

HIRO-NET: Self-Organized Robotic Mesh Networking for Internet Sharing in Disaster Scenarios

Ludovico Ferranti^{*†}, Salvatore D'Oro^{*}, Leonardo Bonati^{*}, Emrehan Demirors^{*}
Francesca Cuomo[†], Tommaso Melodia^{*}

^{*}Northeastern University, Boston, MA - [†]Sapienza University of Rome, Italy
Email: {ferranti.l, s.doro, bonati.l, edemirors, melodia}@northeastern.edu
Email: {ludovico.ferranti, francesca.cuomo}@uniroma1.it

Abstract—In this paper we present HIRO-NET, Heterogeneous Intelligent Robotic Network. HIRO-NET is an emergency infrastructure-less network tailored 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 main goal is to enable users in the disaster to exchange text messages in order 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

I. INTRODUCTION

Recent extreme natural events have shown that cellular infrastructures cannot reliably withstand natural disasters. In particular, up to 90% of cellular infrastructures were knocked down in Florida counties after Hurricane Irma [8] and almost the entirety of the Puerto Rico communications went down after Hurricane Maria [9]. 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 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.
- 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 an 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 II presents the main related works. Section III introduces the system architecture of HIRO-NET and presents the design and implementation of the proposed HIRO-NET prototype.

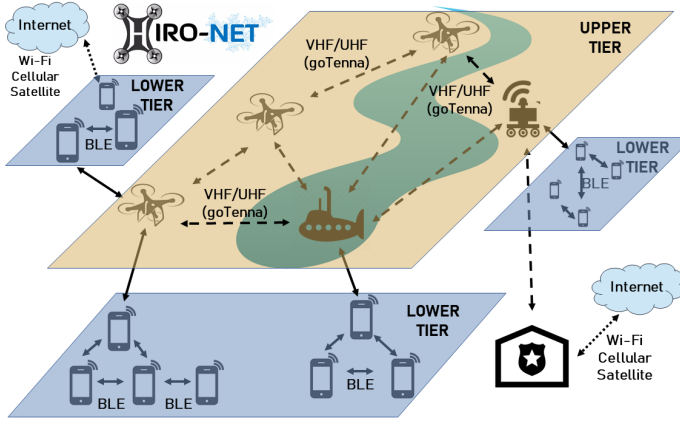


Figure 1: Physical and logical architecture of HIRO-NET: the blue areas are the Lower Tier (BLE), while the orange area represents the Upper Tier (VHF).

In particular, Section IV accurately describes the Lower Tier functionalities of HIRO-NET and Section V explains how such functionalities are extended through the Upper Tier. In Section VI the optimization problems related to HIRO-NET deployment phases are formulated. In Section VII the network deployment and its experimental evaluation are presented. We conclude the paper by discussing the current limitations of HIRO-NET and how we plan to address them in Section IX.

II. RELATED WORK

Recent works proposed the use of air and ground unmanned vehicles for large scale seismic sensing [24] and disaster site exploration [13] but envisioned the use of robots as mere passive tools in the hands of first responders to provide a better knowledge about the environment or to deploy a drop-and-forget sensors network. In [5] and [19] the cooperation of underwater and surface vehicle is presented for deep sea and coral reef exploration. HIRO-NET not only proposes a common framework for air, water and ground unmanned vehicles but also selectively makes use of the ability to move on all terrains to increase the range of the network. Aerial drones have a key role in the discovery phase and efficient communication is critical. Airborne mobile ad-hoc networks have been thoroughly investigated in the past years [25] [16] with special emphasis on the applicability [17][1] and scalability [21] of existing routing protocols. These works back the optimality of the choice of reactive protocols for HIRO-NET Upper Tier routing, although are limited to air-to-air networks and do not well capture the bigger scheme of a two-tier network. Conventional works on two tier networks, on the other hand, rely mostly on fixed nodes [3] or only envision users mobility [11]. HIRO-NET on the other hand, builds a mobile two-tier network based on the position of users in the Lower Tier and provides reconfiguration capabilities to the network topology. The Lower Tier portion of HIRO-NET is a full-fledged mesh network with all related challenges [22]. Despite plenty of mesh networking protocols provide dynamicity [7] and scalability [2], none of them capture the important role of nodes position in a search-and-discovery network. Most of the geographic routing solutions in literature [6] [27], are more suitable for wireless sensor networks fixed nodes and partially envision nodes mobility. HIRO-NET routing

protocol not only provides a tailored solution for the drones discovery phase, but also allows the Internet sharing process to be lightweight and effective. Other works [4], [14] provide high complexity solutions for resource sharing in a mesh network while HIRO-NET Internet Sharing service is oriented to provide simpler and more relevant services in disaster scenarios.

