Multi-Layer Networks and Routing Protocols for Aquatic Robotic Swarm Management

Maria MANNONE[1,2,3,4,5], *Peppino FAZIO*[3], *Norbert MARWAN*[1,6,7], *Achille GIACOMETTI*[3,4]

[1]Institute of Physics and Astronomy, University of Potsdam, Potsdam, Germany
[2]ICAR, National Research Council (CNR), Italy
[3]DSMN, Ca' Foscari University of Venice, Mestre-Venezia, Italy
[4]ECLT, Ca' Foscari University of Venice, Venezia, Italy
[5] MIRMI, Technische Universität München, Germany
[6]Institute of Geosciences, University of Potsdam, Potsdam, Germany
[7]Potsdam Institute for Climate Impact Research (PIK), Member of the Leibniz Association, 14473 Potsdam, Germany

{maria.mannone, peppino.fazio, achille}@unive.it, maria.mannone@uni-potsdam.de, marwan@pik-potsdam.de

DOI: 10.15598/aeee.v22i2.5595

Article history: Received Jul 19, 2023; Revised Feb 19, 2024; Accepted Apr 02, 2024; Published Jun 30, 2024.

Abstract. *The paradigm of multi-layer networks can help devise a set of robotic swarms interacting with mobile computing centrals. We present here a distributed hierarchical network model and a related routing protocol (based on static routing and/or AODV protocol for peer nodes) for swarm robotics in aquatic environment, defining also which packets need to be exchanged to guarantee the mission accomplishment. Joining concepts and techniques from different disciplines allows us building a robust system with potential practical applications in scenarios such as environmental care. We discuss our results and further developments of the proposed approach.*

Keywords

Swarm robotics, networks, routing.

1. Introduction

Tout se tient: hints and suggestions from different fields of human knowledge sometimes seem to converge toward a unitary depiction of a complex phenomenon, despite the apparent differences of jargon and computational techniques. This is the case of studies on distributed intelligence [14], which can be instantiated in natural swarms, their robotic imitations [19], as well as network organization. These systems can be investigated in light of biology, physics, computer engineering, telecommunications, complex network theory. Here, we try to develop a compact depiction of an aquatic robotic swarm as a multi-layer network, joining selected tools and concepts from the aforementioned disciplines. A robotic swarm is a robust and scalable system of decentralized set of multiple robots, where each unit interacts with its peers and accomplishes a simple task. The cooperation between robots allows the achievement a complex task, impossible for the single units [4, 1]. Here, we consider multiple swarms, and each swarm is assigned a portion of space for exploration and information collection. Also, the robots of each swarm interact with a computing central put on a larger device, as a human-driven boat, which sends messages to and receive messages from the swarm. The motion of boats takes into account the concept of Point of No Return (PNR), both from a signal and an energy point of view [5]. Boats also interact between them. They autonomously collect information, to be later analyzed offline, helping create a map of the explored space. In this research, we refer in particular to two recent studies. The first one is a graph-based modeling of a set of swarms, interacting with mobile computing centrals, with a routing model and numerical examples obtained from Webots simulations [16]. Because the chosen scenario was a canal in Venice, the robots were simple aquatic models, and the computing centrals were computers put on gondolas. The motion

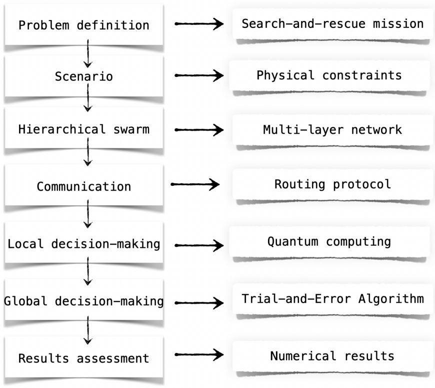

Fig. 1: Block diagram summarizing our overall architecture. Left: steps of problem definition; right: fields involved to face them.

of gondolas is not generating high waves, and thus, it does not damage the underwater structure of the town (the basements of Venice are put at risk by wave motion), and it also does not disturb the stability of the robots of our prototype. In [16], a collaborative networking approach was proposed, for giving instructions to gondolas and their swarm, jointly with an energy and channel quality-aware communication to avoid issues due to charging operations, and losses of robotic units. The second study borrows the quantum-based robotic pairwise interaction and decision-making from [15], extending it with entanglement for connections between gondolas, and using the algebraic formalization of multi-layer network to describe the whole system [17]. Here, we focus on the classic-based decision making, using Python simulations of the robotic swarm. We adopt the architecture of aquatic swarms of robots, where each swarm interacts with a computational unit [16], defining a communication protocol. We also include the formalism of multi-layer networks [17], re-shaping the telecommunication model of [16] according to them. Thus, we also deepen and strengthen some of the hints proposed in [17], focusing on the classic case. In this research, we consider two different frameworks: quantum computing applied to robotics in an ideal scenario with an omniscient platform [15], and a telecommunication protocol developed for simulated aquatic robots [16]. A first attempt of application of the quantum framework to a non-omniscient platform, where simulated robots are really exchanging information, was sketched in [17], jointly with a multi-layer network. Here, the sketch is fully developed, defining a more accurate routing model. The definition of the Trial and Error Algorithm for Missions Accomplishment is also a novelty introduced to this article, with respect to existing former works. The article is organized as follows: in Section 2.1. , we present the formalization of multi-layer networks. In Section 2. , we propose a hierarchical model for robot coordination. In Section 3. , we present our numerical results, and finally, in Section 4. , we summarize our research and discuss ideas for its development.

