algorithm is applied, the Hybrid deployment shows improved throughput performance compared to other schemes, as illustrated in Fig.5.(b). Second, we study the delay of the emergency slice provided by the integrated aerial-terrestrial maritime network. As shown in Fig.6, we observe that the deployment of supplementary non-tethered UAVs contributes reduced delays thanks to their unrestricted mobility. On the one hand, the A2C agent offers enhanced delay performance across the deployment schemes compared to the PPO agent. Specifically, the three approaches satisfy the emergency delay requirements where the Scaling approach provides the lowest delay. On the other hand, while the PPO-based Scaling and Hybrid schemes present reduced delays, the Migration approach fails to fulfill the emergency requirement with delays greater than $D^s_{\max}$. This is due to the additional delay necessary to migrate the VNF instances between the network nodes. Third, we investigate the reliability of the emergency slice supported by the integrated maritime network. As demonstrated in Fig.7, the Migration-based deployment is the only approach that successfully meets the reliability requirements for both DRL algorithms. This is expected as the VNF instances can be migrated towards the network nodes that are closer to the end-users belonging to the emergency slice.



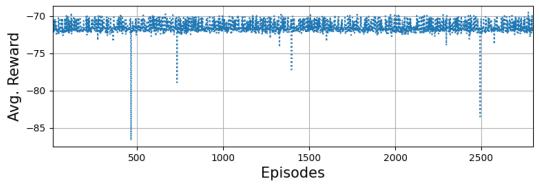
(a) A2C-based Algorithm



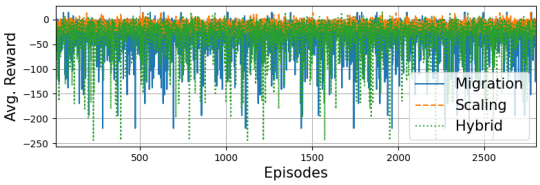
(b) PPO-based Algorithm



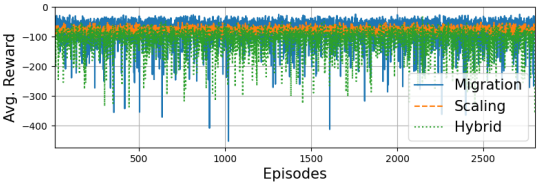
(c) Metaheuristic Benchmark



(d) Static Benchmark



(e) Greedy Benchmark



(f) Random Benchmark

Fig. 3: Average reward over episodes for DRL-based algorithms and benchmarks.

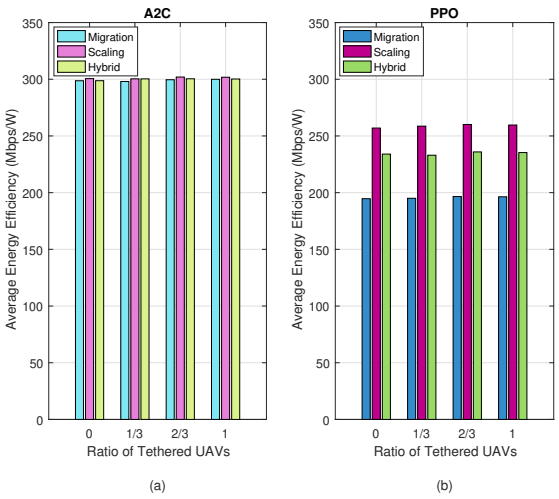


Fig. 4: Energy Efficiency vs. UAVs' Ratio.