III. 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, (ii) long-range VHF radio based on the goTenna mesh communications suite [12]. 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,

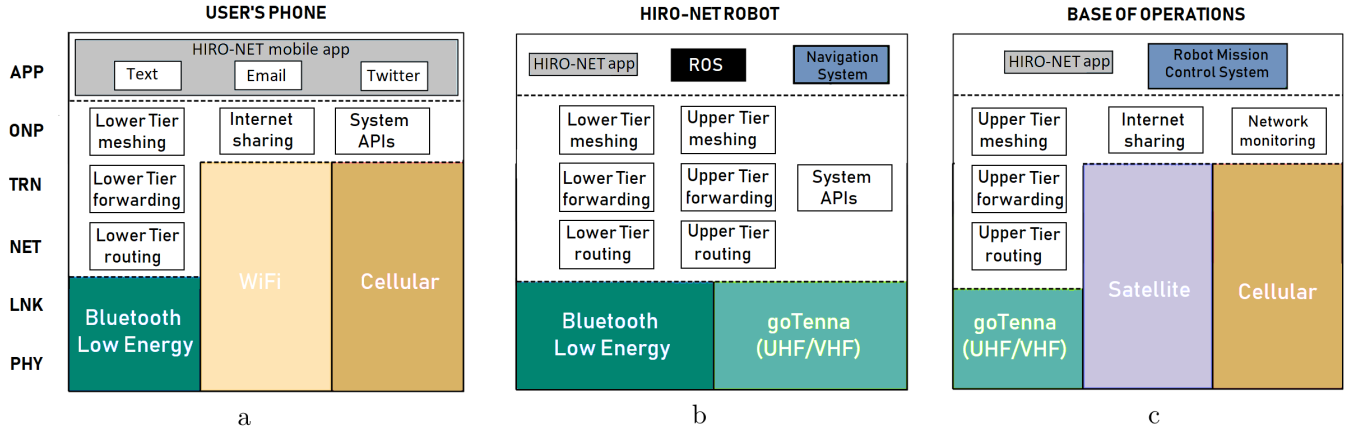


Figure 2: Protocol stack of HIRO-NET nodes.

provided that enough robotic vehicles are dispatched to ensure coverage or bridge disconnected meshes.

A. 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 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 PoIs). 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.

B. 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) [20] 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.

IV. 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; (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 [23]. 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 scans for other Servers in order to being 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 $\in \{1, 7\}$ [18]. 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.

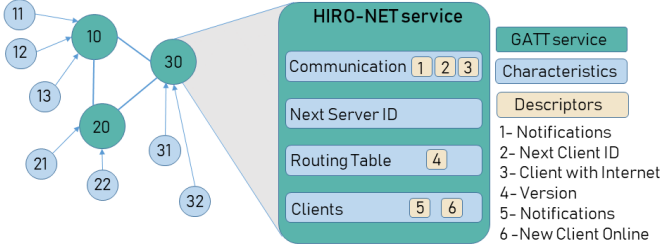


Figure 3: HIRO-NET BLE Gatt Server and topology.

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.

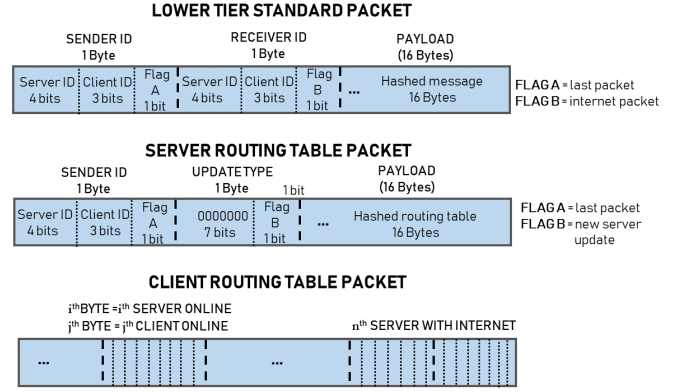


Figure 4: Lower Tier packet format.

