

# HIRO-NET: Heterogeneous Intelligent Robotic Network for Internet Sharing in Disaster Scenarios

Ludovico Ferranti<sup>\*†</sup>, Salvatore D'Oro<sup>\*</sup>, Leonardo Bonati<sup>\*</sup>,  
Francesca Cuomo<sup>†</sup>, Tommaso Melodia<sup>\*</sup>

<sup>\*</sup>Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

<sup>†</sup>Sapienza University of Rome, Italy

Email: {ferranti.l, s.doro, bonati.l, melodia}@northeastern.edu,  
{ludovico.ferranti, francesca.cuomo}@uniroma1.it



**Abstract**—This article describes HIRO-NET, an Heterogeneous Intelligent Robotic Network. HIRO-NET is an emergency infrastructure-less network that aims to address the problem of providing connectivity in the immediate aftermath of a natural disaster, where no cellular or wide area network is operational and no Internet access is available. HIRO-NET establishes a two-tier wireless mesh network where the Lower Tier connects nearby survivors in a self-organized mesh via Bluetooth Low Energy (BLE) and the Upper Tier creates long-range VHF links between autonomous robots exploring the disaster stricken area. HIRO-NET's main goal is to enable users in the disaster area to exchange text messages to share critical information and request help from first responders. The mesh network discovery problem is analyzed and a network protocol specifically designed to facilitate the exploration process is presented. We show how HIRO-NET robots successfully discover, bridge and interconnect local mesh networks. Results show that the Lower Tier always reaches network convergence and the Upper Tier can virtually extend HIRO-NET functionalities to the range of a small metropolitan area. In the event of an Internet connection still being available to some user, HIRO-NET is able to opportunistically share and provide access to low data-rate services (e.g., Twitter, Gmail) to the whole network. Results suggest that a temporary emergency network to cover a metropolitan area can be created in tens of minutes.

**Index Terms**—Emergency Network, Robotic Autonomous Networks, Wireless Mesh Networking, Disaster Scenarios

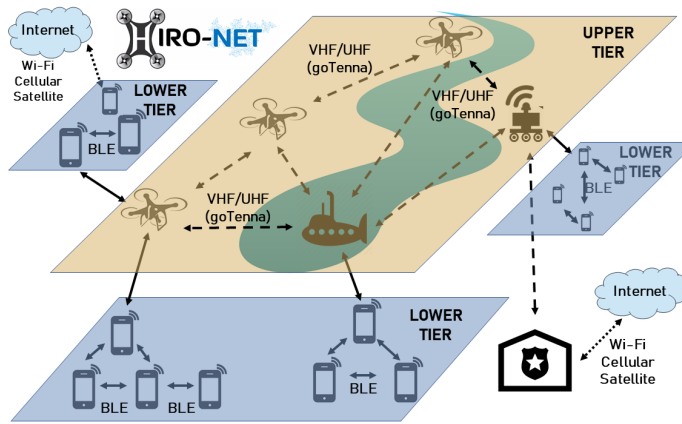
## 1 INTRODUCTION

Recent extreme natural events have shown that cellular infrastructures cannot reliably withstand natural disasters. In particular, up to 80% of cellular infrastructures were knocked down in Florida bay counties after Hurricane Michael [1], 90% after Hurricane Irma [2] and almost the entirety of the Puerto Rico communications went down after Hurricane Maria [3]. Such nefarious circumstances raise the need for authorities and researchers to provide better solutions to tackle the problem of locating and ensuring survivors safety, as well as keeping the communication infrastructures operational. Real-time information on status of survivors needs to be collected and provided to first responders

and authorities efficiently. Differently from legacy mesh networks (e.g., Wireless Sensor Networks, Mobile Ad-hoc Networks) Emergency Networks (EN) need to ensure communication among survivors and first responders with rapid yet energy-efficient deployment that leverages the pre-existing deployment of smart devices (including smartphones and tablets) to the largest extent possible. In a disaster situation, social apps are the first interface that survivors use to get real-time news and coordinate with authorities. As a result, wireless technologies play a critical role in emergency situations. HIRO-NET is a rapidly-deployable EN designed to be deployed in disaster-stricken regions where most individuals have lost access to pre-existing communication infrastructure, including Internet and cellular services. HIRO-NET stands for Heterogeneous Intelligent Robotic Network, as the network includes and orchestrates several heterogeneous communication technologies, acquires information on the explored environment through different autonomous robots, and reacts dynamically to changes, thus embedding intelligence in the rescuing operations. HIRO-NET goal is to establish a self-organizing mesh network with all users in wireless proximity using Bluetooth Low Energy (BLE) wireless standards. In the event of some wide area network still being operational (e.g., surviving cellular networks or satellite Internet links established at rescue headquarters) any HIRO-NET node is able to share its Internet connectivity with other participants in the emergency network. A set of HIRO-NET air, water and ground autonomous vehicles, wirelessly connected through VHF wireless technology, are then deployed to identify, locate and bridge disconnected portions of the peer-to-peer mesh network.

This work makes the following contributions:

- An emergency network solution leveraging the pre-existing pervasive deployment of smart devices (including smart-phones and tablets) that enables seamless and cross-platform (iOS and Android) mesh networking capabilities.



**Figure 1:** Physical and logical architecture of HIRO-NET: the blue areas are the Lower Tier (BLE), while the orange area represents the Upper Tier (VHF).

- An emergency network solution that provides users without Internet or cellular connectivity with the ability to establish on-demand communications. Existing apps (e.g., Firechat, Bridgefy) provide infrastructureless networking, but do not support seamless sharing of Internet connectivity with the entire mesh.
- A two-tier, decentralized, adaptive mesh networking framework to provide survivors with an emergency-tailored notification system aimed at improving safety and reliability of the emergency network. The Upper Tier of the network is created by robotic air/ground/underwater nodes with long-range VHF radio mesh connectivity. HIRO-NET provides optimized deployment of these mesh-connected robotic nodes as network bridges between local, spontaneously created mesh networks, further extending the communication range and coverage.
- Novel distributed coordination mechanism for robots engaged in map exploration and deployment of network services. The self-organization capabilities of a heterogeneous multi-robot system is, to the best of our knowledge, an unexplored branch of research in the field of EN.
- A formal mathematical model of the discovery and bridging problem, as well as algorithms to find optimal solution.

The remainder of this paper is organized as follows. Section 2 presents the main related works. Section 3 introduces the system architecture of HIRO-NET and presents the design and implementation of the proposed HIRO-NET prototype. In particular, Section 4 accurately describes the Lower Tier functionalities of HIRO-NET and Section 5 explains how such functionalities are extended through the Upper Tier. In Section 6 the optimization problems related to HIRO-NET deployment phases are formulated. In Section 7 the network deployment and its experimental evaluation are presented, while numerical results are presented in Section 8. We conclude the paper by discussing the current limitations of HIRO-NET and how we plan to address them in Section 9.

## 2 RELATED WORK

In the past few years the research community has been leveraging UAVs to create airborne wireless networks. Wang et al. discuss the challenges of multi-UAV ad hoc networks and study drone cooperation to enhance the network performance and coverage area [4]. Yanmaz et al., instead, propose and experimentally evaluate the architecture and principles for UAV networks interacting with fixed ground stations [5].

The interactions between UAVs and ground nodes are considered by Kung et al. in [6], and by Barritt et al. in [7]. The former characterizes the wireless channel in a IEEE 802.11 mesh network, while the latter analyzes the advantages of temporospatial software-defined networking approaches applied to the Internet backhaul. A distributed control framework for UAV swarm networks supporting automatic optimization operations is, instead proposed by Bertizzolo et al. in [8].

The use of drones in 5G networks has been investigated by D’Alterio et al. that propose a UAV-enabled cellular base station [9], by Ferranti et al. that prototyped a UAV-based platform to provide cellular coverage in case of outage of the traditional infrastructure [10], and by Bertizzolo et al. that design a control mechanism to mitigate cellular interference caused by 5G-enabled UAVs to ground users [11]. Furthermore, mmWave-enabled drone networks have been considered by Polese et al., which propose an empirical propagation loss model for UAV-to-UAV communications [12], and by Bertizzolo et al., which demonstrate the advantages of location-aided mmWave backhaul link management for UAV cells [13].

The design challenges of multi-tier drone-enabled cellular networks have been investigated by Sekander et al. [14], while Bor-Yaliniz and Yanikomeroglu propose the use of multi-tier drone-cell networks to complement terrestrial heterogeneous networks [15]. Camp et al., instead, characterize and model the propagation environment in a two-tier urban mesh network and propose measurement-driven deployment strategies [16]. Node placement algorithms in wireless mesh networks are, then, devised by Franklin and Murthy in [17].

Communication schemes for multi-tier UAV-assisted networks for disaster recovery scenarios are considered by Mezghani et al. in [18]. Similar to our work, the authors leverage the ubiquity of smartphones to assist search and rescue operations after a natural disaster occurs. However, the focus of this work is more on the theoretical evaluation of different communication technologies, and on data collection rather than on actual demonstration and experimental evaluation through a real testbed. Moreover, it does not provide key features such as Internet sharing to assist search and rescue operations.

Kuperman et al. compare routing algorithms for airborne ad hoc networks [19], while Rovira-Sugranes and Razi introduce optimal routing algorithms for dynamic UAV networks [20]. Furthermore, Zhang et al. propose an optimal layered architecture for swarms of drones and design a low-latency routing algorithm for Internet of Things (IoT)-enabled UAV scenarios [21].

Efficient UAV area coverage has been theoretically investigated by Mozaffari et al. in [22], Galkin et al. in [23], and

Dai et al. in [24]. While some works adjust parameters such as the drone altitude in the case of UAV-enabled wireless base stations and cellular access points [22], [23], the authors of [24] propose a quality-aware energy-efficient drone coverage and path planning scheme to monitor geometrically complex terrains.

Heterogeneous robotic platforms for sensing applications are leveraged by Sudarshan et al. for seismic sensing [25], and by Cruz et al. in the case of deep water seafloor mapping [26].

The use of UAVs for disaster management and recovery has been widely investigated in the past few years. For instance, Hildmann and Kovacs discuss the benefits and challenges of UAV deployment in disaster areas, with a focus on safety and security applications [27], while Erdelj et al. survey the advances and use of UAVs to assist first responders and perform disaster management and rescue operations [28]. Câmara envisions an architecture that leverages sensors-equipped drones to map the disaster terrain, and searches and provides a temporary communication infrastructure to survivors [29]. Gregory et al. present a performance- and resiliency-oriented framework for disaster-site exploration, along with policies to assign areas of interest to UAVs and ground robots [30]. However, the proposed framework does not focus on survival search and rescue operations. Autonomous UAV navigation issues are tackled by Xu et al. leveraging vision and machine learning techniques [31]. Rottondi et al. optimize multitask mission planning of a fleet of UAVs in disaster scenarios [32], while Sánchez-García et al. propose a delay tolerant network-based algorithm for UAVs to locate survivors by exploring the disaster area [33].