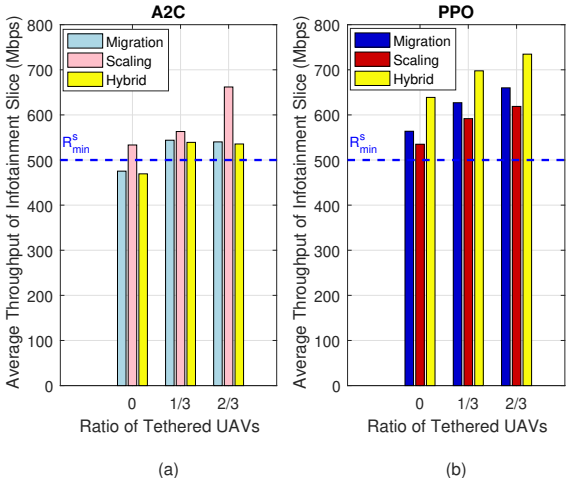


Fig. 5: Infotainment Slice's Throughput vs. UAVs' Ratio.

Moreover, the deployment of supplementary tethered UAVs marginally improves the reliability by around 1% under this scheme since the agent does not receive additional reward once the reliability requirement W^s_{min} is satisfied.

Although all the approaches aim to maximize the energy efficiency, each converges to a distinct policy that accounts for the QoS constraints differently. This is because the scaling and migration actions have different impact on the overall reward and the agents policy updates. In fact, scaling actions produce smooth reward changes that are suitable for both A2C and PPO agents. In contrast, migration actions cause abrupt reward fluctuations that can be captured by the rapid updates of A2C, whereas the long-horizon clipped updates of PPO struggle to adapt. Consequently, these differences lead each agent to prioritize the QoS requirements differently, yielding diverse QoS satisfaction levels. In particular, the A2C-based scaling approach shows superiority in terms of infotainment throughput and emergency delay, while the A2C-based migration scheme achieves improved reliability performance.

5.3 Impact of QoS Requirements' Stringency

In this section, we investigate the impact of the stringency of QoS requirements on the previously studied network performance indicators. First, we consider the infotainment slice throughput by increasing the required R_{min} form

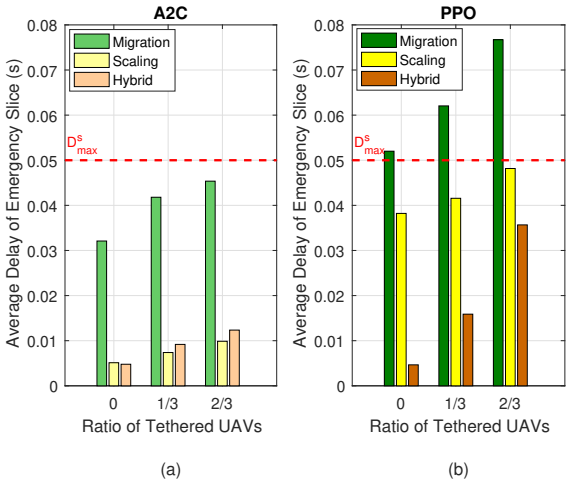


Fig. 6: Emergency Slice's Delay vs. UAVs' Ratio.

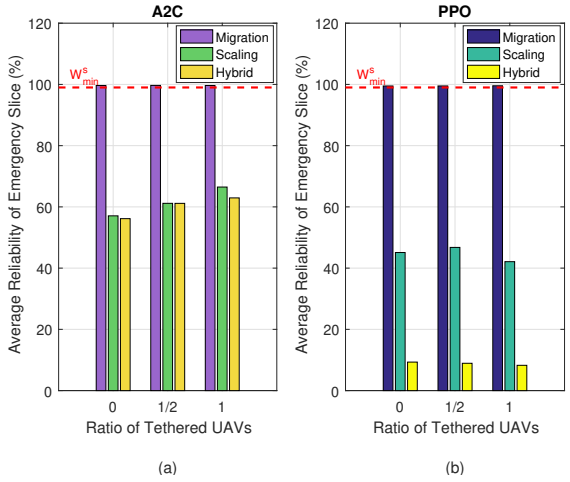


Fig. 7: Emergency Slice's Reliability vs. UAVs' Ratio.