From that moment on, users are able to communicate with others by using the Communication characteristic. To match the standard Bluetooth Low 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 represents the online Clients of the network (e.g. if the j^{th} bit of the i^{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^{th} bit set to 1 if i^{th} Server has Internet). This packet modifies the New Client Online characteristic.

The complete list of Lower Tier packets format is shown in Fig:4.

V. UPPER TIER MESHING

The HIRO-NET Upper Tier is constituted by robotic vehicles (UAVs, mobile ground robots and water surface robots) that are equipped with VHF radios. The ONP mediates the interaction between Lower Tier and Upper Tier by ensuring transparency between the two tiers (e.g. the information of Lower Tier Client/Server role of a certain user does not need to be propagated on the Upper Tier). As the VHF band utilized by goTennas ensures a maximum payload size of 236-byte with

a maximum of 5 total transmissions allowed per end-user per minute, the Upper Tier meshing algorithm implemented by HIRO-NET is a reactive protocol. When a user wants to reach out to another user that is located in a distant mesh network, it generates a Lower Tier packet with the user ID (i.e. phone number) of the requested user. The packet is then sent 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; (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 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; (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.

A. Internet Sharing

As we mentioned in Section III, 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 Fig 5.

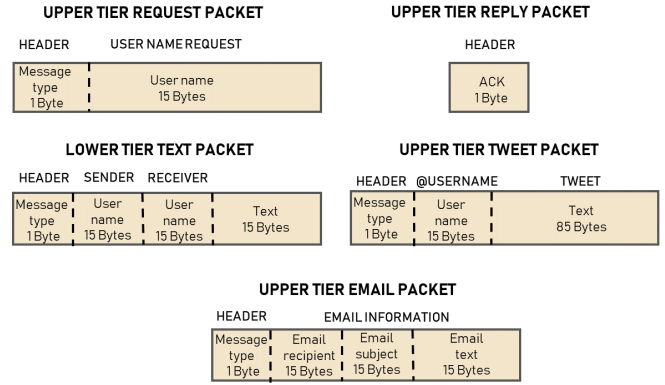


Figure 5: Upper Tier packet format.

VI. HIRO-NET PHASES

A. 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.

B. 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

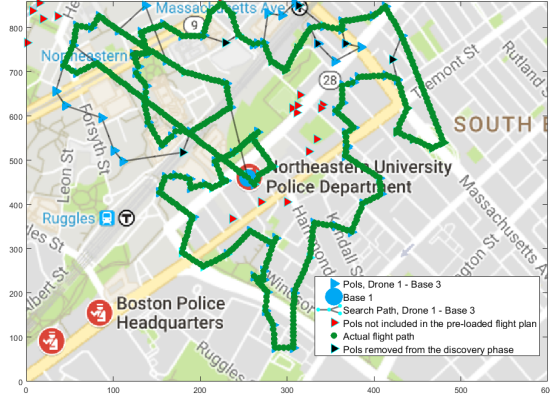


Figure 6: Mesh discovery Process in Phase II.

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 ¹. 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

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

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

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

where D_i is the drone's maximum flight time, Constraint (2) ensures that the drone goes back to the BO to recharge batteries,

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

and Constraint (3) guarantees that each PoI is visited at most once. Unfortunately, Problem 1 is not easy to be solved as it can be proved that it is NP-hard².

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. 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.