Finally, Coletta et al. demonstrate DANGER, a distributed framework to create drone mesh networks and rescue survivors in disaster scenarios [34]. Differently from our work, DANGER leverages WiFi connectivity to connect survivors, which is significantly more energy-consuming than BLE used by HIRO-NET.

In this article, we extend our previous work [35] by:

- Providing a formal definition and proof of the complexity of the discovery problem;
- Expanding the algorithmic solutions provided by HIRO-NET;
- Providing results about battery lifetime and consumption of the HIRO-NET Overlay Network Protocol;
- Studying the scalability and optimality of HIRO-NET aerial nodes' battery management;
- Extend the experimental results to multi-drone scenarios;
- Adding numerous implementation details that were left out in the conference paper.

### 3 SYSTEM MODEL

HIRO-NET's deployment occurs in three distinct phases: (i) Local mesh network formation; (ii) discovery of isolated local mesh networks using autonomous robots, and (iii) robot placement optimization to bridge disconnected mesh networks and Upper Tier formation. Immediately after a disaster, users activate the HIRO-NET application on their

smart-phone/tablet (iOS or Android). Upon launch, HIRO-NET automatically begins the first phase of establishing a self-organizing local mesh network via BLE, connecting nearby survivors in a self-organized mesh. The second phase consists of deploying a group of robotic vehicles (air drones) to identify existing disconnected mesh networks. Each drone carries a polyglot multi-radio node, endowed with at least two radio interfaces: (i) Bluetooth, and (ii) long-range VHF radio based on the goTenna mesh communications suite [36]. HIRO-NET air drones are envisioned to take off from a rescue headquarter (e.g., police stations, fire department) and fly within a confined region to search for any newly formed local mesh networks. HIRO-NET air drones are first directed to specific Points Of Interest (PoI), which are pre-determined areas such as schools, hospitals and stadiums. PoIs are assigned to each drone by using clustering algorithms to create an offline trajectory plan. Graph theory optimization is applied to compute the optimal trajectory of drones considering battery constraints. Each drone runs an online beaconing routine that discovers HIRO-NET networks and dynamically updates its trajectory.

In the third phase, HIRO-NET deploys a robotic mesh network to bridge and connect local disconnected meshes and adds Upper Tier functionalities to all the meshes. At this stage, HIRO-NET fully establishes the two-tier mesh network. The Lower-Tier consists of all local mesh networks generated via BLE while the Upper-Tier uses goTenna radios to interconnect drones and other robotic vehicles operating on VHF frequencies over a range of 0.5-2 miles. The overlay network architecture allows:

- Sharing dynamic control and coordination information among robots and first responders;
- Send text messages generated in the Lower Tier over larger areas;
- Providing low data-rate Internet services such as emails and Twitter;
- Sharing such Internet connectivity inside and outside the mesh network where access to the Internet is still available.

As soon as drones have discovered at least two local disconnected mesh networks, HIRO-NET ground or water robots are automatically dispatched. The specific robot to be deployed depends on the terrain of the path toward the optimal location. For instance, underwater/surface robot would be deployed in a scenario similar to the aftermath of Hurricane Harvey, where large areas of Houston were flooded. Since smart-phones and tablets are battery-operated, power consumption must be minimized. At the end of the three phases, HIRO-NET provides low data rate mesh connectivity to all users of the HIRO-NET app, provided that enough robotic vehicles are dispatched to ensure coverage or bridge disconnected meshes.

#### 3.1 Two-tier Network Architecture

The key innovative feature of HIRO-NET is the two-tier nature of the deployed network. The Lower Tier consists of local mesh networks generated via short-range BLE communications. The Upper Tier uses goTenna, an off-the-shelf VHF radio, to interconnect drones and other robots to connect

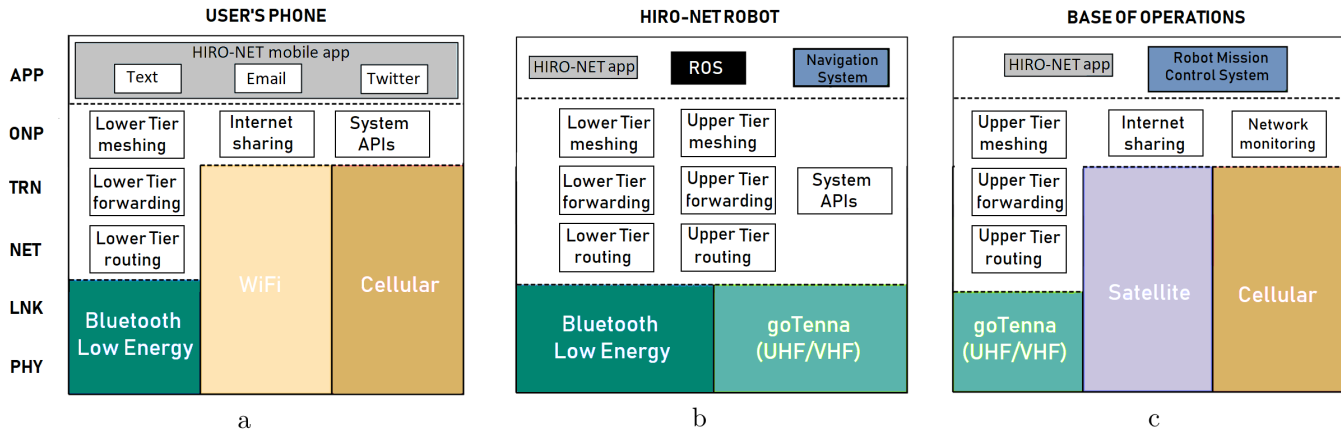


Figure 2: Protocol stack of HIRO-NET nodes.

isolated meshes to the global HIRO-NET network. HIRO-NET is an overlay network that carries dynamic control and coordination information among robots and first responders (e.g., discovered mesh networks, previously visited PolS). While the Lower Tier is used for local coordination among survivors, the Upper Tier constitutes the backbone of HIRO-NET and the interaction between the HIRO-NET network functionalities is mediated by an Overlay Network Protocol (ONP). In Figure 1, a visual representation of HIRO-NET physical and logical architecture is provided. The HIRO-NET robotic overlay network is based on air/ground/water autonomous vehicles, capable of both short and long-range communications, deployed to search and connect survivors. Search and connect operations rely on graph optimization and clustering algorithms that use geographical and topological features of the disaster area to improve the probability of discovering new disconnected meshes and maximize the size of the covered area. Three different type of nodes participate to the network: Users, Robots and Base of Operations (BO). Users only belong to the Lower-Tier, while Robots and BO have both Lower Tier and Upper Tier capabilities.

### 3.2 Protocol Stack

Each type of node participating to HIRO-NET is equipped with a specific protocol stack. While HIRO-NET Lower Tier network is created via BLE, users' phones need to opportunistically capture wide-area network signals (e.g., WiFi and cellular network) in order to be able share it with the whole network. In HIRO-NET the ONP is used to orchestrate efficiently the coexistence of those technologies and ensure that each and every function is provided to the whole system. Users are univocally identified in the system through their phone's ID (e.g., phone number). The ONP runs a meshing daemon that discovers the mesh topology and creates routes according to specific forwarding policies. Figure 2 shows the protocol stack for HIRO-NET. HIRO-NET drones run an exploration routine to find as many Lower-Tier mesh networks as possible. HIRO-NET drones physically carry an Android Things compatible board and a goTenna radio in order to be compatible with Lower Tier and Upper Tier respectively. The HIRO-NET drone app is designed to interact with ROS (Robot Operating System)

[37] that runs on drones, ground robots and water vehicles. ROS forwards information to Flight Controller Unit in order to modify and adjust the drone exploration trajectory. Figure 2 depicts the different configurations of HIRO-NET protocol stack. The BO monitors the state of the network and dispatches robots to the designated bridging areas and represents a base for the coordination of the rescue efforts. The BO also has a higher chance to still be able to capture Internet access through surviving Cellular Networks or Satellite Internet.

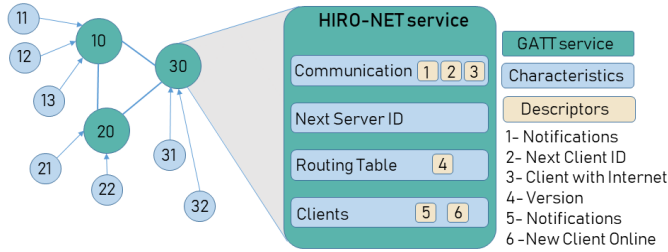
### 4 LOWER TIER MESHING

Each and every local mesh network is established by users' smart-devices in proximity. When the HIRO-NET app is launched, the smart-device starts advertising itself to other users through BLE. Both Android and iOS typically scans for other Bluetooth devices every 0.1 to 1 second with a range that goes from few meters Non-line-of-sight to 100 meters in Line-of-sight. The ONP collects four key information about every node that are of interest for the system, including (i) Phone ID (i.e., phone number); (ii) next hop; (iii) GPS coordinates, and (iv) Internet connection availability.

The users' phone ID is used to univocally identify a node in the system. GPS coordinates are collected as localization needs to be available to first responders and is a key component of HIRO-NET functionalities. Internet connection availability is needed to know which users are still connected to the Internet and could provide it to the whole network.

HIRO-NET Lower Tier is implemented as a Bluetooth Low Energy GATT (Generic Attribute) service [38]. BLE is based on a Client-Server paradigm, where a Client can send and receive information through reads and writes of Server characteristics and descriptors, as well as receiving updates from Server in the form of notifications resulting from specific subscription. A Server can connect multiple Clients in a piconet and run a Client instance to connect to other Servers and forming a scatternet. As BLE GATT Servers were historically developed to run on BLE peripherals (e.g., IoT sensors, wearables), the core of HIRO-NET service is implemented on the Server side, as Clients are mostly consumers of data produced by Servers. The implementation of HIRO-NET service is depicted in Figure 3. Upon entering the

mesh, each user can decide to become a Client or a Server. A Server advertises the HIRO-NET service using a custom UUID (Universally Unique Identifier), while a Client scans for such ID. After a timeout if a Client cannot find a Server, it elects itself as a Server and starts advertising. Moreover, while advertising, Servers also scan for other Servers in order to be able to connect to them and upgrade its piconet to a scatternet. In the current implementation of HIRO-NET, Servers have an ID  $i \in \{1, 16\}$ , while Clients have an ID  $ij$  where  $i$  is the number of the associated Server and  $j$  is a number  $\in \{1, 7\}$  [39]. This configuration has been set to allow 1 byte local mesh addressing and to limit the number of users of each local mesh to 116.