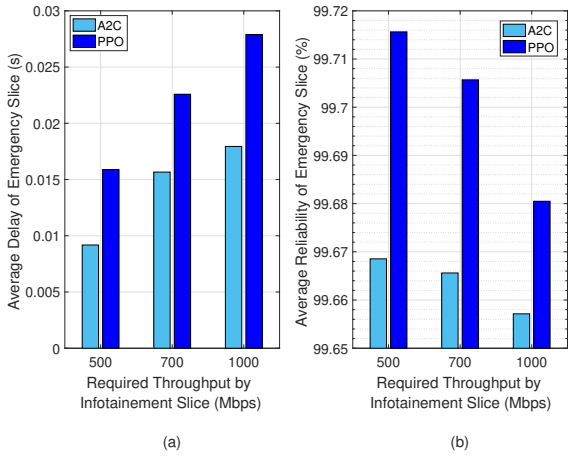


Fig. 8: Impact of Infotainment Slice Requirements' Stringency.

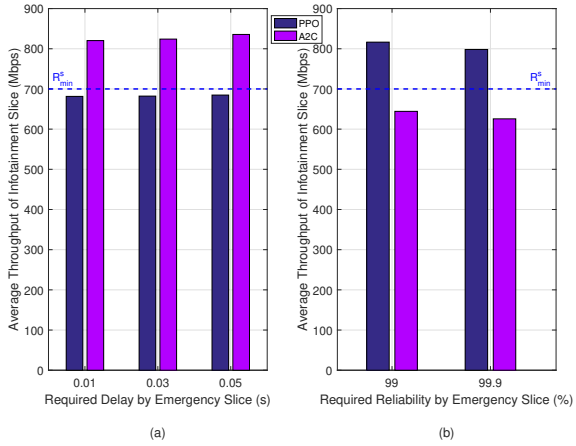


Fig. 9: Impact of Emergency Slice Requirements' Stringency.

$500\ Mbps$ to $700\ Mbps$ and $1000\ Mbps$ respectively. Extensive simulations indicate that this variation has an insignificant impact on energy efficiency. However, the emergency slice indicators are deteriorated as depicted in Fig.8. This is due to the fact that the network resources are predominately allocated to the infotainment slice to satisfy its increasing needs. Specifically, we notice that the average delay of the emergency slice increases and its reliability decreases while meeting the QoS requirements. Notably, the A2C-based algorithm converges to a policy that achieves a lower delay, while the PPO agent guarantees higher reliability.

Moreover, we increase the stringency of the QoS requirements of the emergency slice by decreasing the maximum delay D_{max} form $0.05\ s$ to $0.03\ s$ and $0.01\ s$ and increasing the minimum reliability W_{min} from 0.9 to 0.99. Extensive simulations reveal that this variation has a negligible impact on energy efficiency. Nonetheless, the infotainment throughput is affected as depicted in Fig.9. Specifically, we notice that the average throughput decreases when the QoS requirements get more stringent in terms of delay or reliability. We note also that the A2C-based algorithm guarantees the required throughput R^s_{min} under more stringent delay constraints, while it fails when reliability constraints are more demanding. Contrarily, the PPO-based algorithm achieves the required throughput R^s_{min} under more strict reliability constraints, while it fails when delay constraints become more stringent. These findings suggest that the A2C-based algorithm converges to a policy that maximizes the energy efficiency while prioritizing delay constraints. Meanwhile, the PPO-based algorithm converges to a different policy focusing on reliability.

5.4 Impact of UAVs Trajectory Optimization

In this section, we investigate the effect of non-tethered UAV trajectory optimization on the network performance. Specifically, we notice that the total power consumption can be substantially saved as illustrated in Fig.10. In fact, both algorithms can save around 24% in small-scale maritime network (i.e. 5 ships) and 22% in large-scale maritime network (i.e. 15 ships), where the hybrid approach presents the highest power-saving capabilities as depicted in Fig.10.(a). In addition, we study the impact of increasing the emergency traffic on power savings in Fig.10.(b). We pinpoint that the adoption of the A2C-based algorithm helps