2. Aquatic Hierarchical Model for Robot Coordination

In this paper we propose a hierarchical model for aquatic robot swarm coordination. There are *slave* robots which deal directly with the particular mission on water surface and *master* devices (as On-Board Units - OBUs) which instruct slaves time by time. Before describing the details of our model, we present a matrix depiction of multi-layer networks, which will be useful to contextualize our study. To summarize the proposed architecture, a block diagram is proposed in Figure 1.

2.1. Multi-layer Networks Theory

We introduce here the hierarchical definition of communicating aquatic *slave* swarms and *masters* via the concept of multi-layer networks. A complex network in general consists of nodes which are connected by links; a *multi-layer network* consists in a multitude of different kinds of nodes and connections. The concept of multi-layer networks is very powerful for the description and modeling of systems with multiple types of interactions, subsystems, and multidimensional structures [3]. Let us summarize some definitions proposed in [17], which are used as here the foundational basis for our telecommunication-swarm robotics approach. We represent the links in an unweighted complex network can be represented via the adjacency matrix $\mathbf{A}$, where $A_{i,j} = 1$, indicates a link between nodes i and j (and 0 for no link). We use Greek letters to indicate different layers. In a multi-layer network, for each individual layer α there is a separate $\mathbf{A}^{\alpha}$ representing the intra-layer links. The links between different layers (inter-layer linking) are indicated by the adjacency matrix $\mathbf{A}^{\alpha,\beta}$, where α, β indicate the corresponding layers. For our special case of *slave* robots connected to *master* OBUs on floating vehicles, we have to use a special multi-layer approach. At time t, the connected *masters* are represented by one layer, with a long-range communication between them. We assume that *master* boats are connected with each other (complete network); then, the corresponding adjacency matrix consists everywhere of values 1 except at the main diagonal (to exclude self-links). For an example of 3 *master* OBUs, we have:

$$\mathbf{A}^{\mathrm{m}} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \tag{1}$$

The robots within a swarm at time t are all connected with each other; thus, the corresponding adjacency matrix would be similar to Eq. (1). The adjacency matrix of *masters* are of the same kind; they form just one layer without links between the different swarms, distinguished through colors in eq. 2.

$$\mathbf{A}^{\mathrm{s}} = \left(\begin{array}{ccc|cc|cccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right). \tag{2}$$

Joining the adjacency matrices $\mathbf{A}^{\mathrm{m}}$ (highlighted in red) and $\mathbf{A}^{\mathrm{s}}$ with the inter-layer connectivity matrices $\mathbf{A}^{\mathrm{m,s}}$, we obtain the global multi-layer network $\mathbf{A} = \begin{pmatrix} \mathbf{A}^{\mathrm{m}} & \mathbf{A}^{\mathrm{s,m}} \\ \mathbf{A}^{\mathrm{m,s}} & \mathbf{A}^{\mathrm{s}} \end{pmatrix}$:

$$\mathbf{A} = \left(\begin{array}{ccc|ccccccccc} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right). \tag{3}$$

The colored diagonal represent intra-layer communications, while the off-diagonal ones (i.e., $\mathbf{A}^{\mathrm{s,m}}$) indicate inter-layer communications. For simplicity of notation, we will explicitly write the time dependence in the next subsection. Further research may address diffusive information flow through the network, that is faster in multi-layer networks than in separate single layers [11], and multiplex networkscan enhance congestions [12].Such a model approach can be based on a Markov chain model [13], a simple standardized model [12],or diffusion model [11].

2.2. Maritime Multi-layer Networks as Telecommunication Architecture and Model

Let us connect the discussed matrix model with concrete telecommunication ad-hoc networks. We focus on top and bottom layers. The top layer is represented by the set of *master* controllers, put on-board of floating vehicles, such as boats and gondolas. They are human-supervised, but acts autonomously, once the mission configuration is set. Each *master* node can be seen as a portable and water-proof device (e.g., an IP65 waterproof/dustproof tablet or a rugged notebook) and a charging station, that is a battery whose dimensions/-capacities allow superficial robots to have a large number of fast charging operations (unlimited if compared with a mission duration). Each floating vehicle hosting a *master* node is completely autonomous, without any kind of energy limitations. We assume, additionally, that storage capability of *master* nodes is not limited and data may arrive directly from the robots in two ways: in real-time, when data are sent from a *slave* to its *master* via communication protocol and a wireless standard (e.g. 5G), or off-line, when data are downloaded after a *slave* node completed its tasks. The bottom layer is composed by the *slave* robots, creating a decentralized swarm, whose possible structure is proposed in Figure 2, showing the *RoboWood*[*] [16]. This is a simple, virtual prototype of aquatic robot, constituted by a wood tablet, with four propellers, GPS sensors emerging from the surface of water, and distance sensors. The robot is also equipped with an underwater camera, to help collect information about the ground. In case of overturning, the robot can be equipped with a set of two underwater cameras, allowing the device to keep taking pictures even if upside-down. At the moment, we are considering surface-only *slaves*, but the same proposed approach is suitable also for underwater communications (equipping *masters* and *slaves* with acoustic transceivers).