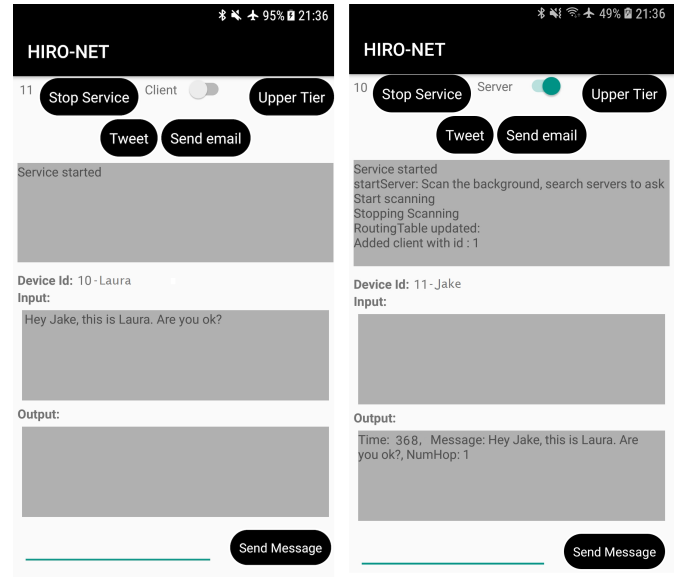
**Figure 3:** HIRO-NET BLE GATT Server and topology.

The HIRO-NET service has the following characteristics ( $\mathcal{C}$ ) and descriptors ( $\mathcal{D}$ ):

- **Communication:**  $\mathcal{C}$  used for messages exchange and general Server-level communication.
  - *Notification:*  $\mathcal{D}$  used by Servers to notify a Client that a new message is available. Clients subscribe to this.
  - *Next Id:*  $\mathcal{D}$  containing the value of the next ID to be assigned to a new Client
  - *Client with Internet:*  $\mathcal{D}$  used to store the ID of a Client that has Internet access
- **Next Server ID:**  $\mathcal{C}$  containing a globally synchronized value that represents the next available Server ID in the local mesh;
- **Routing Table:**  $\mathcal{C}$  used to send and receive the routing table, used only at Server-side to globally synchronize the routing table.
  - *Version:*  $\mathcal{D}$  used by Servers to keep the routing table updated. Every new update, increase the version value. Servers subscribe to this.
- **New Client Online:**  $\mathcal{C}$  used by Servers to maintain a list of active Clients.
  - *Notification:*  $\mathcal{D}$  used by Servers to communicate a list of active Clients.
  - *New Client Online:*  $\mathcal{D}$  used by Servers to notify a new Client is online. Clients and Servers subscribe to this.

Upon entering the mesh, Server scans for other Servers and read the Next Server ID in order to avoid addressing conflicts. When a new Client gets connected to the local mesh, the Server assigns it a new Client ID by reading the Next ID descriptor in an incremental way (e.g., the fourth Client of the second Server has ID 24) and notify the whole

network through the descriptor New Client Online. Once a new node (Client or Server) enters the network, it receives the current version of the routing table. From that moment on, users are able to communicate with others by using the Communication characteristic. Clients can directly write on the Communication characteristic, while Servers change their local value and then send a notification to the appropriate client. A screenshot of the HIRO-NET Android app is shown in Figure 4. To match the standard Bluetooth Low



**Figure 4:** A screenshot of the HIRO-NET Android app. The Upper Tier button is used to request users to the nearest HIRO-NET robot.

Energy MTU (Maximum Transmission Unit) all HIRO-NET packets have the size of 18 bytes. The standard Lower Tier packet, uses 1 byte for the sender ID (4 bits for Server ID, 3 bits for Client ID and 1 bit as a flag to notify the last packet), 1 byte for receiver ID (4 bits for Server ID, 3 bits for Client ID and 1 bit as a flag to notify whether or not that Client has Internet access) and 16 byte for payload. Payload usually consists in the hashed message/tweet/email. By reading the Sender ID and Destination ID of every packet, each Server can orchestrate reliable reads and write in the Communication characteristic. Routing table packets are different from standard packets: their transmission is triggered only when a new node enters the network and depend on the role of each node. Server routing table packets have the second byte empty (only 1 bit of the second byte is used as a flag) and are sent only among Servers. When a new Server enters the network, it receives the whole routing table (packet flag set to 1), while if a Client enters the network, only a local update is sent (packet flag set to 0). This mechanism introduces a delay when a new Server enters the network, but is necessary to ensure network convergence. Routing table packets use the Routing Table and New Client Online characteristic, in the same way the Communication characteristic is used to exchange messages. To keep a list of active Clients/Servers, Servers send a special Client Routing Table packet is sent. In this packet, the 16 bytes payload represent the online Servers of the local mesh network and the bits of each byte

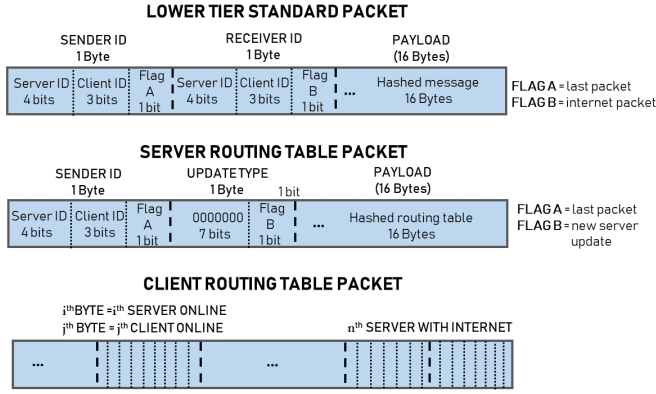


Figure 5: Lower Tier packet format.

represents the online Clients of the network (e.g., if the  $j^{\text{th}}$  bit of the  $i^{\text{th}}$  byte is set to one, the Client  $j$  of Server  $i$  is online). The last two Bytes map which Server has Internet connection ( $i^{\text{th}}$  bit set to 1 if  $i^{\text{th}}$  Server has Internet). This packet modifies the New Client Online characteristic.

The complete list of Lower Tier packets format is shown in Figure 5.

## 5 UPPER TIER MESHING

The HIRO-NET Upper Tier consists of the robotic vehicles (UAVs, mobile ground robots, water surface robots) controlled by first responders and equipped with radios operating at both BLE and VHF frequencies. The ONP controls the interactions between Lower Tier and Upper Tier by ensuring seamless communications between the two tiers.

Since goTenna supports packets with a maximum payload size of 236 bytes, and allows up to 5 transmissions per end-user per minute, the Upper Tier meshing algorithm implemented by HIRO-NET relies upon a reactive protocol that limits the number of transmissions to comply with regulations and limitations of the equipped VHF/UHF radios. Recall the UAVs and ground robots are connected via BLE links to survivors in the Lower Tier. Accordingly, when a user wants to communicate with another user located in a different—and possibly distant—mesh network, it generates a Lower Tier packet with the user ID (i.e., the phone number) of the requested user.

The packet is then sent **over BLE** to the closest robot to query the Upper Tier. The Lower Tier packet is received at the robot side, where the ONP decapsulates the information about the user that generated the query and creates an Upper Tier Request packet that is broadcasted to all the robots in proximity. A request packet contains (i) the requested name; (ii) the User ID of the user generating the request, and (iii) the goTenna ID of the request originator.

If the requested user is present at the first hop, the robot that has the requested user in its Routing Table, respond with a reply message, which is essentially an ACK. When a reply packet is received by the robot that initially generated the request, the ONP creates an entry in the Routing Table and seamlessly show the newly added user in the app dashboard of the one that generated the request. The two users involved in the request process will appear as a new members of the respective meshes and a Routing Update will be generated

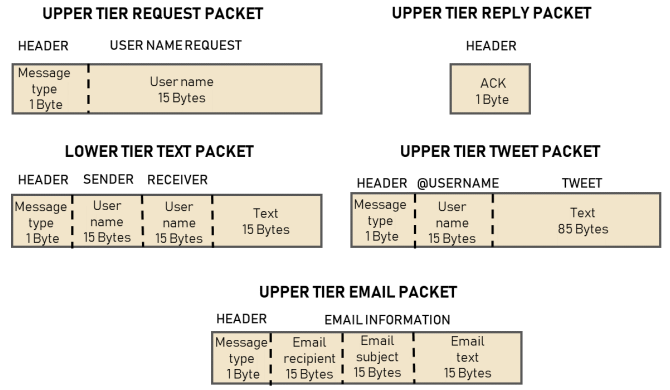


Figure 6: Upper Tier packet format.

only for these two users. The peculiarity of extra-mesh users is that their entry in the Routing Table data structure contains a goTenna ID as next hop instead of a ID. This enables HIRO-NET to structurally treat users seamlessly and allows users to initiate the communication via text. When a text message is generated for an extra-mesh discovered user, the message is routed to the robot that generated/responded the request. At the robot side, the ONP will decapsulate the Lower Tier text message and generate an Upper Tier LTT (Lower Tier Text) message to be sent through goTenna. An LTT message contains (i) Sender ID; (ii) Receiver ID, and (iii) Body of the message.

When the LTT message is received at the robot destination, the ONP decapsulates the packet and generates a Lower Tier BLE message to be sent to the intended recipient.

### 5.1 Internet Sharing

As we mentioned in Section 3, each routing table entry contains the information about whether a user an Internet connection to share and Internet access information is propagated through the mesh by standard Lower Tier packets. At the Lower Tier, when a user wants to send a tweet to a certain Twitter account, it generates a Lower Tier packet that contains the Twitter user name and the body of the tweet. The HIRO-NET Twitter account will tweet to the selected user with the text specified in the body. Similarly, when a user wants to send an email to a certain email account, it generates a Lower Tier packet that contains the email address, the subject and the body of the email. The HIRO-NET Gmail account will send an email to the selected address with the specified subject and body. These packets are routed to the node that has Internet connection. If Internet connection is not available in a mesh, but a robot has discovered the mesh, the HIRO-NET ONP routes the Tweet or Email packet to the closest robot, which generates an Upper Tier Tweet or an Upper Tier email packet. This packet is broadcasted to other robots in proximity and, if a user has Internet connection in its local mesh network, the ONP routes a Lower Tier Tweet or Email packet to them. The full list of Upper Tier packets format is shown in Figure 6.

It is worth mentioning that HIRO-NET users can at any time decide whether or not activating their BLE connection and join HIRO-NET. Also they can specify their willingness to share their internet connection with other survivors (a feature that is disabled by default) from the HIRO-NET application settings. Although we are aware that a higher network

activity leads to increased battery consumption, HIRO-NET aims to foster virtuous behavior in critical conditions, such as natural disasters. In difficult times, HIRO-NET can provide a solid emergency framework at little cost to the single user device so as the community benefits for the capillarity of local meshes connections.

## 6 HIRO-NET OPERATIONS

In this section, we describe the core operations of the HIRO-NET framework. Specifically, mesh generation operations (Phase I) are detailed in Section 6.1, mesh discovery procedures (Phase II) in Section 6.2, and mesh bridging (Phase III) in Section 6.3.

### 6.1 Phase I: Mesh Generation

HIRO-NET Phase I starts immediately after a natural disaster happens in a certain area. Users scattered all over the area activate the HIRO-NET app and create multiple local mesh networks. Generally speaking the Lower Tier independently takes care of the organization of the local mesh network, but it may happen that not all meshes that forms are connected into one. This could lead to a partition of the network, limiting the scope of the communication among survivors. Not only robots are envisioned to provide Upper Tier capabilities, but when two local mesh networks are found and an overlapping area is identified, robots are sent to a specific bridging point in order to enlarge the merge the two meshes into one. The problem of mesh discovery and mesh bridging is formalized and analyzed in the upcoming sections.