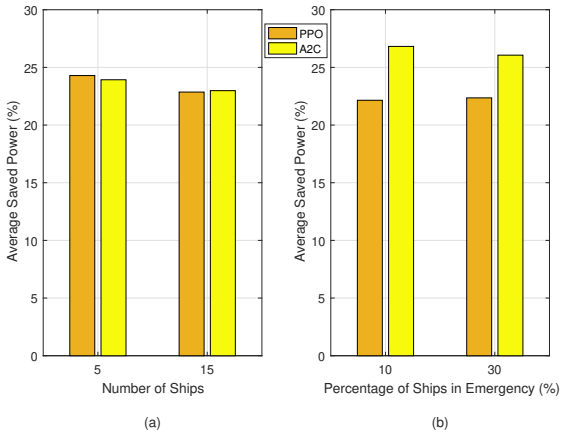


Fig. 10: Impact of UAVs Trajectory Optimization.

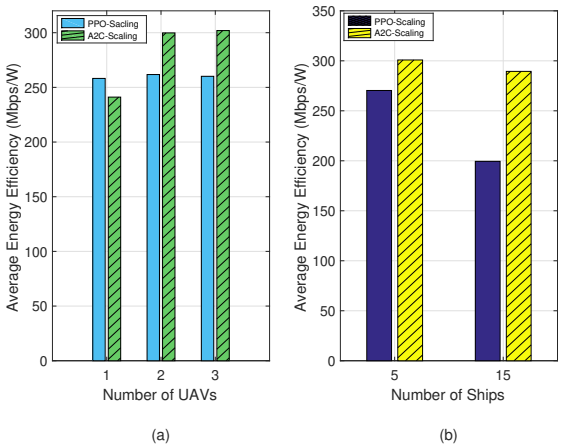


Fig. 11: Impact of Network Scalability.

to increase power saving by 4% compared to the PPO-based algorithm, when more ships are in emergency.

5.5 Impact of Integrated Maritime Network Scalability

In this section, we investigate the scalability performance of the proposed DRL agents. First, we vary the number of aerial nodes and we examine its impact on energy efficiency, as depicted in Fig 11.(a). We observe that the energy efficiency improves in case of the A2C-based scaling approach when the number of UAVs increases. This is due to the short-horizon policy updates of A2C that allows it to explore various actions that increase the throughput, when more UAVs are deployed, with minimized power consumption, leading to energy efficiency gains. In contrast, the PPO-based algorithm shows minor variations which is caused by the clipping mechanism that limits the agent's ability to benefit from the additional UAVs. Second, we increase the number of maritime end-users by varying the number of ships, as shown in Fig 11.(b). We note that the energy efficiency is substantially deteriorated when the PPO-based algorithm is applied, whereas the A2C-based agent shows stable performance with minimal degradation as the number of ships increases. This is expected since the integrated network is required to serve additional maritime users using the same resources. Hence, we can deduce that the A2C agent offers superior network scalability performance compared to the PPO-based approach.

6 Conclusion and Future Directions

In this paper, we proposed an AI-based network slicing framework for O-RAN integrated aerial-terrestrial maritime networks that offers ubiquitous connectivity and satisfies various maritime users requirements. We improved the energy efficiency of the proposed O-RAN maritime network while meeting the requirements of two heterogeneous slices in terms of throughput, reliability and delay. Our findings are twofold. From a DRL perspective, our results highlight the superiority of the A2C-based RAN slicing and VNF deployment. Specifically, the A2C-based algorithm offers better network indicators performance in terms of energy efficiency and it satisfies all the QoS requirements of the infotainment slice and the emergency slice. Therefore, we recommend adopting the A2C-based algorithm for real-world implementation. From a communication perspective, our results show that the migration approach is the most suitable for real-world deployment, as it satisfies all QoS requirements, whether the A2C-based algorithm or PPO-based algorithm is used. This approach comes at the cost of increased delay for the emergency slice and reduced throughput for the infotainment slice, as the reliability of the emergency slice is prioritized. Our future work will focus on the extension of the proposed O-RAN integrated aerial-terrestrial maritime network to a satellite-aerial-terrestrial networks in order to improve connectivity in under-connected maritime areas. We plan to consider satellites mega-constellations capable of offering ubiquitous connectivity in the open-sea to serve efficiently large cargo and cruise ships besides fishing boats and small vessels. In addition, we intend to adopt a multi-agent hybrid DRL approach to handle the continuous and discrete actions of the RAN slicing and VNF deployment problem, and a distributed learning scheme to deal with network scalability.

