Energy-Efficient RL-Based Aerial Network Deployment Testbed for Disaster Areas

Mehmet Ariman, Mertkan Akkoç, Talip Tolga Sarı, Muhammed Raşit Erol, Gökhan Seçinti, and Berk Canberk

***Abstract*—Rapid deployment of wireless devices with 5G and beyond enabled a connected world. However, an immediate demand increase right after a disaster paralyzes network infrastructure temporarily. The continuous flow of information is crucial during disaster times to coordinate rescue operations and identify the survivors.**

Communication infrastructures built for users of disaster areas should satisfy rapid deployment, increased coverage, and availability. Unmanned air vehicles (UAV) provide a potential solution for rapid deployment as they are not affected by traffic jams and physical road damage during a disaster. In addition, ad-hoc WiFi communication allows the generation of broadcast domains within a clear channel which eases one-to-many communications. Moreover, using reinforcement learning (RL) helps reduce the computational cost and increases the accuracy of the NP-hard problem of aerial network deployment.

To this end, a novel flying WiFi ad-hoc network management model is proposed in this paper. The model utilizes deep-Q-learning to maintain quality-of-service (QoS), increase user equipment (UE) coverage, and optimize power efficiency. Furthermore, a testbed is deployed on Istanbul Technical University (ITU) campus to train the developed model. Training results of the model using testbed accumulates over 90% packet delivery ratio as QoS, over 97% coverage for the users in flow tables, and 0.28 KJ/Bit average power consumption.

***Index Terms*—Ad-hoc, aerial network, energy-efficient, MavLink, QoS, SDN.**

I. INTRODUCTION

DEPLOYMENT of 5G and beyond networks in conjunction with connectivity-based mobile services increased the utilization of radio frequency (RF) bands. These innovations create a connected world and introduce problems in disaster cases as every mobile entity acts as a communication hub. The RF environment has no safety lane to allow essential communication to survive during a disaster. The most recent example of such a scenario was observed after an earthquake in Croatia on date March 22, 2020, as depicted in Fig. 1 [1], [2]. These incidents, combined with physical access constraints in disaster areas, make the aerial network (AN) a potential solution to provide temporary infrastructure for maintaining continuity of communication in the disaster areas. The nature of infrastructure within disaster areas is agile, and demand is on a fast-changing trend.

As detailed in the rest of this section, existing work in the literature indicates that AN is still an open topic with its agile nature. Furthermore, it has great potential for handling non-persistent traffic. However, it also introduces new challenges to address. Though current work in literature for AN focus on decoupling the control layer from unmanned air vehicle (UAV), reducing latency, and easing configuration, these works lack addressing physical network resource allocations for UAVs and deployment of the proposed method due to the infeasible assumption that is not compatible with implementation facts. On the contrary, the work proposed within this paper focuses on realizing a testbed for the proposed method. It focuses on physical resource allocations of the network, that is, WiFi channels in this specific case. This paper defines a novel network architecture to provide an AN infrastructure for disaster areas with zero handover delay, quality-of-service (QoS) awareness, and energy efficiency. This new architecture has software defined networking (SDN) in the control plane with a reinforcement learning (RL) powered decision mechanism to enable the deployment of a controller on general-purpose computers.

Network infrastructures utilizing UAVs are suitable for handling backend connections for on-demand, rapidly changing networks. That is why the number of articles focusing on AN is increasing in literature. For example, [3] proposes an architecture that can be constructed quickly and effectively considering the delay performance of the network in disaster scenarios. Furthermore, [4] also highlights the similarities of AN with mobile ad-hoc networks (MANET) and vehicular ad-hoc networks (VANET) and challenges introduced by AN which are 3-D mobility, network convergence time and lack of computing resources for solving complex routing decisions. [5] provides a detailed performance analysis on AN for complex routing algorithms widely used in MANETs and VANETs, highlighting the bottleneck on network convergence. To this end, [4] suggests the utilization of SDN to decouple complex routing decisions from UAV nodes in AN. In addition, [4] focuses on resolving the resilience and fail tolerance in the control domain by introducing Block-chain for distributed controller layer. What is more, [6] also favors the use of SDN for AN deployments as it eases the configuration of the network for the operators by utilizing the network function virtualization (NFV) phenomenon. Both [4] and [6] lack inspection of physical resource allocation that is vital for deploying proposed methods. In addition, [6] indicates that current hardware and software limitations prevent the proposed

Manuscript received March 30, 2022 revised November 13, 2022; approved for publication by Yajun Zhao, Guest Editor, December 12, 2022.