### 6.2 Phase II: Mesh Discovery

In Phase II, HIRO-NET air drones are deployed over the map to search for survivors. Drones are stored at multiple BO in the disaster stricken area. Their primary objective is discover as many mesh networks as possible, and provide the BO with their position. On one hand, first responders aim at collecting as much information as possible. On the other hand, drones are battery-powered and have a limited life-time, typically in the order of 20 – 30 minutes.

Let  $\mathcal{B}$  be the set containing all the BO in the stricken area. Each BO  $b \in \mathcal{B}$  is associated to a 2-tuple  $(x_b, y_b)$  that represents its position in the area.

In order to improve the effectiveness of search operations, each BO is provided with statistical information about possible survivors' positions. In our system, such an information is represented by a set  $\mathcal{P}$  of the so-called *Points of Interest* (PoIs). PoIs are pre-determined areas such as schools, hospitals, stadiums, shops and gathering areas where survivors are expected to be located with high probability. In general, PoIs might be associated with side information such as the expected number of individuals at different parts of the day (e.g., a school would not be considered a PoI if the disaster strikes at night).

Although the introduction of PoIs makes it possible to discretize the area to be explored, their number is generally high and it might not be possible to visit all of them in a short time with a single drone. Given the location of all BOs in the area, we generate a Voronoi Diagram that clusterizes the area to be explored. Let  $\mathcal{V} = \{v_b\}_{b \in \mathcal{B}}$  be the set of Voronoi

Regions, where  $v_b$  represents the region whose generating point is the BO  $b \in \mathcal{B}$ . Each region  $v_b$  contains a subset  $\mathcal{P}_b \subseteq \mathcal{P}$  of PoIs whose position lies inside the Voronoi region  $v_b$ . Of course, we have that  $\mathcal{P} = \bigcup_{b \in \mathcal{B}} \mathcal{P}_b$ . We assume that each BO  $b$  is equipped with a  $n_b$  flying drones. Also, we leverage the spatial correlation among PoIs in proximity to speed up the survivor search phase. Specifically, for each Voronoi region  $v_b$ , the corresponding PoI set  $\mathcal{P}_b$  is partitioned into  $n_b$  partitions, and each partition is assigned to exactly one of the drones deployed at  $b$ . As an example, a suitable partitioning of  $\mathcal{P}_b$  can be achieved by using *k-means* clustering algorithm with  $k = n_b$  partitions. For each BO  $b$ , let  $\mathcal{P}_{b,i}$  be the partition of  $\mathcal{P}_b$  assigned to the  $i$ -th drone. The objective of Phase II is to visit as many PoIs in  $\mathcal{P}_{b,i}$  as possible while guaranteeing that the drone returns back to the BO to be recharged for a successive exploration round. That is, for each drone  $i$  we need to compute an *exploration path*  $\tau_i \subseteq \mathcal{P}_{b,i}$ . Each PoI  $p$  is associated to a 2-tuple  $(x_p, y_p)$  that represents its position in the map, and any exploration path is an ordered sequence of PoIs to be visited. Also, it is worth noting that it suffices to visit each PoI once, i.e., only one drone should be sent to visit a given PoI. Let  $d(a, b)$  be a *distance function* that measures the time needed by the flying drone for reaching point  $b$  from point  $a$ <sup>1</sup>. Without loss of generality, we assume that  $d(a, b) = d(b, a)$ . The above problem can be modeled as a longest path problem over an undirected complete graph where: i) each visited PoI provides a unitary reward; and ii) PoIs can be visited multiple times but the reward is obtained only once. Let  $\tau_i = (p(1), p(2), \dots, p(|\tau_i|))$  be the *exploration path* for drone  $i$  represented by an ordered sequence of PoIs, and let  $p_i$  be the PoI representing the BO associated to drone  $i$ . Since the drone must return to the BO before it runs out of battery, we define  $\mathcal{T}_i$  as the set of all the feasible paths for drone  $i$  such that  $p(1) = p(|\tau|) = p_i$  for any  $\tau \in \mathcal{T}_i$ .

Accordingly, the mesh discovery problem can be stated as

$$\underset{\tau_i \in \mathcal{T}_i}{\text{maximize}} \quad |\tau_i| \quad (1)$$

$$\text{subject to} \quad \sum_{k=1}^{|\tau_i|} d(p(k-1), p(k)) \leq D_i \quad (2)$$

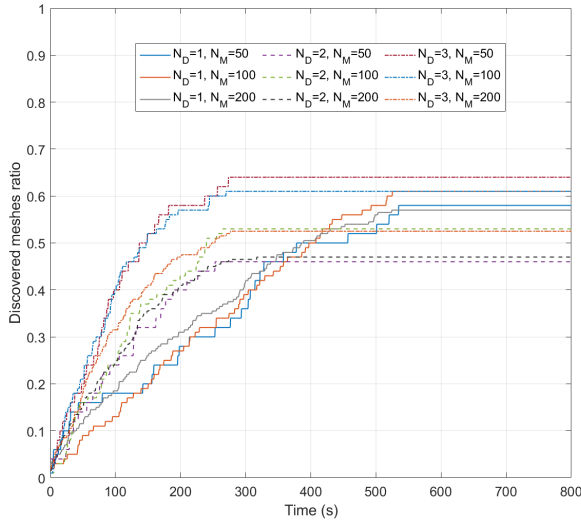
$$p(k) \neq p(m), \forall p(k) \in \tau_i, p(m) \in \tau_i. \quad (3)$$

where  $D_i$  is the drone's maximum flight time, Constraint (2) ensures that the drone goes back to the BO to recharge batteries, and Constraint (3) guarantees that each PoI is visited at most once. Unfortunately, Problem 1 is not easy to be solved. Specifically, Theorem 1 shows that Problem (1) is NP-hard.

**Theorem 1.** *The longest path problem in Problem 1 is NP-hard.*

*Proof:* Let us first note that the relationship  $|\tau_i| \leq |\mathcal{P}_{b,i}|$  always holds for any  $\tau_i \in \mathcal{T}_i$ . That is, there is no solution to Problem 1 that is better than the one where all PoIs are explored by the drone. Let us consider an instance of Problem 1 where there exists a subset  $\tilde{\mathcal{P}}_{b,i} \subset \mathcal{P}_{b,i}$  such that  $d(p', p'') > D_i$  for any  $p', p'' \in \tilde{\mathcal{P}}_{b,i}$ . That is, we assume that there is a subset of PoIs that cannot be directly reached by any other PoI in the same subset. Since the drone

1. In our model, points  $a, b$  are used to represent both PoIs and BOs.



**Figure 7:** Discovered meshes for different number of drones and meshes.

must return to the corresponding BO  $p_i$ , Problem 1 can be restated as follows: *can we found an exploration path  $\tau_i$  such that all PoIs are explored? If not, what is the best exploration path that explores the highest number of PoIs?* The first question is equivalent to finding an Hamiltonian cycle (i.e., a path that explores all vertices of a graph exactly ones) over the graph whose vertices are represented by the PoIs in  $\mathcal{P}_{b,i}$ . Due to the existence of  $\tilde{\mathcal{P}}_{b,i}$ , such a graph is not complete, and to check the existence of an Hamiltonian cycle in such a graph is well-known to be NP-complete [40]. We have now found a polynomial time reduction of the Hamiltonian Cycle problem to an instance of Problem 1. Since such an instance is NP-complete, we can conclude that, unless  $P \neq NP$ , Problem 1 is NP-hard.  $\square$

It is worth noting that it is always possible to compute an optimal solution offline, and the computed exploration path can be pre-loaded on the drone. However, we must stress the fact that while discovering the network, information with respect to discovered meshes and visited PoIs is constantly updated. Such an information can be leveraged to update the exploration path, for example, by avoiding or removing those PoIs where the presence of local meshes has already been discovered. Figure 7 depicts the discovery ratio (i.e., number of discovered meshes over number of total meshes) at varying number of drones obtained by using the emulator we discuss in Section 8. It is interesting to note that the ratio stops increasing after a while. This is due to the fact that the drone discovered all meshes on its path but its path didn't cover the global number of meshes.

It is worth to mention that although Constraint (2) ensures that the drone flies back to the base of operations to recharge batteries, it does not consider the realistic case where visiting one or more PoIs included in the static flight plan is no longer needed because, for example, meshes of survivors in the area have already been discovered, or the PoI has already been visited by another UAV. In this case, visiting these PoIs would not result in any practical gain, and

would only impact UAV flight lifetime due to the unnecessary battery consumption.

To overcome these inefficiencies, in the next section we provide a dynamic version of Problem 1 that leverages information updates received from other drones and BOs to update the exploration path in an online fashion and include new PoIs to the flight plan.

### 6.2.1 Dynamic Exploration Mission Path

As previously mentioned, drones first follow a static pre-loaded optimal exploration mission. However, drones themselves are HIRO-NET nodes and as soon as they connect to a local mesh network, they immediately get a Routing Update containing the topology of such a network.

Let us assume that updated information is sent to drone  $i$  at time instant  $t$ , and let  $\tau_i(t-1)$  be the exploration path up to time instant  $t$ . Two cases should be considered. If the received information requires the removal of PoIs from  $\tau_i(t-1)$ , a new exploration path, say  $\tau_i(t)$ , should be computed. Otherwise, the drone will still follow the previous exploration path and  $\tau_i(t) = \tau_i(t-1)$ .

Let us focus on the case where the exploration must be updated. Let  $D_i^R(t) \leq D_i$  be the residual flight time at time  $t$ . An updated optimal exploration path  $\tau_i(t)$  can still be computed by solving Problem (1) by substituting  $D_i$  with  $D_i^R(t)$  in Constraint (2). However, Problem 1 is NP-hard and it is unlikely for a small drone to be able to independently compute the optimal solution with limited computational resources. Therefore, a low-complexity approach should be considered.

To achieve low-complexity and efficiency, in the following we propose a greedy algorithm to compute a sub-optimal solution to the longest path problem.

Let  $p_i(t) = (x_i(t), y_i(t))$  be the position of the drone at time instant  $t$ , and  $\mathcal{P}_{b,i}^*$  be the set of PoIs that have been either already visited, or their exploration is no longer needed. To continue the mesh discovery phase, a new exploration path needs to be computed.

The proposed greedy online exploration path computation algorithm works as follows: i) an empty exploration path  $\tilde{\tau}_i$  is created; ii) the position of the drone at iteration  $\nu$  is set to  $p(\nu)$ . If  $\nu = 1$ ,  $p(1) = (x_i(t), y_i(t))$ , otherwise,  $p(\nu) = \tilde{\tau}_i(\nu-1)$ ; iii) we select  $\tilde{p} = \arg \min_{p \in \tilde{\mathcal{P}}_{b,i}} d(p(\nu-1), p)$ ; iv) if  $d(p(\nu-1), \tilde{p}) + d(\tilde{p}, p_{b,i}) \leq D_i^R(t)$ , we set  $\tilde{\tau}_i(\nu) = \tilde{p}$ ,  $\nu = \nu + 1$ ,  $D_i^R(t) = D_i^R(t) - d(p(\nu-1), \tilde{p})$ ,  $\tilde{\mathcal{P}}_{b,i} = \tilde{\mathcal{P}}_{b,i} \cup \tilde{p}$  and we restart from step ii). Otherwise, we force the drone to go back to the corresponding BO, i.e.,  $\tilde{\tau}_i(\nu) = p_b$ .