References

[1] F. M. Insights. Marine communication market outlook (2023 to 2033). [Online]. Available: https://www.futuremarketinsights.com/reports/marine-communication-market

[2] J. Lindner. Must-know cruise ship sinking statistics. [Online]. Available: https://gitnux.org/cruise-ship-sinking-statistics/

[3] F. S. Alqurashi, A. Trichili, N. Saeed, B. S. Ooi, and M.-S. Alouini, "Maritime communications: A survey on enabling technologies, opportunities, and challenges," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3525–3547, 2023.

[4] P. Hadinger, "Inmarsat global xpress the design, implementation, and activation of a global Ka-band network," in *ICSSC*, 2015, p. 4303.

[5] M. Messmer, B. Kiefer, L. A. Varga, and A. Zell, "UAV-assisted maritime search and rescue: A holistic approach," in *IEEE ICUAS*, 2024, pp. 272–280.

[6] S. Ammar, O. Amin, and B. Shihada, "Tethered UAV-based communications for under-connected near-shore maritime areas," in *IEEE BlackSeaCom*, 2024, pp. 42–47.

[7] L. Liu, B. Lin, and Y. Che, "Joint UAV-BS deployment and power allocation for maritime emergency communication system," in *IEEE WCSP*, 2021.

[8] N. Nomikos, A. Giannopoulos, A. Kalafatelis, V. Özduran, P. Trakadas, and G. K. Karagiannidis, "Improving connectivity in 6G maritime communication networks with UAV swarms," *IEEE Access*, vol. 12, pp. 18 739–18 751, 2024.

[9] G. Mildh, E. Myhre, H. Flinck, C. Mannweiler, L. Wan, C. C. Chen, and G. Ericson, "Architecture principles for a cloud-friendly future 6G RAN architecture," O-RAN Next Generation Research Group (nGRG), Tech. Rep. RR-2024-01, 2024.

[10] C.-L. I and S. Katti, "O-RAN: Towards an Open and Smart RAN," *O-RAN Alliance White Paper*, no. WP-2018, October 2018.

[11] B. Agarwal, R. Irmer, D. Lister, and G.-M. Muntean, "Open ran for 6g networks: Architecture, use cases and open issues," *IEEE Communications Surveys & Tutorials*, 2025.

[12] Y. Wu, H.-N. Dai, H. Wang, Z. Xiong, and S. Guo, "A survey of intelligent network slicing management for industrial IoT: integrated approaches for smart transportation, smart energy, and smart factory," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1175–1211, 2022.

[13] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36 009–36 028, 2020.

[14] M. Dubey, A. K. Singh, and R. Mishra, "Ai based resource management for 5g network slicing: History, use cases, and research directions," *Concurrency and Computation: Practice and Experience*, vol. 37, no. 2, p. e8327, 2025.

[15] S. Ammar, C. Pong Lau, and B. Shihada, "An in-depth survey on virtualization technologies in 6G integrated terrestrial and non-terrestrial networks," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 3690–3734, 2024.

[16] M. Gharbaoui, B. Martini, S. Noto, A. L. Ruscelli, P. Pagano, and P. Castoldi, "Experimenting SDN/NFV solutions for flexible maritime transport & logistics (T&L) services," in *IEEE NFV-SDN*, 2023, pp. 27–33.

[17] A. Celik, N. Saeed, B. Shihada, T. Y. Al-Naffouri, and M.-S. Alouini, "A software-defined opto-acoustic network architecture for internet of underwater things," *IEEE Commun. Mag.*, vol. 58, no. 4, pp. 88–94, 2020.