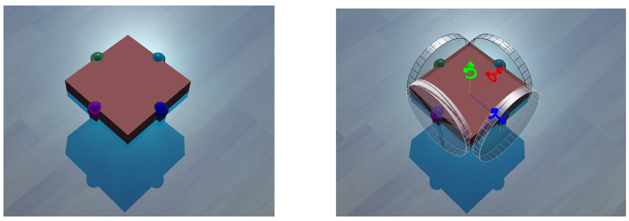

Fig. 2: Visualization in Webots of the aquatic *RoboWood* prototype (left); clicking on it, the software makes the propellers visible (right).

The complete architecture will be described in terms of telecommunication network. We will refer to a *master* node as the boat or its OBU device, without any difference. We can define the set of *master* nodes as $M = \{m_1, m_2, ..., m_n\}$, with $|M| = n$. Each $m_i \in M$, with $i = 1, ..., n$, has a radio coverage radius R_M^i (we can assume to be the same R_M for each *master* node, if all of them are based on the same RF technology). Time by time, the n hosting vehicles are moving, so the connections between each couple m_i, m_j with $i \neq j, i, j = 1, ..., n$ may be available or

[*]https://github.com/medusamedusa/RoboWood

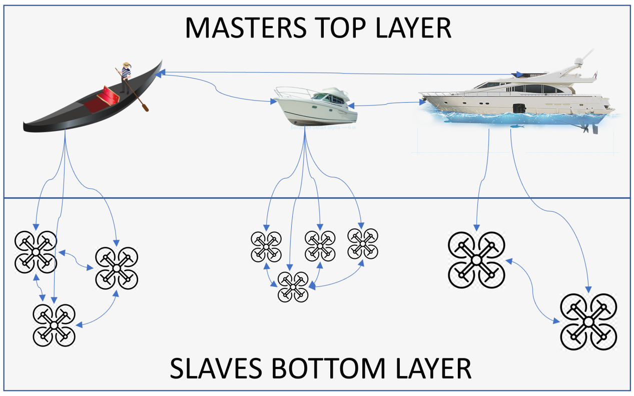

Fig. 3: A graphical representation of a general hierarchical aquatic network, with 3 *master* vehicles and 9 *slave* drones.

not, with a consequent topology change during time. Thus, we can define the connections between all the possible couples of the n *master* nodes in function of time, $E_M(t) = \{e_{1,1}(t), e_{1,2}(t), e_{1,n}(t), e_{i,1}(t), e_{i,2}(t), ..., e_{1,n}(t), ..., e_{n,1}(t), e_{n,2}(t), ..., e_{n,n}(t)\}$, with $|E_M(t)| = n^2$. According to sub-section 2.1. , the generic edge in $E_M(t)$ will be:

$$e_{i,j}(t) = \begin{cases} 1, \text{ if } m_i \text{ is connected to } m_j \text{ at time } t \\ 0, \text{ if } m_i \text{ is not connected to } m_j \text{ at time } t \\ 0, \text{ if } i = j \ (\forall t). \end{cases} \tag{4}$$

The two sets M and $E_M(t)$ define a fully connected graph $G_M(t) =< M, E_M(t) >$ which can be represented by an adjacency matrix A^m, see eq. (1). Concerning the *slave* aquatic robots, we assume that each $m_i \in M$ has its own set of *slaves* $S_i = \{s_{i,1}, s_{i,2}, ..., s_{i,p_i}\}$, $|S_i| = p_i$, constituting the i-th swarm of the system, where p_i is the number of robots associated to *master* m_i. As for the M set, we define a coverage radius $R_S^{i,j}$, assumed to be the same R_S if all *slaves* nodes communicate with the same technology. Also, *slave* nodes have an additional sensing radius—acoustic, optical—for detecting other objects and the target: it can be indicated with $R_{SS}^{i,j}$ or, assuming to be the same, as R_{SS}. The swarm topology may change during a mission, so a connectivity set of edges can be defined as $E_{S_i}(t) = \{e_{i_{1,1}}(t), e_{i_{1,2}}(t), ..., e_{i_{1,p_i}}(t), e_{i_{2,1}}(t), e_{i_{2,2}}(t), ..., e_{i_{2,p_i}}(t), ..., e_{i_{p_i,1}}(t), e_{i_{p_i,2}}(t), ..., e_{i_{p_i,p_i}}(t)\}$, with $|E_{S_i}(t)| = p_i^2$. The values of the elements of $E_{S_i}(t)$ are still 0 and 1, see (4) for the *master* nodes. If no communication is possible between *slaves* $\in S_i$ and *slaves* $\in S_j$, with $i \neq j$, then the two sets S_i and $E_{S_i}(t)$ define a disconnected graph $G_S(t) =< S_i, E_{S_i}(t) >$ which can be represented by an adjacency matrix A^s, as in eq. (2). We need now to connect $G_M(t)$ and $G_S(t)$, defining the interactions between *masters* and *slaves*. Assuming that each boat can communicate with its robots in a full-duplex way, the interactions are undirected. A graph G for the overall system can thus be defined as:

$$G(t) =< \{M \cup S_1 \cup ... \cup S_n\}, \{E_M(t) \cup E_{S_1}(t) \cup ... \\ ...E_{S_n}(t) \cup E_{M,S}(t)\} >=< V, E(t) >, \tag{5}$$

where $E_{M,S}(t)$ represents the set of edges from *master* nodes to *slave* nodes, and its structure depend on how the swarms behave (e.g., they can be clustered and only the *slave* cluster-head may have a connection with its *master*). An example of the representation of $E_{M,S}$ in terms of adjacency matrix is given in eq. (3), where $A^{m,s}$ and $(A^{m,s})^T = A^{s,m}$ represent the interactions between the two hierarchical levels. We notice that A^m, A^s, $A^{m,s}$, $A^{s,m}$, hence A, defined in the previous sub-section, are time-dependent, so we consider them as time-dependent matrices $A^m(t)$, $A^s(t)$, $A^{m,s}(t)$, $A^{s,m}(t)$, $A(t)$, reflecting the topological evolution of the entire system. For the sake of completeness, we illustrate a different representation of the example in Figure 3, to give a clearer view of the definitions given so far. In Figure 4, the top layer (*masters*) is composed by $n = 3$ floating vehicles, while the bottom layer (*slaves*) comprises $p_1 = 3$, $p_2 = 2$, $p_3 = 4$ ($p_1 + p_2 + p_3 = 9$) aquatic robots. The interactions between top and bottom layers are indicated with the dotted circles and the edges inside them, belonging to the $E_{M,S}(t)$ set.

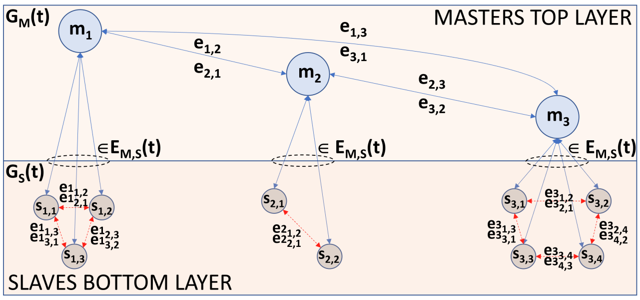

Fig. 4: The hierarchical 2-layers conceptual graph model representation for the system in Figure 3, with $n = 3$, $p_1 = 3$, $p_2 = 2$, $p_3 = 4$.

2.3. Suitable RF technologies and protocols

We assume here that *master* OBUs can transfer data among them by their wireless RF interface and technology, neglecting if it consists of a Wi-Fi, Bluetooth, LoraWAN, [10] or equivalent interface. They are GPS-equipped, so it is possible to know *exactly* which is their position, time by time. We assume the water surface to be flat (no huge waves), so the *slave* robots' mobility occurs in a 2D scenario (depth is not taken into account). Of course, robots are also subject to water surface movements and even sudden waves (we do not consider abrupt waves yet), in addition to movements needed to accomplish a mission, through the communication with *master* nodes. Initially, *slaves* are put on-board by their own *masters* with their fully charge battery. Then, when a *master* $m_i \in M$ reaches a suitable point, the drop-down operation of each *slave* $s_{i,j} \in S_i$ is made, storing the GPS position of node m_i as Return To Home (RTH) position (it can be changed during the mission). The position of $s_{i,j}$, indicated with $P_{i,j}$, is updated time by time (with a frequency f_u) and sent to the related m_i. A mission may provide to reach a known target (or checkpoint), to find a target on an unknown position, or to observe a certain geographical area, while capturing audio, video or other data, depending on the sensors on-board each *slave*. The target is indicated with T, and its position with P_T.

In this paper, we are not considering underwater operations. In that case, the idea is still valid considering dead-reckoning approaches [6, 7]. Once all *masters* deployed their set of slaves, they start to collaborate via m_i. Given the limited energy autonomy of *slave* nodes, the RTH and transmission power management have to be taken into account, to avoid the Point of No Return event (PNR) [16]. Concerning routing, we manage nodes communications through a dynamic protocol. In the considered scenarios, there are no scalability issues ($|V| < 100$), so we propose a simple routing protocol, such as the Ad-hoc On-demand Distance Vector (AODV). See [8] for details on AODV; see [9] for other *Distance-Vector* protocols.