From the computational complexity point of view, only Line 6 requires  $\mathcal{O}(|\tilde{\mathcal{P}}|)$ , while the remaining operations have  $\mathcal{O}(1)$  complexity. Accordingly, the proposed algorithm has linear complexity  $\mathcal{O}(|\tilde{\mathcal{P}}|)$ . In general, the complexity of Algorithm 1 is  $\mathcal{O}(P)$ , with  $P$  being the total number of PoIs in the area. This shows that, not only Algorithm 1 can be executed in a fully distributed fashion, but the complexity grows linearly with the number of PoIs, which makes it possible to launch Algorithm 1 on resource-constrained devices such as drones. An example of the simulated algorithm is depicted in Figure 8.

### Algorithm 1 Greedy Online Exploration Path Computation

```

1: Output : A greedy exploration path  $\tilde{\tau}$  from  $p_0$  to BO at
    $p_{BO}$ 
2: procedure GREEDYPATH( $p_0, \tilde{P}, D^R, p_{BO}$ )
3:    $\nu \leftarrow 1$ 
4:    $\tilde{\tau} \leftarrow \{p_0\}$ 
5:   while  $D^R \geq d(\tilde{\tau}(\nu), p_{BO})$  and  $\tilde{P} \neq \emptyset$  do
6:      $\tilde{p} \leftarrow \arg \min_{p \in \tilde{P}} d(\tilde{\tau}(\nu), p)$ 
7:     if  $d(\tilde{\tau}(\nu), \tilde{p}) + d(\tilde{p}, p_{BO}) \leq D^R$  then
8:        $\tilde{\tau}(\nu + 1) \leftarrow \tilde{p}$ 
9:     else
10:       $\tilde{\tau}(\nu + 1) \leftarrow p_{BO}$ 
11:      $\nu \leftarrow \nu + 1$ 
12:   return  $b$ 

```

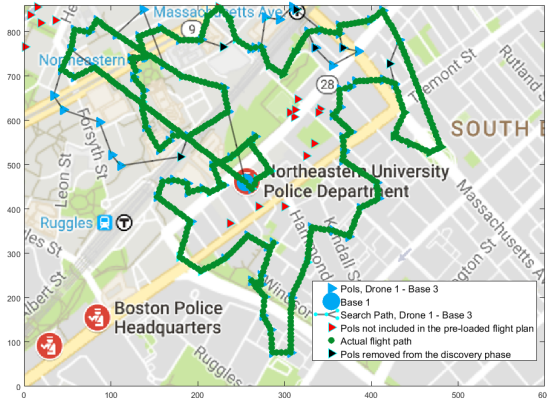


Figure 8: Mesh discovery Process in Phase II.

### 6.3 Phase III: Mesh Bridging

Let  $\mathcal{M}$  be the set of discovered mesh networks, and let  $\mathcal{D}$  be the set of candidate deployment region to bridge multiple mesh networks. For each region  $d \in \mathcal{D}$  and mesh network  $m \in \mathcal{M}$ , let us define the *covering function*  $\chi(d, m) = \{0, 1\}$ . Specifically,  $\chi(d, m) = 1$  if the deployment of a bridging robot in region  $d$  makes it possible to serve users in mesh network  $m$ . Otherwise,  $\chi(d, m) = 0$ . The set of mesh networks covered by each deployment region  $d$  can be obtained as  $\mathcal{M}_d = \{m \in \mathcal{M} : \chi(d, m) = 1\}$ . Thus, the number of meshes covered by deployment region  $d \in \mathcal{D}$  is  $n_d = |\mathcal{M}_d|$ . Let  $\delta = \{d\}_{d \in \mathcal{D}} \subseteq \mathcal{D}$  be a *deployment policy*, i.e., a set of deployments point where to deploy one or more bridging robots. For any deployment policy  $\delta$ , let  $\mathcal{M}^C(\delta) \subseteq \mathcal{M}$  be set of mesh networks covered by  $\delta$ . Such a set can be represented as  $\mathcal{M}^C(\delta) = \bigcup_{d \in \delta} \mathcal{M}_d$ .

The objective of the bridging phase is to find a deployment policy  $\delta$  that maximizes the network coverage and reaches as many survivors as possible. Since the number of available robots at each BO is limited and generally small, their efficient deployment should be considered. Let  $R$  be the available number of robots at each BO, and let  $\mathcal{P}$  be the set of feasible deployment policies. The network coverage

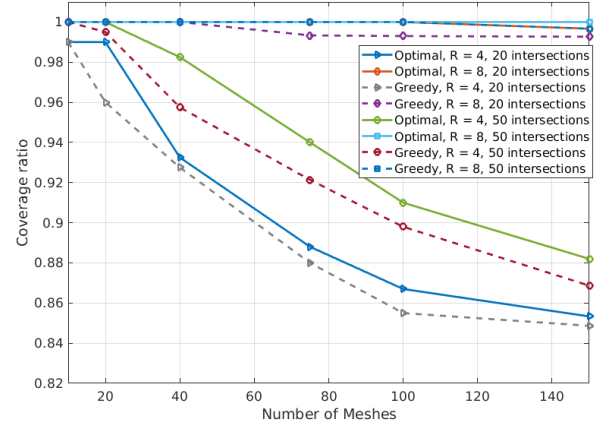


Figure 9: Meshes bridged at varying number of robots  $R$  and mesh intersections.

maximization problem can be formulated as follows:

$$\underset{\delta \in \mathcal{P}}{\text{maximize}} \quad \sum_{m \in \mathcal{M}} |\mathcal{M}^C(\delta)| \quad (4)$$

$$\text{subject to} \quad |\delta| \leq R. \quad (5)$$

Constraint (5) in Problem 4 ensures that the number of deployed robots does not exceed the actual robots availability. For any deployment policy  $\delta$ , let  $x_d = \{0, 1\}$  be an indicator variable to indicate whether or not the deployment region  $d \in \mathcal{D}$  belongs to  $\delta$ . In more detail,  $x_d = 1$  when  $d \in \delta$ , and  $x_d = 0$  otherwise. Problem 4 can be restated as follows:

$$\underset{\mathbf{x}}{\text{maximize}} \quad \sum_{m \in \mathcal{M}} \phi_m(\mathbf{x}) \quad (6)$$

$$\text{subject to} \quad \sum_{d \in \mathcal{D}} x_d \leq R \quad (7)$$

$$x_d \in \{0, 1\} \quad (8)$$

where  $\mathbf{x} = \{x_d\}_{d \in \mathcal{D}}$ , and  $\phi_m(\mathbf{x})$  is defined as

$$\phi_m(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{d \in \mathcal{D}} x_d \chi(d, m) \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where we recall that  $\chi(d, m) = 1$  if  $m \in \mathcal{M}_d$ , i.e., deployment point  $d$  covers mesh  $m$ , and  $\chi(d, m) = 0$  otherwise. Intuitively, the utility function of Problem 6 represents the number of mesh covered by deployment policy  $\delta$ . Furthermore, since all the optimization variables in Problem 6 are 0-1 valued variables and the constraints are linear, the problem is an Integer Linear Problem (ILP). It is well-known that the class of ILP problems is generally hard to be solved. It is easy to show that Problem 6 can be formulated as a Maximum Coverage Problem, which is well-known to be NP-hard [41]. Despite its complexity, such a problem has been extensively investigated in the literature and many efficient solutions have been proposed by using Branch-and-Bound (BB), Lagrangian relaxation, greedy [41] and approximation [42] algorithms. In Figure 9, we present results obtained via the emulator described in Section 8 and show the performance of our greedy approach to mesh bridging, compared to the optimal solution by means of coverage ratio (i.e., number of covered meshes over number of all meshes).

## 7 EXPERIMENTAL RESULTS

HIRO-NET currently has a working prototype that has been extensively tested in all its components. All videos of HIRO-NET experiments and Internet sharing functionalities can be appreciated at [43]. A picture of HIRO-NET nodes is shown in Figure 10 while Figure 11 shows the hardware architecture of our prototype.

### 7.1 Lower Tier

Lower Tier performance has been extensively analyzed in order to evaluate the viability of HIRO-NET as a real life solution for emergency situations. The relevant metrics to evaluate in mesh networks are convergence time and end-to-end delay. By convergence we mean that every node in the network is able to communicate to every other node in the network. We tested the Lower Tier by creating a 16 devices mesh network using 4 Android phones and 12 Raspberry Pis running Android Things. Phones were acting as Servers and Raspberry Pis were acting as Clients. Every Server phone had 3 Raspberry Pi Clients and every Server was connected at most to two other Servers. In all configurations, every node sequentially entered the network with a random wait between 0 and 15 seconds after the previous node joined the network. Convergence time can be seen in Figure 12. Results show how every introduction of a new Server introduces a consistent delay (e.g the convergence time steepens going from 8-9 nodes and from 12-13 nodes). Figure 13 shows the end-to-end delay, intended as a time to deliver a full message between two any nodes in the mesh. As expected, increasing the number of devices in the lower-tier results in increased end-to-end latency due to collisions and increased medium-access procedures. Similarly, the higher the number of hops to reach the destination, the higher the end-to-end latency. It is worth mentioning that although the latency increases with the number of devices, the increase is almost linear and not exponential. This is due to the fact that nodes in dense meshes are likely to be closer to each other, and thus latency can be mitigated by using next hops that are closer to the intended destination.

Upper Tier performance has been evaluated to determine responsiveness and scalability of the HIRO-NET system. As goTennas offer a maximum payload size of 236 bytes with a maximum of 5 total transmissions allowed per end-user per

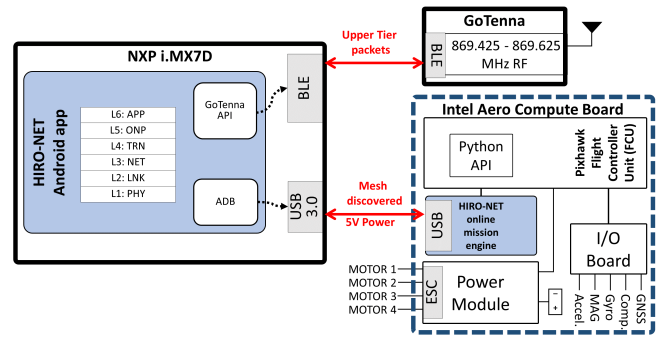


Figure 11: HIRO-NET Hardware architecture.