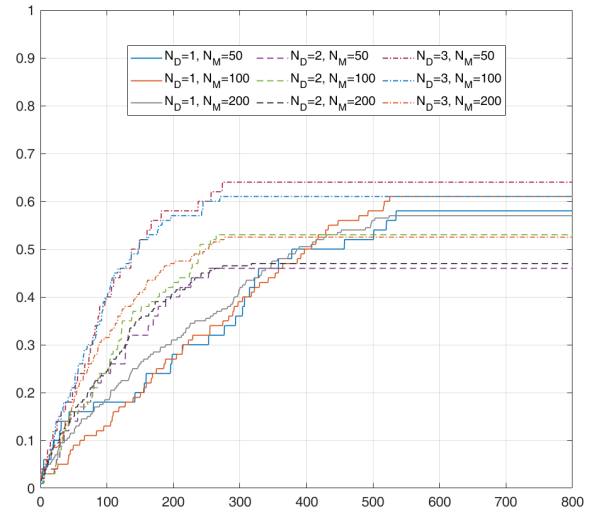


Figure 7: Discovered meshes for different number of drones and meshes.

In the next section, we provide a dynamic version of Problem 1 that leverages information update received from other drones and BOs to update the exploration path in an online fashion.

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

²To prove the NP-hardness, it suffices to show that we can find a polynomial time reduction of the Hamiltonian Cycle problem to an instance of Problem 1.

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 ν 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 $\tilde{\tau}_i(\nu) = \tilde{p}$, $\nu = \nu + 1$, $D_i^R(t) = D_i^R(t) - d(p(\nu - 1), \tilde{p})$, $\tilde{\mathcal{P}}_{b,i} = \tilde{\mathcal{P}}_{b,i} \setminus \tilde{p}$ and we restart from step ii). Otherwise, we force the drone to go back to the corresponding BO, i.e., $\tilde{\tau}_i(\nu) = p_b$.