2.4. A possible Trial and Error Algorithm for Missions Accomplishment (TEAM-A)

We illustrate here the core idea of target-search algorithm. *Masters* and *slaves* move inside a defined 2D Geographical Area (GA), whose dimensions are X and Y; it does not matter if we are dealing with Cartesian or GPS/GNSS coordinates. $GA(X,Y)$ sides are divided into q smaller segments, identifying a GRid (GR), composed by q^2 small areas (we use the notation GR_q). For each area $GR_q(x,y) \in GA$, with $x, y = 1, ..., q$, the following relations are valid:

$$\bigcap_{k=1}^{q}\bigcap_{l=1}^{q} GR_q(k,l) = O, \quad \bigcup_{k=1}^{q}\bigcup_{l=1}^{q} GR_q(k,l) = GA. \quad (6)$$

The area of each GR_q is $\frac{X \cdot Y}{q^2}$. Given $GR_q(x,y)$, a time-dependent square matrix $MR(t)$ can be associated to GA and its elements:

$$MR_{(l,q)}(t) = \begin{cases} 1, \text{ if } MR_{(l,q)}(t) \text{ has been visited} \\ 1, \text{ if } MR_{(l,q)}(t) \text{ contains an obstacle} \\ 0, \text{ else.} \end{cases} \quad (7)$$

Here MR is Boolean. However, if the storage of more information is needed, MR can be a generic matrix. E.g., visited condition and obstacle presence can be stored as different statuses; see Figure 5 for a visual summary. MR is initially set to the *null* matrix, then

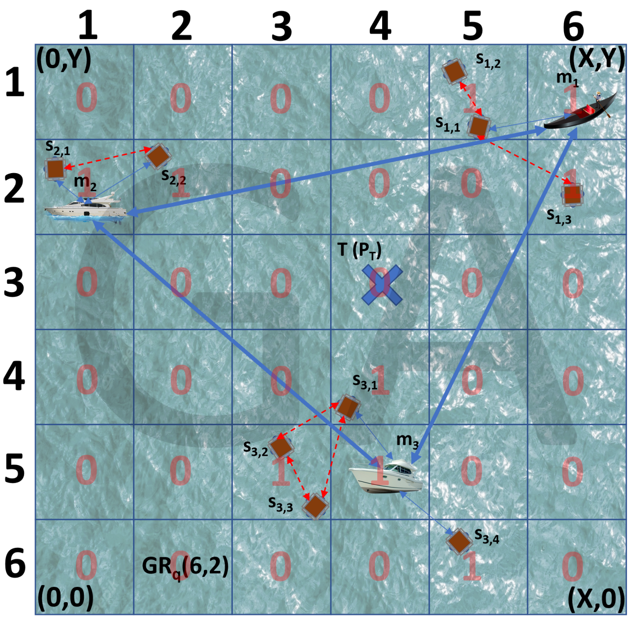

Fig. 5: A possible operative scenario and its representation in terms of $G(t) =< V, E(t) >$, GA, the related grid, and MR, with $q = 6$. The MR elements are indicated with transparent red zeros and ones; red dotted links indicate *slave2slave* activities for AODV protocol route maintenance. Long blue lines represent *master2master* communications, while short and thin blue lines the communications between *slave* gateways (nearest to the own *master*) and *masters*.

it is shared among all the $m_i \in M$, but only *slave* nodes can change indirectly the status of each element of GR: a *slave* is able to communicate with its *master* m_i directly or hop-by-hop (by the AODV protocol). At the application layer, we provide very few packets to be sent among *masters*/*slaves* or between *masters* and *slaves*:

- VISIT_UPDATE($s_{i,j}, m_i, P_{i,j}$): this message is sent back from $s_{i,j}$ to m_i once the chosen destination $P_{i,j}$ is reached by $s_{i,j}$;
- TARGET_FOUND($s_{i,j}, m_i, P_{i,j}$): this message is sent back to m_i once the target has been found in position $P_{i,j}$ by $s_{i,j}$; each *master* is able to map $P_{i,j}$ into the related grid element;
- MATRIX_UPDATE($m_i, m_j, GR_q(k,l)$, Bool): once the master m_i has received a VISIT_UPDATE from its $s_{i,j}$ regarding a reached destination (or target), it sends to the other $n-1$ *masters* (m_j) the request to update their own GR. To reduce the signaling overhead, a centralized matrix can be shared among *masters*,

but those issues are out of the scope of the paper. The packet payload contains also the interested matrix element and the Boolean value (generally *True*, to indicate a visited element);

- MISSION_ACCOMPLISHED($m_i, m_j, GR_q(k,l)$): once the master m_i has received a TARGET_FOUND from one of its $s_{i,j}$, it sends to the other $n-1$ *masters* (m_j) the request to update their own GR and to stop the movements of their *slaves*, indicating the $GR_q(k,l)$ where the target has been found;

- STOP_MISSION($m_i, s_{i,j}$): this message is sent from m_i to its reachable $s_{i,j}$ once the target has been found. *Slave* nodes, then, start the RTH procedure.