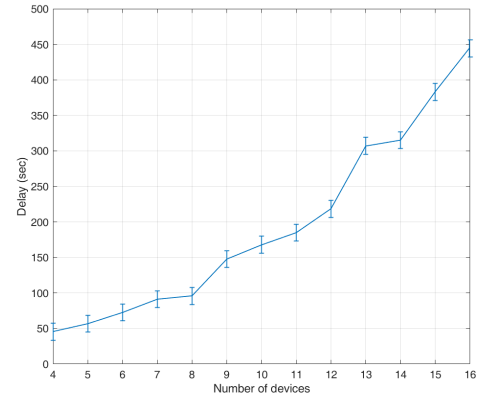


Figure 12: Lower Tier convergence time at varying dimension of the mesh.

minute which amounts to a bitrate of 156 bps, we evaluated transmission delay of every packet. Experiments were conducted in Europe where goTenna operates at 869MHz in a suburban area.

Figure 14 shows the Upper Tier packet transmission delay for different distances. As expected, delay increases with the increase of the dimension of the packet to send, e.g., from User request packets (smallest) to Email packets (largest). For instance, the transmission of an Email packet takes almost one second more than the time required to transmit a User request packet. Although Figure 14 shows that the distance between the communicating nodes minimally impacts delay measurements, it is worth noticing that delay measurements show fluctuations (up to 400 ms in the case of Tweet packets) within measurements related to the same packet type, mostly due to the drones moving in an urban environment. After extensive experiments, we observed similar transmission times for a dense urban environment, where the transmission range decreased to 800 meters, and for a rural environment, where the transmission range increased up to 1600 meters. Transmission times did not change consistently while varying environments, so they are omitted for the sake of brevity. By using GoTenna, HIRO-NET can seamlessly be used in diverse emergency environment ranging from sparse to very dense network deployments with minimal to negligible performance loss. Results suggests that as many as 74 robots are sufficient to create a temporary network that covers an area as large as Manhattan.



Figure 10: HIRO-NET aerial node prototype.

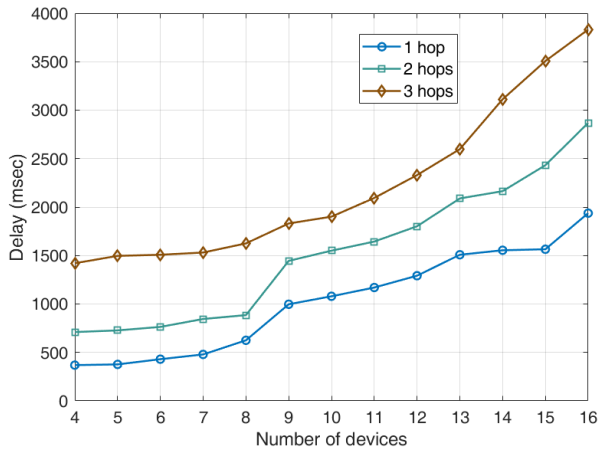


Figure 13: Lower Tier end-to-end delay.

## 7.2 Network Lifetime

To evaluate the feasibility of HIRO-NET network in real life disaster scenarios we performed extensive tests on network lifetime and battery drain. Regarding the Upper Tier, although GoTenna claims a nominal battery life of 24h in standby [44], the use of such radios are meant to inherently last for shorter periods. In fact, HIRO-NET aerial nodes only leverage Upper Tier functionalities during flight time which are consistently shorter than GoTenna radios battery lifetime. As for HIRO-NET ground robots, that aim to be stationary at bridging points, they are envisioned to be powered up by larger batteries and therefore do not represent a critical point of the system. For these reasons, we focused our experiments mainly on Lower Tier battery consumption and discharge rate. As we did for convergence time and end-to-end delay experiments, we assumed to create Lower Tier meshes composed by one Server towards which are connected up to 3 clients. For these experiments, we used 4 identical Samsung S6 equipped with a 2800 mAh battery. First we started our devices in Airplane mode except for Bluetooth Low Energy and started the HIRO-NET app to scan the environment for other HIRO-NET nodes and let the battery run out while we logged battery level. We used such battery times as baseline for next experiments. No remarkable difference was observed between Clients and Servers

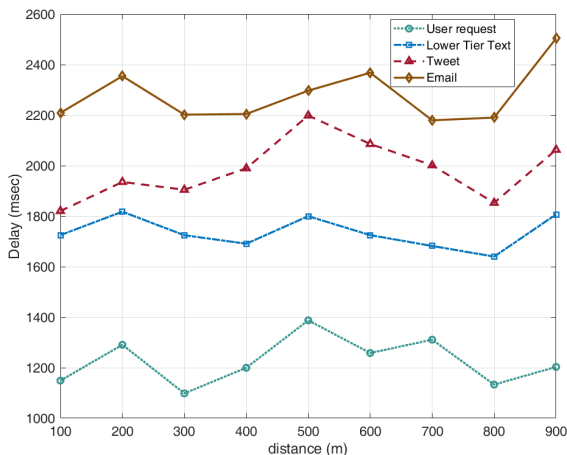


Figure 14: Upper Tier packets transmission delay.

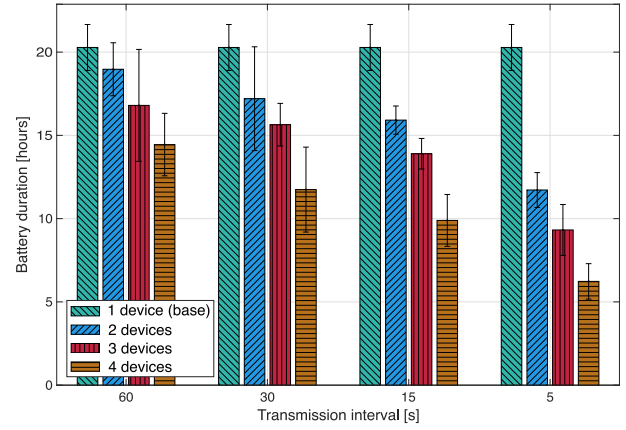


Figure 15: Lower Tier battery duration.

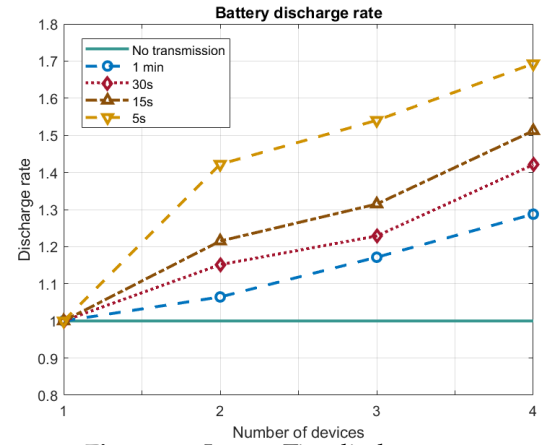


Figure 16: Lower Tier discharge rate.

battery consumption baseline, as the app was impacting on the battery only during the low-level BLE scan phase. We've repeated this battery draining test by generating mockup traffic in the mesh network at difference transmission interval.

## 7.3 Upper Tier

Results are shown in Figure 15. It is possible to note that the overall network lifetime of HIRO-NET on smartphones battery highly depends on the number of devices and on the traffic. Obviously this setup is usually hardware dependent as different smartphones use different BLE chips and different battery capacity. To better visualize the behaviour of HIRO-NET Lower Tier mesh at varying number of nodes and traffic loads, Figure 16 shows the discharge rate of smartphone batteries. It is possible to note that other than the number of devices in a mesh, what impacts the most is the transmission interval. In fact, under stress conditions, battery drains 1.4x to 1.7x faster than with respect to not using HIRO-NET.

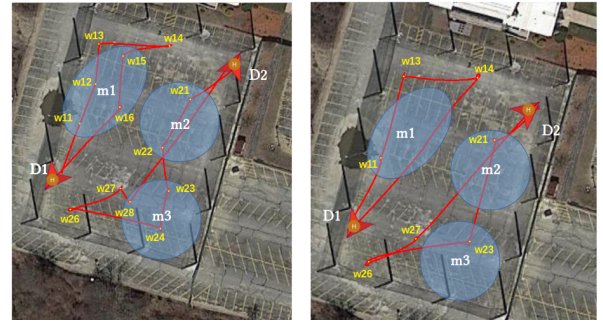
## 7.4 Indoor Drone Discovery

HIRO-NET drone discovery functionalities have been implemented on Intel Aero Ready To Fly models carrying an NXP Imx7dpico board that runs the HIRO-NET app on Android Things. On the Intel Aero a Python script read information from the HIRO-NET app via serial USB using ADB (Android

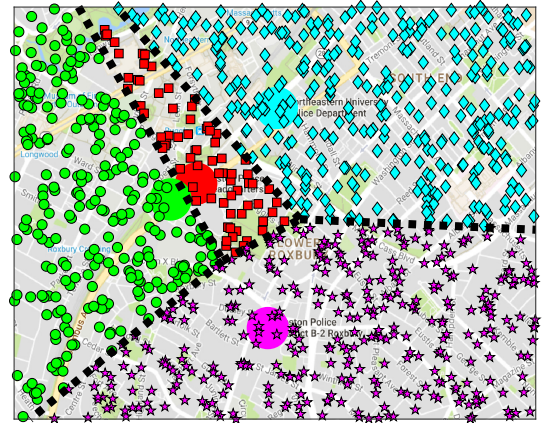
Debug Console). At the same time, periodical checks on the routing table of HIRO-NET ONP compared the discovered meshes geographical extension to the mission waypoints. If an upcoming waypoint is in the area covered by a newly discovered mesh, HIRO-NET drone updates mission waypoints as from Phase II and sends updated to the Flight Controller using the Dronekit library. Experiments were conducted indoor. In Figure 17 a screenshot from the drone control software Qgroundcontrol compares the original flight path and the HIRO-NET enabled flight path.

## 7.5 Outdoor Multi-drone Discovery

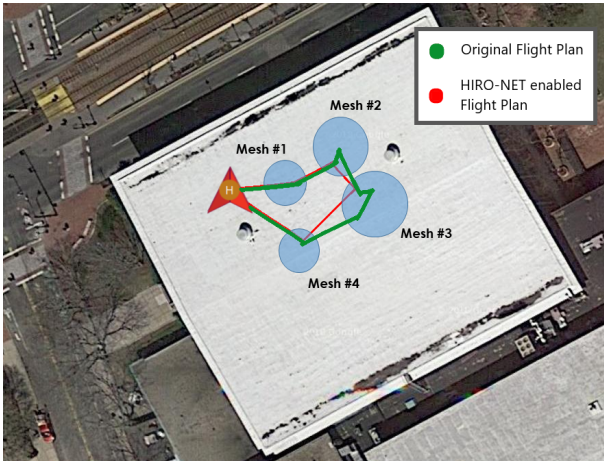
After testing indoor the ability of HIRO-NET, more test were conducted outdoor. We've conducted discovery experiments in the drone cage at Northeastern University's Kostas Research Institute to validate the effectiveness of HIRO-NET online mission engine. Snapshots of Qgroundcontrol software showing two different experiments in the same scenario are shown in Figure 18. Throughout the cage were deployed mesh networks, namely  $m_1$ ,  $m_2$  and  $m_3$ . Two HIRO-NET aerial nodes,  $D_1$  and  $D_2$  were deployed to explore half of the cage each. On the left, the HIRO-NET mission engine was disabled, so drones were just following their offline uploaded mission. Specifically, prior to return to the take-off point  $D_1$  travels to  $w_{11}, w_{12}, w_{13}, w_{14}, w_{15}, w_{16}$  and  $D_2$  travels to  $w_{21}, w_{22}, w_{23}, w_{24}, w_{25}, w_{26}, w_{27}$ . On the left the HIRO-NET mission engine was enabled so the drones don't visit those waypoints that are already covered by the mesh networks. Specifically, prior to return to the take-off point,  $D_1$  travels to  $w_{11}, w_{13}, w_{14}$  and  $D_2$  travels to  $w_{21}, w_{23}, w_{26}, w_{27}$ .