M. Ariman, M. Akkoç, T. Sarı, M. R. Erol, and G. Seçinti are with Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkey, emails: {ariman, akkocm, sarita, erolm15, secinti}@itu.edu.tr.

B. Canberk is with Department of Artificial Intelligence and Data Engineering, Istanbul Technical University, Istanbul, Turkey, and School of Computing, Engineering and The Build Environment, Edinburgh Napier University, UK, email: b.canberk@napier.ac.uk.

M. Ariman is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2022.000057

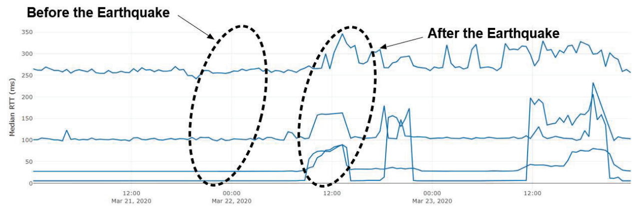


Fig. 1. Median RTT value on Croatia earthquake on March 22, 2020 [1].

method from realization.

In [7], traffic offloading from wireless access points (WAP) using UAV base stations (BS) and cellular communication nodes is inspected to reduce queuing delay for the wireless subscribers (WS). [7] utilizes SDN to exploit global view and address spectrum and channel allocation for the access nodes. [7] exploits the location of UAV BS, spatial spectrum occupancy, and active flow information within the network.

The research on using UAVs in collaborative data collection and processing has also increased in recent years. [8] and [9] utilizes UAVs within data collection. In [8], UAVs are proposed to address the network scarcity problem in the Internet of things (IoT) networks. UAVs are used as relay agents in delay tolerant blockchain to handle secure data storage with help of federated learning that depends on iterative data sharing to prevent the leak of private information. In addition to information transfer in [8], [9] implements the IoT capability on UAVs that serves as surveillance entities.

In [10], a dynamic clustering algorithm has been developed to manage aerial network deployment. Load and battery information of the UAV is utilized by [10] to generate policy output. The policy definition requires a static model for the network parameters such as user mobility rate, traffic demand rate, etc. This limits the solution's applicability to a single environment due to a lack of adaptation.

The most fundamental problems of ANs are UAVs' limited energy and computing capacity. [11] and [12] propose methods that improve energy efficiency and latency by offloading the computing operations on UAVs to the neighboring UAV, edge, or cloud servers, depending on their availability. Effective topology management is also used to increase the availability of the network. In this context, [13] and [14] not only provide energy management by controlling the replenishment and movements of UAVs thanks to their proposed topology management method but also specifically try to improve the quality of video traffic on the network. [15] tries to improve the latency and packet loss performances of the AN by utilizing SDN network technology and controlling the routing tables.

In [16], UAV BS are utilized to increase resource utilization and reduce cost for the infrastructure provided. Moreover, the area coverage problem is attacked from the physical standpoint to define the size of the deployed network. SINR is used as the QoS metric for balancing the optimization problem between the number of UAV BS and the bandwidth perceived by the UEs.

This paper attacks the QoS, service fairness and availability, and energy efficiency by contributing to the literature as follows:

- Definition of opportunistic aerial network infrastructure utilizing ad-hoc WiFi communication.
- Aerial network deployment model using RL to reduce UAV deployment complexity to cover UEs and maintain pre-defined QoS metrics.
- Definition of QoS feedback mechanism using UE coverage and flow density rather than area coverage and signal quality parameters widely used in the existing literature.
- Implementation of aerial network testbed.