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( $p_0, \mathcal{P}, D^R, p_{BO}$ )
3:    $\nu \leftarrow 1$ 
4:    $\tilde{\tau} \leftarrow \{p_0\}$ 
5:   while  $D^R \geq d(\tilde{\tau}(\nu), p_{BO})$  and  $\tilde{\mathcal{P}} \neq \emptyset$  do
6:      $\tilde{p} \leftarrow \arg \min_{p \in \tilde{\mathcal{P}}} d(\tilde{\tau}(\nu), p)$ 
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  $\tilde{\tau}$ 

```

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 6.

C. 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 $\delta = \{d \in \mathcal{D} : d \text{ is a deployment policy}\}$, i.e., a set of deployments point where to deploy one or more bridging robots. For any deployment policy δ , let $\mathcal{M}^C(\delta) \subseteq \mathcal{M}$ be set of mesh networks covered by δ . 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 δ 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 maximization problem can be formulated as follows:

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

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

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

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

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

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

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 δ . 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 [15]. Despite its complexity, such a problem has been extensively investigated in the literature and

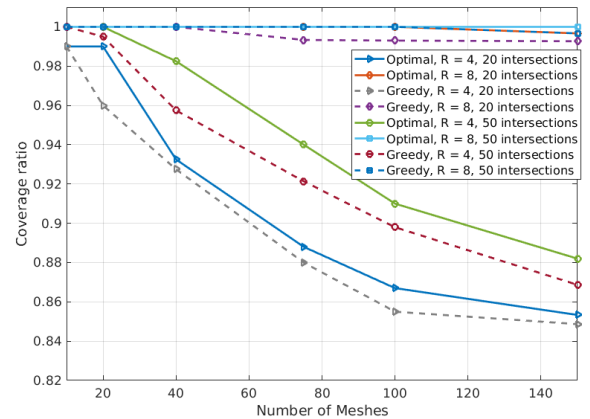


Figure 8: Meshes bridged at varying number of robots R and mesh intersections.

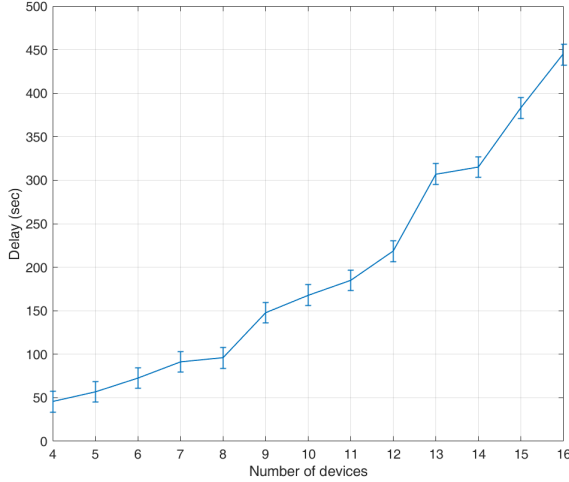


Figure 9: Lower Tier convergence time at varying dimension of the mesh.

many efficient solutions have been proposed by using Branch-and-Bound (BB), Lagrangian relaxation, greedy [15] and approximation [10] algorithms. Figure 8 shows 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).

VII. NETWORK DEPLOYMENT

HIRO-NET currently has a working prototype that has been extensively tested in all its components.

A. 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 9. 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 10 shows the end-to-end delay, intended as a time to deliver a full message between two any nodes in the mesh.

B. Upper Tier

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-Byte with a maximum of 5 total transmissions allowed per end-user per 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 11 shows that Upper Tier packets in

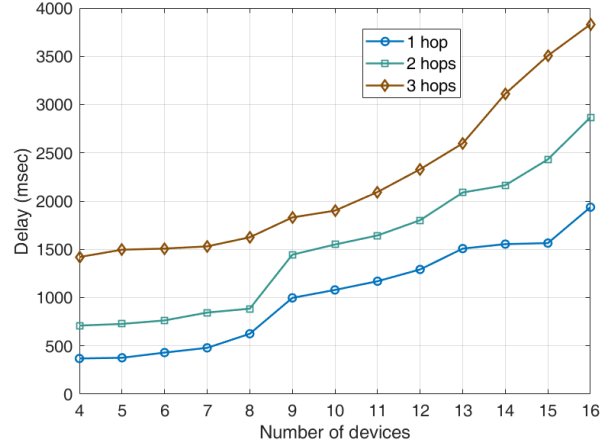


Figure 10: Lower Tier end-to-end delay.

terms of seconds up to 900 meters of distances. After extensive experiments, we observed similar transmission times for a dense urban environment, where 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 environment, 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.

C. 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

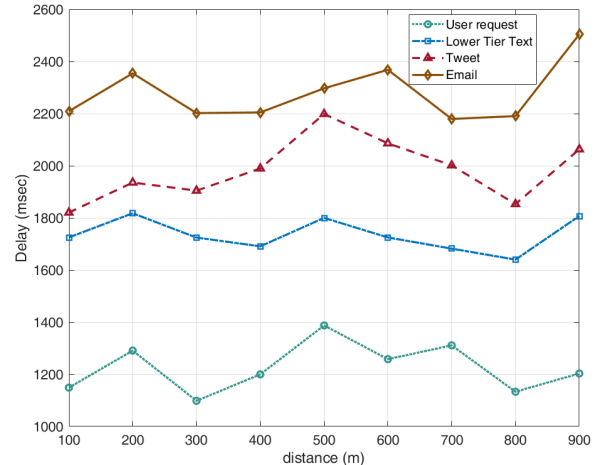


Figure 11: Upper Tier packets transmission delay.

as from Phase II and sends updated to the Flight Controller using the Dronekit library. Experiments were conducted both indoor and outdoor. In Figure 12 a screenshot from the drone control software Qgroundcontrol compares the original flight path and the HIRO-NET enabled flight path.

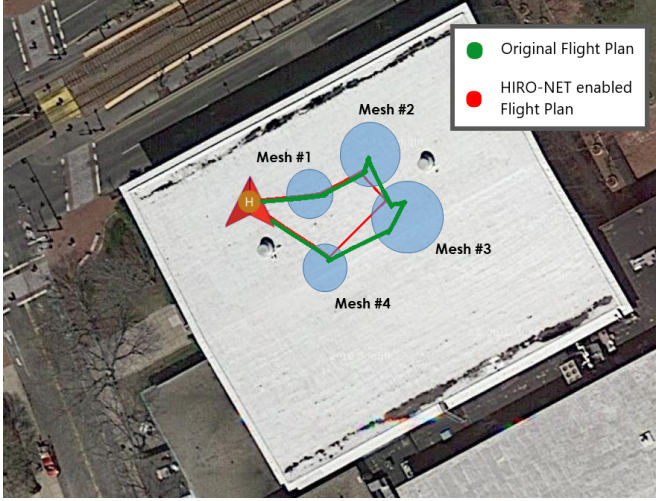


Figure 12: Indoor experiment of the HIRO-NET discovery phase.

All videos of HIRO-NET experiments and Internet sharing functionalities can be appreciated at [26].

VIII. 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.

IX. ACKNOWLEDGEMENTS