**Figure 18:** Outdoor multi-drone experiments of the HIRO-NET discovery phase



**Figure 19:** A snapshot of BOs and PoIs deployment for a single simulation run.



**Figure 17:** Indoor experiment of the HIRO-NET discovery phase.

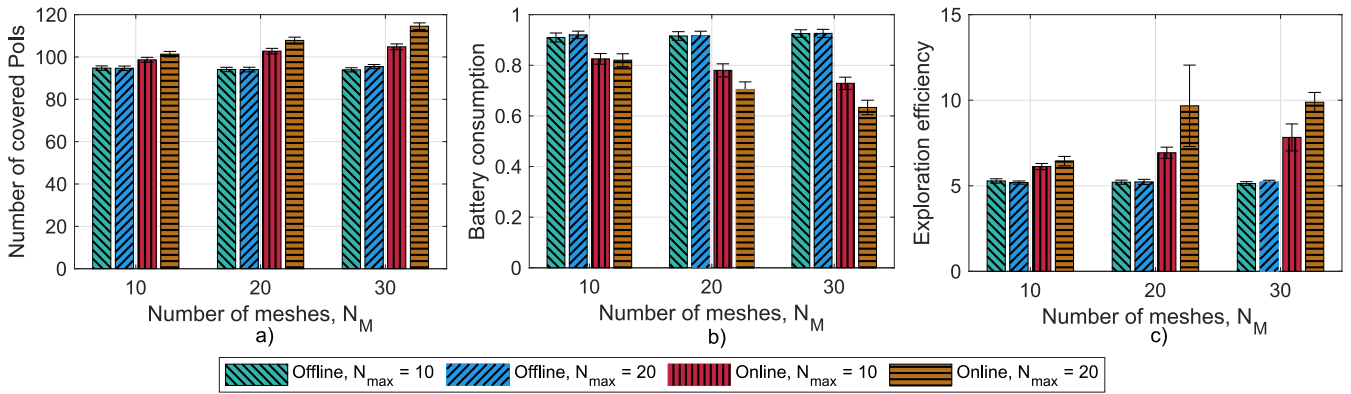
## 8 NUMERICAL RESULTS

To objective of this section is to shed light upon two important aspects of HIRO-NET, i.e., energy consumption and scalability. Due to regulations and the difficulties in deploying a large scale drone network in a practical use-case, we evaluate these two relevant metrics in a MATLAB simulator implemented to emulate the HIRO-NET deployment in city-scale disaster scenarios.

We have considered the scenario depicted in Figure 8. We select 4 BOs, each equipped with 1 HIRO-NET drone.

At each simulation run, we generate 200 PoIs and  $N_M$  survivor's mesh networks, each containing a maximum number  $N_{max}$  of survivors. Survivors generate meshes and connect to HIRO-NET drones via BLE. We assume that communications are subject to free-space path loss and the coverage range of BLE devices is set to 50 meters. Flight speed of drones is set to 3 m/s, and maximum flight time is set to 20 minutes. We now present both average values and 95% confidence intervals over 100 independent simulation runs. An snapshot of the simulated scenario is shown in Figure 19, where we show the 4 BOs (large circle markers), the corresponding Voronoi regions, and the randomly generated PoIs.

As pointed out in Section 6.2.1, HIRO-NET leverages upper-tier networking via goTenna links to continuously notify drones about the status of Phase II (i.e., mesh discovery). More specifically, drones exchange information about which PoIs have been already visited, as well as if discovered meshes extend their coverage to PoIs that have not been visited yet. This information is utilized by HIRO-NET drones to update the exploration path in an online fashion through Algorithm 1. To demonstrate the effectiveness of online exploration path computation over the offline approach where drones follow the pre-loaded exploration path computed by solving (1), in Figure 20 we compare their performance in



**Figure 20:** Comparison of offline and online exploration path algorithms as a function of the number of  $N_M$  mesh networks and maximum number  $N_{max}$  of survivors in each mesh.

terms of discovered PoIs and battery consumption.

The average number of PoIs covered by HIRO-NET drones as a function of different path planning schemes and values of  $N_{max}$  and  $N_M$  is reported in Figure 20(a). It is shown that updating the exploration path via Algorithm 1 always improve the performance of HIRO-NET in terms of PoIs visited by drones. Benefits brought by online exploration increase as the number of meshes and survivors therein increases as well. As an example, Figure 20(a) shows that online exploration path computation allows HIRO-NET drones to visit up to 20 more PoIs if compared to the offline path pre-loaded on drones prior to their deployment.

An important aspect of online path computation is that it removes PoIs already visited by other drones (or covered by already discovered meshes) from the exploration path. This way, Algorithm 1 avoids unnecessary energy consumption to reach these PoIs, which can be instead utilized to explore PoIs that were not included in the original exploration path. Figure 20(b) compares the normalized battery consumption of drones under both online and offline schemes as a function of  $N_{max}$  and  $N_M$ . Figure 20(a) already showed that the online scheme increases the number of PoIs visited by HIRO-NET, more surprisingly, Figure 20(b) shows that this gain can be achieved with even lower battery consumption if compared to the offline scheme. This result stems from the fact that Algorithm 1 removes already visited PoIs while adding PoIs closer to the drone, thus sparing battery lifetime.

These advantages can be summarized by Figure 20(c), where we show the discovery efficiency—measured as the number of visited PoIs per minute—of offline and online schemes as a function of  $N_{max}$  and  $N_M$ . Figure 20(c) shows that in dense network deployments (i.e.,  $N_{max} = 20$  and  $N_M = 20, 30$ ) online exploration path update is twice as effective as the offline scheme, thus demonstrating the scalability of our approach and its effectiveness in city-scale disaster scenarios.

## 9 CONCLUSIONS

In this paper we presented HIRO-NET, a self-organizing emergency mesh network to address the problem of providing connectivity in the immediate aftermath of a natural disaster. HIRO-NET is able to establish a two-tier mesh

network where the Lower Tier connects nearby users via Bluetooth Low Energy and the Upper Tier creates long-range VHF links between autonomous robots exploring the disaster stricken area. HIRO-NET also enables Internet sharing to provide low bit-rate services to survivors. Experimental and simulation results from drone exploration, Lower Tier network convergence and Upper Tier transmission prove that HIRO-NET could be deployed to cover a large metropolitan area to discover survivors and create a temporary infrastructure-less network in tens of minutes. Future work will be focused on increasing the scale of Lower Tier experiments as well as improve network performances and provide experimental results for Phase III. Experiments will be conducted using diverse robotics platform, as well a higher number of robots. Furthermore, set theory will be applied to evaluate the optimality of mesh bridging.

## ACKNOWLEDGMENTS

This article is based upon work supported in part by the U.S. National Science Foundation under grant CNS-1618727 and by the US Air Force Research Laboratory under Contract FA8750-18-C-0122. The authors would like to thank the students Andrea Lacava, Pierluigi Locatelli and Gianluigi Nero for their invaluable help in the implementation of Lower Tier functionalities.

## REFERENCES

- [1] FCC, "Fcc communications status report for areas impacted by hurricane michael," <https://docs.fcc.gov/public/attachments/DOC-357387A1.pdf>, 2018.
- [2] —, "Fcc communications status report for areas impacted by hurricane irma," [https://apps.fcc.gov/edocs\\_public/attachmatch/DOC-346655A1.pdf](https://apps.fcc.gov/edocs_public/attachmatch/DOC-346655A1.pdf), 2017.
- [3] —, "Fcc communications status report for areas impacted by hurricane maria," [https://transition.fcc.gov/Daily\\_Releases/Daily\\_Business/2017/db1006/DOC-347118A1.pdf](https://transition.fcc.gov/Daily_Releases/Daily_Business/2017/db1006/DOC-347118A1.pdf), 2017.
- [4] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks: Small and mini drones," *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 73–82, September 2016.
- [5] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Elsevier Ad Hoc Networks*, vol. 68, pp. 1–15, January 2018.