II. ARCHITECTURE

This paper defines a network architecture to realize an AN utilizing UAVs for handling rapidly changing networks via centralized control embracing the SDN approach on the control plane. The overall architecture of the proposed infrastructure is depicted in Fig. 2. This work focus on the access network deployed in the disaster area. Furthermore, a 5 GHz backend connection is intended to transfer the data to one of the multiple local backend gateway entities in disaster areas. It is assumed that the local backend gateway entity is equipped with reliable communication interfaces such as satellite, cellular, optical, etc.

The proposed network architecture consists of four different entities. The first architectural entity is a user equipment (UE), the primary user of the infrastructure provided. These entities have a 2.4 GHz WiFi interface capable of maintaining ad-hoc communication. These entities are the source of the traffic generated for the given architecture. The second entity is the communication entity (CommEnt) which is responsible for bridging UE data with the data backend. Moreover, these entities are equipped with 2.4 GHz WiFi, 5 GHz WiFi, and a USB interface. WiFi is a standard communication interface on phones and tablets and allows disaster victims to communicate through UAVs using opportunistic channels. The third entity is the flight management unit (FMU), which runs an autopilot, maintains a 433 MHz RF control link, and provides telemetry data for the control plane decisions. Long-range communication interface for low bandwidth communication traffic for the ground control station of the UAV. 433 MHz is picked for long-range communication as it resides in industrial scientific medical (ISM) band that does not require any license for communication. This frequency band would be 868 MHz in the US. The last entity is the ground control station (GCS). This entity is the workhorse of the control plane, and its details are given in Section II-A.

A centralized SDN controller orchestrates the network utilizing the information retrieved from the UAVs. The information provided by the UAV is constructed by a communication entity placed on top of it. The information the communication entity provides is consumed by the SDN controller placed in GCS to generate control decisions of the network. SDN controller requires WiFi channel status from the UAVs to select the WiFi channel for UEs and UAVs to communicate in the next time slot. This information consists of each channel's received signal strength to interference (RSSI) on the 2.4 GHz

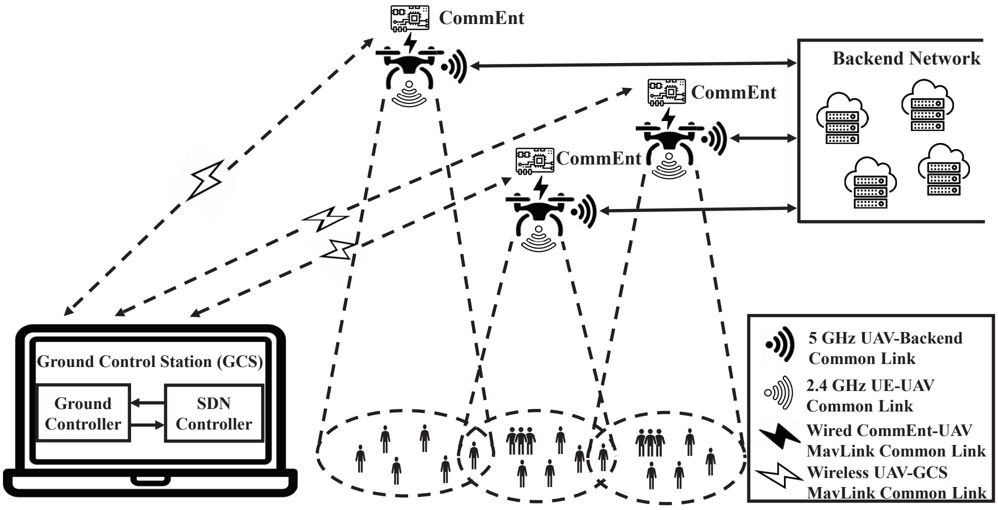


Fig. 2. Opportunistic aerial network service model.

baseband. Furthermore, an active flow list is also sent from UAV which is the list of flows within the flow table that has at least a single transaction since the last query of the SDN controller. This information is crucial for the SDN controller to determine the number of active flows within the system. In addition, battery information is also provided by UAVs to the SDN controller so that the SDN controller is capable of mitigating low-battery UAVs and capable of optimizing the load on the UAVs. Last, location information is also provided by UAVs so that the SDN controller is aware of the actual position of the UAVs. The SDN controller optimizes the number of active UAVs analogous to the set-cover problem by executing policies for handovers, load-balancing, and swapping the low-battery UAVs as needed using RL. The policies executed by the SDN controller provide scalable power consumption performance while maintaining SLA levels for the UE coverage and the QoS. The link between the UAV and the GCS is sustained through the 433 MHz MavLink interface through the telemetry module.