This work was supported in part by the NSF grant CNS-1618727. 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] B. Barritt, T. Kichkaylo, K. Mandke, A. Zalzman, and V. Lin. Operating a uav mesh & internet backhaul network using temporospatial sdn. In *Aerospace Conference, 2017 IEEE*, pages 1–7. IEEE, 2017.
- [2] R. Baumann, S. Heimlicher, V. Lenders, and M. May. Heat: Scalable routing in wireless mesh networks using temperature fields. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–9. IEEE, 2007.
- [3] 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*, pages 96–109. ACM, 2006.
- [4] S. Chakraborty, S. Sharma, and S. Nandi. Mac layer channel access and forwarding in a directional multi-interface mesh network. volume 14, pages 565–578. IEEE, 2015.
- [5] 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*, pages 1–7, Sept 2017.
- [6] R. Ding and L. Yang. A reactive geographic routing protocol for wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on*, pages 31–36. IEEE, 2010.
- [7] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy. Dart: Dynamic address routing for scalable ad hoc and mesh networks. volume 15, pages 119–132. IEEE Press, 2007.
- [8] FCC. Fcc communications status report for areas impacted by hurricane irma. https://apps.fcc.gov/edocs_public/attachmatch/DOC-346655A1.pdf, 2017.
- [9] FCC. Fcc communications status report for areas impacted by hurricane maria. https://transition.fcc.gov/Daily_Releases/Daily_Business/2017/db1006/DOC-347118A1.pdf, 2017.
- [10] 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)*, volume 14, pages 601–612. LIPIcs, 2012.
- [11] 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*, pages 4823–4827. IEEE, 2007.
- [12] GoTeNNa. "gotenna — text and gps on your phone, even without service". <https://www.gotenna.com>, 2017.
- [13] 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*, pages 133–140. IEEE, 2017.
- [14] H. Huang, P. Li, S. Guo, and W. Zhuang. Software-defined wireless mesh networks: architecture and traffic orchestration. volume 29, pages 24–30. IEEE, 2015.
- [15] S. Khuller, A. Moss, J. Naor, et al. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.
- [16] 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*, pages 1799–1804. IEEE, 2010.
- [17] 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 *Proceedings of the third ACM workshop on Airborne networks and communications*, pages 17–22. ACM, 2014.
- [18] 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.
- [19] S. Pieterkosky, A. T. Ziegwied, C. Cavanagh, and L. Thompson. Biv meets asv: Bio-inspired fish drones and autonomous surface vehicles for coral reef monitoring. *OCEANS 2017 – Anchorage*, pages 1–5, 2017.
- [20] ROS. "ros.org — powering the world's robots". <https://www.ros.org>, 2018.
- [21] A. Rovira-Sugranes and A. Razi. Predictive routing for dynamic uav networks. In *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pages 43–47, Oct 2017.
- [22] S. Y. Shahdad, A. Sabahath, and R. Parveez. Architecture, issues and challenges of wireless mesh network. In *Communication and Signal Processing (ICCSP), 2016 International Conference on*, pages 0557–0560. IEEE, 2016.
- [23] B. SIG. Gatt overview. <https://www.bluetooth.com/specifications/gatt/generic-attributes-overview>, 2019.
- [24] 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. volume 2, pages 1328–1335. IEEE, 2017.
- [25] H. Suzuki, Y. Kaneko, K. Mase, S. Yamazaki, and H. Makino. An ad hoc network in the sky, skymesh, for large-scale disaster recovery. In *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, pages 1–5. IEEE, 2006.
- [26] WinesLab. Hiro-net youtube channel. <https://www.youtube.com/channel/UCkEtseZk7SFmGNkk1222igg>, 2019.
- [27] M. Zorzi and R. R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. volume 2, pages 337–348. IEEE, 2003.