Once the *slaves* are dropped into the water, they start to be operative by sending their first VISIT_UPDATE messages, regarding the drop-down position. The mission finishes either when one or more *slave* nodes find the target (in Figure 5 when $GR_q(3,4)$ is set to 1) or if all the matrix elements are set to 1 and the target has not been found.

3. Numerical Results

We present here the results achievable by the proposed TEAM-A scheme. The simulator, taking into account all the defined variables, has been completely implemented in *python*, and several system parameters have been changed (e.g. q, X, Y, n, p_i, f_u, R_M, R_S, R_{SS}). Due to space limitations, we provided to set some of them to typical values: $X = Y = 600\,m$, $f_u = 1\,Hz$, $R_M = 300\,m$ (if we assume an RF standard at least equal to WiFi 4, because no high data-rates are needed [18]), $R_S = 100\,m$ (the standard will be the same, but we set a lower transmission power to optimize energy consumption). A missing element, to be taken into account for realistic results, is represented by a mobility model for *slave* nodes (in our simulations *master* nodes are mostly stopped during each mission). We considered *slaves* mobility parameters through Webots simulations, and some of the obtained profiles are shown in Figure 6. Webots' investigations show that the maximum reachable speed is lower than $0.25m/s$ while, in the average, *slaves* move with a speed of $0.1m/s$. The considered mission consists in finding T, which is placed at the center of a randomly chosen $GR_q(i,j)$ for each run. Figure 7 shows the performance of the AODV in terms of PDR: the number of *mastersn* has been changed from 1 to 6, as well as the number of *slaves* for each $m_i \in M$ ($p_i = 1, ..., n$), assuming that each swarm is composed by the same number of *slaves*. In this way, the total number of *slaves* is changed from

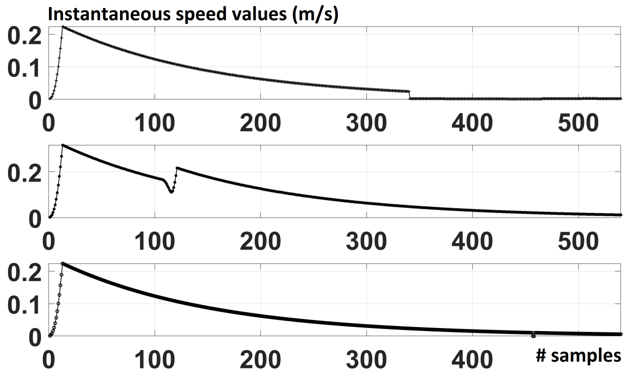

Fig. 6: Typical mobility profiles for *slave* nodes from [16]. We used the parameters from [16] to run our analyses.

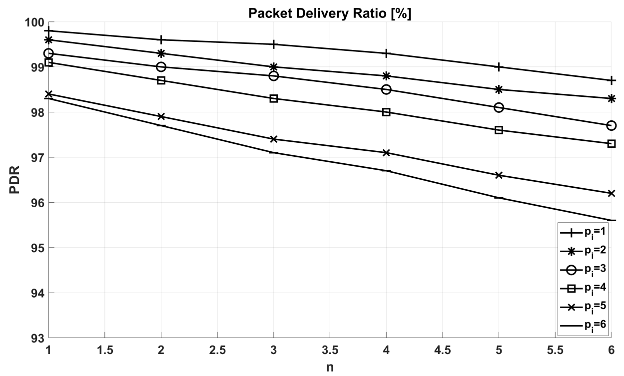

Fig. 7: The average PDR in function of n and p_i.

1 to 36. The PDR (referred to both AODV signaling protocol and TEAM-A packets) does never go under 95% and this can be considered as a very good performance of the overall system. In fact, it is very important to have low packet droppings, in order to guarantee that *slaves* will not loose their connection. At this point, we illustrate what happens in terms of nodes coordination, between *masters* and *slaves*, and the way missions are accomplished. Figure 8 shows the average

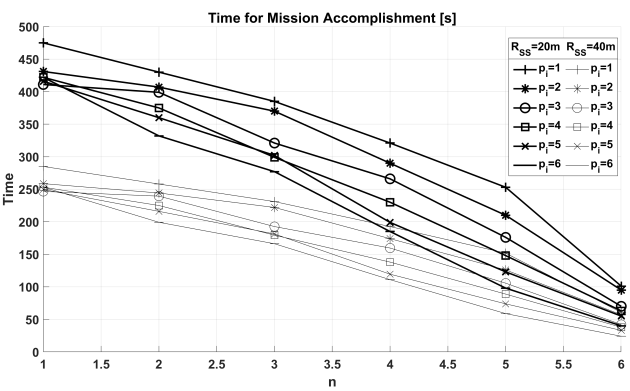

Fig. 8: The average time for mission accomplishment in function of n and p_i.

amount of time needed to find the target and stop the mission (it is referred to the time taken by the first

$s_{i,j}$ to find T). We considered the cases in which the sensing radius R_{SS} is equal to both $20m$ and $40m$; the higher the radius, the higher the energy consumption. We have a decreasing trend for higher p_i, reflecting the validity of the collaborative approach (there will be a higher probability to find T in less time), the lower the trend for higher n, leading to more independent swarms of "slave" nodes.