[18] T. Yang, J. Li, H. Feng, N. Cheng, and W. Guan, "A novel transmission scheduling based on deep reinforcement learning in software-defined maritime communication networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, pp. 1155–1166, 12 2019.

[19] O. M. Bushnaq, I. V. Zhilin, G. D. Masi, E. Natalizio, and I. F. Akyildiz, "Automatic network slicing for admission control, routing, and resource allocation in underwater acoustic communication systems," *IEEE Access*, vol. 10, pp. 134 440–134 454, 2022.

[20] C. Zhu, W. Zhang, Y. H. Chiang, N. Ye, L. Du, and J. An, "Software-defined maritime fog computing: Architecture, advantages, and feasibility," *IEEE Network*, vol. 36, pp. 26–33, 2022.

[21] F. Zhang, H. Lu, F. Guo, and Z. Gu, "Traffic prediction based vnf migration with temporal convolutional network," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, 2021.

[22] X. Yu, R. Wang, J. Hao, Q. Wu, C. Yi, P. Wang, and D. Niyato, "Priority-aware deployment of autoscaling service function chains based on deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, 2024.

[23] S. Agarwal, F. Malandrino, C. F. Chiasserini, and S. De, "Vnf placement and resource allocation for the support of vertical services in 5g networks," *IEEE/ACM Transactions on Networking*, vol. 27, pp. 433–446, 2 2019.

[24] T. V. Le, M. V. Nguyen, T. N. Nguyen, H. N. Nguyen, and S. Vu, "Optimizing resource allocation and vnf embedding in ran slicing," *IEEE Transactions on Network and Service Management*, 2023.

[25] H. Shen, Q. Ye, W. Zhuang, W. Shi, G. Bai, and G. Yang, "Drone-small-cell-assisted resource slicing for 5G uplink radio access networks," *IEEE Trans. Veh. Technol.*, vol. 70, pp. 7071–7086, 7 2021.

[26] G. Zhou, L. Zhao, G. Zheng, S. Song, J. Zhang, and L. Hanzo, "Multi-objective optimization of space-air-ground integrated network slicing relying on a pair of central and distributed learning algorithms," *IEEE Internet Things J.*, 2023.

[27] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-aware dynamic sfc mapping and scheduling in SDN/NFV-enabled space-air-ground-integrated networks for internet of vehicles," *IEEE Internet Things J.*, vol. 9, pp. 5824–5838, 4 2022.

[28] M. Pourghasemian, M. R. Abedi, S. S. Hosseini, N. Mokari, M. R. Javan, and E. A. Jorswieck, "AI-based mobility-aware energy efficient resource allocation and trajectory design for NFV enabled aerial networks," *IEEE Trans. Green Commun. Netw.*, vol. 7, pp. 281–297, 3 2023.

[29] X. Feng, M. He, L. Zhuang, Y. Song, and R. Peng, "Service function chain deployment algorithm based on deep reinforcement learning in Space–Air–Ground integrated network," *Future Internet*, vol. 16, 1 2024.

[30] Y. Peng and B. Di, "Joint VNF deployment and resource allocation in integrated terrestrial-aerial access networks enabled by network slicing," in *IEEE EUC*, 2022, pp. 74–80.

[31] J. Wang, H. Zhou, Y. Li, Q. Sun, Y. Wu, S. Jin, T. Q. Quek, and C. Xu, "Wireless channel models for maritime communications," *IEEE Access*, vol. 6, pp. 68 070–68 087, 2018.

[32] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 4, pp. 2804–2821, 2018.

[33] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[34] R.-J. Reifert, H. Dahrouj, A. A. Ahmad, A. Sezgin, T. Y. Al-Naffouri, B. Shihada, and M.-S. Alouini, "Rate-splitting and common message decoding in hybrid cloud/mobile edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 5, pp. 1566–1583, 2023.

[35] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, 2019.

[36] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 1, pp. 732–794, 2015.