- [6] H. Kung, C.-K. Lin, T.-H. Lin, S. J. Tarsa, and D. Vlah, "Measuring diversity on a low-altitude uav in a ground-to-air wireless 802.11 mesh network," in *GLOBECOM Workshops (GC Wkshps)*, 2010 IEEE. IEEE, 2010, pp. 1799–1804.
- [7] B. Barritt, T. Kichkaylo, K. Mandke, A. Zalcman, and V. Lin, "Operating a uav mesh & internet backhaul network using temporospatial sdn," in *Aerospace Conference*, 2017 IEEE. IEEE, 2017, pp. 1–7.
- [8] L. Bertizzolo, S. D'Oro, L. Ferranti, L. Bonati, E. Demirors, Z. Guan, T. Melodia, and S. Pudlewski, "SwarmControl: An automated distributed control framework for self-optimizing drone networks," in *Proc. of IEEE INFOCOM*, Toronto, ON, Canada, July 2020.
- [9] F. D'Alterio, L. Ferranti, L. Bonati, F. Cuomo, and T. Melodia, "Quality aware aerial-to-ground 5G cells through open-source software," in *Proc. of IEEE GLOBECOM*, Waikoloa, HI, USA, December 2019.
- [10] L. Ferranti, L. Bonati, S. D'Oro, and T. Melodia, "SkyCell: A Prototyping Platform for 5G Aerial Base Stations," in *Proc. of IEEE Workshop on Wireless Networking, Planning, and Computing for UAV Swarms (SwarmNet)*, Cork, Ireland, August 2020.
- [11] L. Bertizzolo, X. Tran, Tuyen, B. Amento, B. Balasubramanian, R. Jana, H. Purdy, Z. Yu, and T. Melodia, "Live and let live: flying UAVs without affecting terrestrial UEs," in *Proc. of ACM Intl. Workshop on Mobile Computing Systems and Applications (HotMobile)*, Austin, TX, USA, March 2020.
- [12] M. Polese, L. Bertizzolo, L. Bonati, A. Gosain, and T. Melodia, "An experimental mmWave channel model for UAV-to-UAV communications," in *Proc. of ACM Workshop on Millimeter-wave Networks and Sensing Systems (mmNets)*, London, UK, September 2020.
- [13] L. Bertizzolo, M. Polese, L. Bonati, A. Gosain, M. Zorzi, and T. Melodia, "mmBAC: Location-aided mmWave backhaul management for UAV-based aerial cells," in *Proc. of ACM mmNets*, Los Cabos, Mexico, October 2019.
- [14] S. Sekander, H. Tabassum, and E. Hossain, "Multi-tier drone architecture for 5G/B5G cellular networks: Challenges, trends, and prospects," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 96–103, March 2018.
- [15] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in ran heterogeneity: Multi-tier drone-cells," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, November 2016.
- [16] J. Camp, J. Robinson, C. Steger, and E. Knightly, "Measurement driven deployment of a two-tier urban mesh access network," in *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM, 2006, pp. 96–109.
- [17] A. A. Franklin and C. S. R. Murthy, "Node placement algorithm for deployment of two-tier wireless mesh networks," in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*. IEEE, 2007, pp. 4823–4827.
- [18] F. Mezghani, P. Kortoçi, N. Mitton, and M. Di Francesco, "A multi-tier communication scheme for drone-assisted disaster recovery scenarios," in *Proc. of IEEE PIMRC*, Istanbul, Turkey, September 2019.
- [19] G. Kuperman, L. Veytser, B.-N. Cheng, S. Moore, and A. Narula-Tam, "A comparison of OLSR and OSPF-MDR for large-scale airborne mobile ad-hoc networks," in *Proc. of ACM Airborne*, August 2014.
- [20] A. Rovira-Sugranes and A. Razi, "Predictive routing for dynamic UAV networks," in *Proc. of IEEE WiSEE*, Montreal, QC, Canada, October 2017, pp. 43–47.
- [21] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3727–3742, April 2019.
- [22] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647–1650, August 2016.
- [23] B. Galkin, J. Kibilda, and L. A. DaSilva, "Coverage analysis for low-altitude UAV networks in urban environments," in *Proc. of IEEE GLOBECOM*, Singapore, December 2017.
- [24] R. Dai, S. Fotedar, M. Radmanesh, and M. Kumar, "Quality-aware UAV coverage and path planning in geometrically complex environments," *Elsevier Ad Hoc Networks*, vol. 73, pp. 95–105, May 2018.
- [25] S. K. Sudarshan, V. Montano, A. Nguyen, M. McClimans, L. Chang, R. R. Stewart, and A. T. Becker, "A heterogeneous robotics team for large-scale seismic sensing," vol. 2, no. 3. IEEE, 2017, pp. 1328–1335.
- [26] N. Cruz, N. Abreu, J. Almeida, R. Almeida, J. Alves, A. Dias, B. Ferreira, H. Ferreira, C. Gonçalves, A. Martins, J. Melo, A. Pinto, V. Pinto, A. Silva, H. Silva, A. Matos, and E. Silva, "Cooperative deep water seafloor mapping with heterogeneous robotic platforms," in *OCEANS 2017 #8211; Anchorage*, Sept 2017, pp. 1–7.
- [27] H. Hildmann and E. Kovacs, "Using unmanned aerial vehicles (UAVs) as mobile sensing platforms (MSPs) for disaster response, civil security and public safety," *Drones*, vol. 3, no. 3, pp. 59–85, September 2019.
- [28] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, January 2017.
- [29] D. Câmara, "Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios," in *Proc. of IEEE CAMA*, Antibes Juan-les-Pins, France, November 2014.
- [30] J. M. Gregory, I. Brookshaw, J. Fink, and S. Gupta, "An investigation of goal assignment for a heterogeneous robotic team to enable resilient disaster-site exploration," in *Safety, Security and Rescue Robotics (SSRR)*, 2017 IEEE International Symposium on. IEEE, 2017, pp. 133–140.
- [31] J. Xu, K. Ota, and M. Dong, "LUNA: Lightweight UAV navigation based on airborne vision for disaster management," in *Proc. of iThings and IEEE GreenCom and IEEE CPSCom and IEEE SmartData*, Atlanta, GA, USA, July 2019.
- [32] C. Rottondi, F. Malandrino, A. Bianco, C. F. Chiasserini, and I. Stavrakakis, "Scheduling of emergency tasks for multiservice UAVs in post-disaster scenarios," *Computer Networks*, pp. 1–13.
- [33] J. Sánchez-García, D. Reina, and S. Toral, "A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario," *Future Generation Computer Systems*, vol. 90, pp. 129–148, 2019.
- [34] A. Coletta, G. Maselli, M. Piva, and D. Silvestri, "DANGER: A drones aided network for guiding emergency and rescue operations," in *Proc. of Intl. Symp. on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, Virtual Event, USA, October 2020.
- [35] L. Ferranti, S. D'Oro, L. Bonati, E. Demirors, F. Cuomo, and T. Melodia, "HIRO-NET: self-organized robotic mesh networking for internet sharing in disaster scenarios," in *Proc. of IEEE WoW-MoM*, Washington, D.C., USA, June 2019.
- [36] GoTeNNA, "gotenna — text and gps on your phone, even without service," <https://www.gotenna.com>, 2017.
- [37] ROS, "ros.org — powering the world's robots," <https://www.ros.org>, 2018.
- [38] Bluetooth SIG, "GATT Overview," <https://www.bluetooth.com/specifications/gatt/generic-attributes-overview>, 2019.
- [39] A. Lacava, G. Nero, P. Locatelli, F. Cuomo, and T. Melodia, "Demo abstract: BE-Mesh: bluetooth low energy mesh networking," in *Submitted to: 2019 IEEE INFOCOM Demo (INFOCOM 2019 Demo)*, Paris, France, Apr. 2019.
- [40] M. R. Garey, D. S. Johnson, and R. E. Tarjan, "The planar hamiltonian circuit problem is np-complete," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 704–714, 1976.
- [41] S. Khuller, A. Moss, J. Naor et al., "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.
- [42] Y. Filmus and J. Ward, "The power of local search: Maximum coverage over a matroid," in *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, vol. 14. LIPIcs, 2012, pp. 601–612.
- [43] WinesLab, "Hiro-net youtube channel," <https://www.youtube.com/channel/UckEtseZk7SfMGNkk1222igg>, 2019.
- [44] GoTeNNA, "gotenna mesh specs," <https://gotennamesh.com/pages/gotenna-mesh-specs>, 2020.



**Ludovico Ferranti** (S'19) received his Master of Science in Engineering in Artificial Intelligence and Robotics (110/110) at Sapienza - University of Rome in 2015. He is a PhD student at Sapienza - University of Rome at DIET (Dipartimento di Ingegneria Elettronica e Telecomunicazioni) and Northeastern University of Boston at Graduate School of Engineering from November 2015. His research interests include Unmanned Aerial Networks, Emergency Networks, Wireless Sensor Networks and Robotics

Swarms Algorithms.



**Salvatore D'Oro** (S'12, M'17) received his Ph.D. degree from the University of Catania in 2015. He is currently a Research Assistant Professor at Northeastern University. He serves on the Technical Program Committee (TPC) of Elsevier Computer Communications journal and the IEEE Conference on Standards for Communications and Networking (CSCN) and European Wireless. He also served on the TPC of Med-Hoc-Net 2018 and several workshops in conjunction with IEEE INFOCOM and IEEE ICC. In

2015, 2016 and 2017 he organized the 1st, 2nd and 3rd Workshops on Competitive and COoperative Approaches for 5G networks (CO-COA). Dr. D'Oro is also a reviewer for major IEEE and ACM journals and conferences. Dr. D'Oro's research interests include game-theory, optimization, learning and their applications to 5G networks. He is a Member of the IEEE.



**Leonardo Bonati** (S'19) received his B.S. in Information Engineering and his M.S. in Telecommunication Engineering from University of Padova, Italy in 2014 and 2016, respectively. He is currently pursuing a Ph.D. degree in Computer Engineering at Northeastern University, USA. His research interests focus on 5G and beyond cellular networks, network slicing, and software-defined networking for wireless networks.



**Francesca Cuomo** received her "Laurea" degree in Electrical and Electronic Engineering in 1993, magna cum laude, from the University of Rome "La Sapienza", Italy. She earned the Ph.D. degree in Information and Communications Engineering in 1998, also from the University of Rome "La Sapienza". From 2005 she is Associate Professor at the University of Rome "La Sapienza" and teaches courses in Telecommunication Networks. Prof. Cuomo has advised numerous master students in computer

engineering, and has been the advisor of 10 PhD students in Networking. Her current research interests focus on: Vehicular networks and Sensor networks, Low Power Wide Area Networks and IoT, 5G Networks, Multimedia Networking, Energy saving in the Internet and in the wireless system. She has participated in several National and European projects on wireless network systems such as the RAMON, VICOM, INSYEME, IST WHYLESS, IST EPERSPACE, IST CRUISE, H2020 symbloTe. Francesca Cuomo has authored over 145 peer-reviewed papers published in prominent international journals and conferences. Her Google Scholar h-index is 28, 3400 citations. Relevant scientific international recognitions: 2 Best Paper Awards, she has been in the editorial board of Computer Networks (Elsevier) and now is member of the editorial board of the Ad-Hoc Networks (Elsevier), IEEE Transactions on Mobile Computing and Sensors (MDPI). She has been the TPC co-chair of several editions of the ACM PE-WASUN workshop, TPC Co-Chair of ICCCN 2016, TPC Symposium Chair of IEEE WiMob 2017, Guest Editor of the SI of Ad-Hoc Networks on "Modeling and Performance Evaluation of Wireless Ad-Hoc Networks", 2015, Guest Editor of the SI on "Green Communications and Networking for IoT", Sensors MPDI, 2018, Guest Editor of the SI on "Recent Developments on Mobile Ad-Hoc Networks and Vehicular Ad-Hoc Networks", Electronics MPDI, 2020. General Co-Chair of the First Workshop on Sustainable Networking through Machine Learning and Internet of Things (SMILING), in conjunction with IEEE INFOCOM 2019; Workshop Co-Chair of Aml 2019; European Conference on Ambient Intelligence 2019. She has been reviewer, panelist and panel Chair (2019) of three panels for the Academy of Finland (Academy of Finland and National Science Foundation 2017) (INDUSTRIAL INTERNET ICT 2023, 2017) (COMMUNICATION ENGINEERING panel 2019). She is IEEE senior member.



**Tommaso Melodia** (M'07, SM'16, F'18) is the William Lincoln Smith Chair Professor with the Department of Electrical and Computer Engineering at Northeastern University in Boston. He is also the Founding Director of the Institute for the Wireless Internet of Things and the Director of Research for the PAWR Project Office. He received his Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2007. He is a recipient of the National Science Foundation CAREER award.

Prof. Melodia has served as Associate Editor of IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, Elsevier Computer Networks, among others. He has served as Technical Program Committee Chair for IEEE Infocom 2018, General Chair for IEEE SECON 2019, ACM Nanocom 2019, and ACM WUWnet 2014. Prof. Melodia is the Director of Research for the Platforms for Advanced Wireless Research (PAWR) Project Office, a \$100M public-private partnership to establish 4 city-scale platforms for wireless research to advance the US wireless ecosystem in years to come. Prof. Melodia's research on modeling, optimization, and experimental evaluation of Internet-of-Things and wireless networked systems has been funded by the National Science Foundation, the Air Force Research Laboratory the Office of Naval Research, DARPA, and the Army Research Laboratory. Prof. Melodia is a Fellow of the IEEE and a Senior Member of the ACM.