4. Conclusions and Future Works

In this article, we presented a routing approach for swarm robotics interacting with computational units, modeled as a multi-layer network. We considered a distributed hierarchical network model with a static/-dynamic routing approach (AODV protocol has been simulated for peer nodes) for swarm robotics in aquatic environment, defining also new messages for the application layer. We started from recent and cutting-edge research [16, 17], developing some aspects and strengthening the connections between them. We showed what happens when master and slave nodes interact through a routing protocol without any scalability issue. Simulation results are encouraging, given that the time needed to accomplish a mission can be drastically reduced. Next research will focus on strategies for robotic decision-making, comparing the quantum approach proposed in [15] with the classic approach discussed here. Also, the next theoretical development should algebraically connect the multi-layer network matrix with the single-swarm block matrix discussed in [15]. The quantum-based approach to pairwise robotic decision-making, even if successfully compared against an instance of particle swarm optimization in [15], is not an optimization method, but rather an exploration of the potentialities of quantum computing embedded in a classic-quantum algorithm. While optimization methods can be applied to robotic search-and-rescue algorithms, as well as to the definition of communication protocols [8, 9], this was not the primary goal of our present research. We aimed to join hints from different fields, deepening pioneering insights, and defining a robust routing protocol jointly with a physics-derived multi-layer network. Future research can compare pairwise-interaction methods against heuristic learning, or apply deep learning to our framework. Thus, a discussion on hyperparameters potentially affecting the optimization outcome is here out of scope. Next research may explore the impact of the physical constraints imposed on the robots and on the simulation arena, weighting them against the computational resources required by quantum circuits.

Our work focused on three main pillars: quantum computing, telecommunication protocols, and multi-layer networks. The connection between disciplines, languages, and formalism allows the definition and enhancement of an aquatic robotic system, to help protect what ultimately matters: the beauty of natural and artificial marine architectures.

References

[1] SCHRANZ, M. *et al.* Swarm Robotic Behaviors and Current Applications. *Frontiers in Robotics and AI.* 2020, vol. 7, 2020, doi: https://doi.org/10.3389/frobt.2020.00036.

[2] DE RANGO, F., N. PALMIERI, X.S. YANG *et al.* Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks. *Springer Soft Computing*, vol. 7, no. 4, pp. 4251–4266, 2018, doi: https://doi.org/10.1007/s00500-017-2819-9.

[3] DE DOMENICO, M., C. GRANELL, M. PORTER, A. & ARENAS. The physics of spreading processes in multilayer networks. *Nature Physics.* 2016, vol. 8

[4] PALMIERI, N., X.S. YANG, F. DE RANGO *et al.*. Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption. *Springer Neural Computing and Applications*, vol. 31, pp. 263–286, 2019, doi: https://doi.org/10.1007/s00521-017-2998-4.

[5] TROPEA, M., N. PALMIERI, F. DE RANGO. Modeling the coordination of a multiple robots using nature inspired approaches. In: Proceedings of *Communications in Computer and Information Science (CCIS)*, 2020, pp. 124–133, doi: https://doi.org/10.1007/978-3-030-45016-8$_$13.

[6] BJORGO SAKSVIK, I., A. ALCOCER, and V. HASSANI. A Deep Learning Approach To Dead-Reckoning Navigation For Autonomous Underwater Vehicles With Limited Sensor Payloads. In: *Proceedings of IEEE Oceans*, San Diego – Porto, 2021, doi: https://doi.org/10.23919/OCEANS44145.2021.9706096.

[7] KLEIN, I., and R. DIAMANT. Dead Reckoning for Trajectory Estimation of Underwater Drifters under Water Currents. *Journal of Marine Science and Engineering*, 2020, Vol. 8 Issue 3, doi: https://doi.org/10.3390/jmse8030205.

[8] DEEPIKA, S., N. NISHANTH, A. MUJEEB. An Assessment of Recent Advances in AODV Routing Protocol Path Optimization Algorithms for Mobile Ad hoc Networks. *Fourth International*

Conference on Microelectronics, Signals & Systems, pp. 1–6, 2021, doi: https://doi.org/10.1109/ICMSS53060.2021.9673632.

[9] AL-ZAHRANI, F. A. On Modeling Optimizations and Enhancing Routing Protocols for Wireless Multihop Networks. In: *IEEE Access*, vol. 8, pp. 68953–68973, 2021, doi: https://doi.org/10.1109/ACCESS.2020.2986010.