[37] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction, 2nd edition*. MIT press Cambridge, 2018.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[40] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[41] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *ICML*, 2016, pp. 1928–1937.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[43] S. V. Albrecht, F. Christianos, and L. Schäfer, *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024. [Online]. Available: https://www.marl-book.com

[44] T. Wei, W. Feng, Y. Chen, C.-X. Wang, N. Ge, and J. Lu, "Hybrid satellite-terrestrial communication networks for the maritime internet of things: Key technologies, opportunities, and challenges," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8910–8934, 2021.

[45] Y. Liu, A. Halev, and X. Liu, "Policy learning with constraints in model-free reinforcement learning: A survey," in *IJCAI*, 2021.

[46] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[47] M. Holzleitner, L. Gruber, J. Arjona-Medina, J. Brandstetter, and S. Hochreiter, "Convergence proof for actor-critic methods applied to ppo and rudder," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLVIII: Special Issue In Memory of Univ. Prof. Dr. Roland Wagner*. Springer, 2021, pp. 105–130.

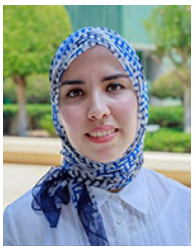
[48] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.

[49] M. Dai, C. Dou, Y. Wu, L. Qian, R. Lu, and T. Q. Quek, "Multi-UAV aided multi-access edge computing in marine communication networks: A joint system-welfare and energy-efficient design," *IEEE Trans. Commun.*, 2024.

[50] D. W. Matolak and R. Sun, "Air–ground channel characterization for unmanned aircraft systems—part I: Methods, measurements, and models for over-water settings," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 26–44, 2016.

[51] J. Xu, M. A. Kishk, and M.-S. Alouini, "Space-air-ground-sea integrated networks: Modeling and coverage analysis," *IEEE Trans. Wireless Commun.*, vol. 22, no. 9, pp. 6298–6313, 2023.

Sahar Ammar received her Diplôme d'ingénieur from Ecole Polytechnique de Tunisie, Tunisia, in 2020 and her M.Sc. degree in electrical and computer engineering in 2022 from King Abdullah University of Science and Technology (KAUST), Saudi Arabia. She is currently pursuing her Ph.D. degree in electrical and computer engineering with the Networking Lab at KAUST. Her research interests include next-generation wireless networks, network virtualization technologies, and optical wireless communications.



Wiem Abderrahim (S'14 - M'18) received her Doctoral Degree in Information and Communication Technologies from the Higher School of Communications of Tunis (Sup'Com), Carthage University, Tunisia in 2017. In 2019, she joined King Abdullah University of Science and Technology (KAUST),Thuwal, Saudi Arabia as postdoctoral fellow within the Computer, Electrical and MathematicalSciences and Engineering (CEMSE) Division. Since 2023, she holds the position of assistant professor at Ecole Nationale d'Ingénieurs de Gabès (ENIG) and she is a research fellow at the MEDIATRON Lab within Sup'Com, Carthage University, Tunisia.

Basem Shihada (M'04-SM'12, IEEE) obtained his Ph.D. in Computer Science from the University of Waterloo, Canada, in 2007. Shortly after completing his studies, Prof. Shihada joined King Abdullah University of Science and Technology as a Founding Faculty member in 2008. His expertise lies in developing cutting-edge wireless systems, where he has made groundbreaking contributions across various domains, including intelligent wireless systems, wireless underwater systems, molecular communication systems, and non-terrestrial systems. Prof. Shihada's notable achievements are the creation and successful demonstration of Aqua-Fi, the world's first underwater Wi-Fi. Sun-Fi, the world first communication via building glass, and communication via breath. Prof. Shihada's work has been recognized with several best paper awards at renowned conferences within his field. His invaluable contributions have also been published in prestigious scientific journals such as Nature Electronics and many IEEE Transactions. In 2023, he become an area editor and received the exemplary editor award from the IEEE Communications Letter journal.