[10] VAN DEN ABEELE, F., J. HAXHIBEQIRI, I. MOERMAN, and J. HOEBEKE. Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3. *IEEE Internet of Things Journal*, vol. 4, no. 6, 2017, doi: https://doi.org/10.1109/JIOT.2017.2768498.

[11] GÓMEZ, S., A., DÍAZ-GUILERA, J. GÓMEZ-GARSEÑES, C. PÉREZ-VICENTE, Y. MORENO, and A: ARENAS. Diffusion Dynamics on Multiplex Networks. *Physical Review Letters*. **110**, 028701 (2013,1)

[12] SOLÉ-RIBALTA, A., S. GÓMEZ, and A. ARENAS. Congestion Induced by the Structure of Multiplex Networks. *Physical Review Letters*. **116**, 108701 (2016,3).

[13] GRANELL, C., C. GÓMEZ, C. GRANELL, S. GÓMEZ, and A. ARENAS. Competing spreading processes on multiplex networks: Awareness and epidemics. *Physical Review E*. **90**, 012808 (2014,7)

[14] EBERHART, R., Y. SHI, & J. KENNEDY. *Swarm Intelligence*. Burlington, MA, USA: Morgan Kaufman, 2001.

[15] MANNONE, M., V. SEIDITA, A. CHELLA. Modeling and designing a robotic swarm: A quantum computing approach. *Swarm and Evolutionary Computation*. 2023, 79, 101297, doi: https://doi.org/10.1016/j.swevo.2023.101297.

[16] MANNONE, M., V. SEIDITA, A. CHELLA, A. GIACOMETTI, and P. FAZIO, P. Energy and SNR-Aware Robotic Swarm Coordination for Aquatic Cleaning Operations. In: *Proceedings of IEEE Vehicular Technology Conference - VTC*, VTC Spring 2023 Proceedings, IEEE, pp. 1-7, doi: https://doi.org/10.1109/VTC2023-Spring57618.2023.10200327.

[17] MANNONE, M., N. MARWAN, V. SEIDITA, A. CHELLA, A. GIACOMETTI, and P. FAZIO. Entangled Gondolas. Design of Multi-layer Networks of Quantum-Driven Robotic Swarms. In: Villani, M., Cagnoni, S., Serra, R. (eds) Artificial Life and Evolutionary Computation. WIVACE 2023. Springer, Cham, 2024, doi: https://doi.org/10.1007/978-3-031-57430-6_14.

[18] NISHANDHI, S. *et al.*, Survey on various IOT standard protocols and its applications. In: *Proceedings of International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2020, pp. 1-5, doi: https://doi.org/10.1109/ICSCAN49426.2020.9262444.

[19] ŞAHIN, E. Swarm robotics: From sources of inspiration to domains of application. In: *International Workshop on Swarm Robotics* (Lecture Notes in Computer Science, Vol. 3342), E. Şahin and W. M. Spears (Eds.). Berlin/Heidelberg: Springer, 2004.

About Authors

Maria MANNONE achieved her MSc in Theoretical Physics at the University of Palermo (Italy) in 2012, and her PhD in Composition at the University of Minnesota (USA) in 2017. Her research concerns swarm robotics, quantum computing, complex networks, and mathematics and music. She is a researcher at CNR and PhD student in Physics at the University of Potsdam.

Peppino FAZIO received the Ph.D. degree in Electronics and Communications Engineering, University of Calabria (UNICAL - Italy) in January 2008 and completed his habilitation as Associate Professor in April 2017. After 12 years of Research Fellowship/Assistant Professorship in UNICAL, he is Associate Professor at the Ca' Foscari University of Venice, in the Department of Molecular Sciences and Nanosystems (DSMN), from January 2021. From May 2008 to September 2008, he was a Visiting Research Fellow with the UPV of Valencia (Spain), GRC Research Group. He collaborates also, from 2017, with the VSB – Technical University of Ostrava (Czech Republic), as senior research fellow. He is co-author of more than 135 papers (55 in International Journals) and one book, all indexed in Scopus and/or WoS. He has been a member of various start-up companies and recipient of national and international awards (e.g. the Intel Business Challenge Award in 2013). His research interests include mobile communication networks, IP QoS architectures and interworking, wireless and wired networks, mobility modelling for WLAN environments, mobility analysis for prediction purposes, routing, vehicular networking, MANET, VANET and Quantum Key Distribution Networks.

Norbert MARWAN received his PhD in Physics from the University of Potsdam (Germany) in 2003. His research interests include complex systems, nonlinear data analysis, time series analysis, and climatology. He is currently Professor at the University of Potsdam.

Achille GIACOMETTI gratuated in Physics (MSc) at the University of Padua (Italy) in 1986, and achieved his PhD in Physics at the Purdue University (USA) in 1993. His research interest includes physics of complex systems. He is currently Director of the European Centre for Living Technology (ECLT) and Professor at Ca' Foscari University of Venice, Italy.