UAVs maintain the communication of the UEs in the novel agile infrastructure defined within the context of this paper through a 2.4 GHz ad-hoc WiFi link. In addition, UAVs bridge the data plane to the backend through a 5 GHz infrastructure WiFi link. The control plane of the UAV nodes is connected to the GCS via a 433 MHz MavLink connection, as explained earlier in this section. FMU placed on the UAV is responsible for handling auto-pilot tasks and gathering sensor information. In addition, FMU also handles MavLink messaging interfaces. Furthermore, FMU also feeds location and battery data to the communication entity as needed.

The information required by the SDN controller and generated by UAV in this architecture is facilitated by utilizing a reserved message type in the "Message Type" field of the MavLink protocol, which is illustrated in Fig. 3. The AN-specific protocol handling for passing the information required by the SDN controller is realized using the message protocol

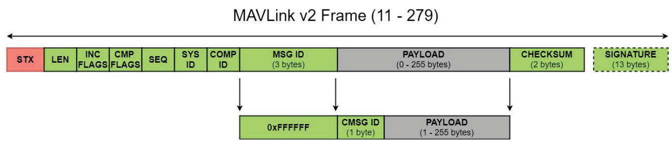


Fig. 3. MAVLink message format [17].

explained in Table I. The message ID used within standard MavLink is 0xFFFFFF.

A. *Ground Control Station*

GCS consists of Ground Controller and SDN Controller, as depicted in Fig. 4.

1) Ground controller: Ground Controller is responsible for bridging inter-UAV and SDN controller messages. Furthermore, it has direct access to the 433 MHz telemetry module connected to the controller via USB. MavLink messages received from UAV are encapsulated into a TCP message and transmitted to the SDN controller for the decision mechanism to operate.

Ground Controller also collects the control messages generated by the SDN controller and delivers them to the UAVs for effective orchestration of the network.

2) SDN controller: SDN controller is responsible for processing the information received from the UAVs through MavLink through Ground Controller. The information received is processed via an RL engine for each UAV agent to decide on the following action if needed. This action is generated depending on the battery, location, flow table load, and the number of users assigned to the UAV and its closest neighbor. The cost induced by RL occurs only in the training phase to better adapt the weights for the inference. Once weights are fixed, the computational cost of running inference is constant. In addition, the rationale behind utilizing RL is to reduce

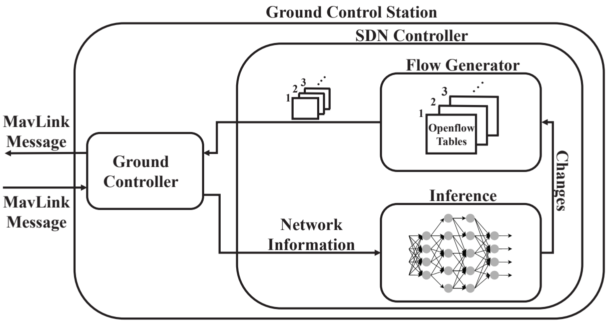


Fig. 4. Network control modules in ground control station.

the computational cost of the problem at hand. The problem is deploying a minimum number of UAVs to ensure cell coverage and maintaining agreed QoS metrics. This problem is proved to be NP-hard and is detailly explained in [16]. The rationale behind including neighbor status in input parameters is to allow offloading of the current traffic to the neighbor and reduce overall power consumption when inference result provisions acceptable QoS even when the number of active UAVs is reduced.

QoS element of the constraint list is ignored to prove the NP-hardness of the problem at hand using (1). In (1), ω_i is the status of the UAV, $\mathcal{F}$ is the combined set of flow tables for all UAVs, $\mathcal{F}_i$ is the flow table of UAV_i, A is the set of active UAVs, C is the set of charging UAVs, $\mathcal{U}$ is the list of all UEs, id_i is the rule of UE_i, and N_{UAV} is the number of all UAVs. Optimization problem (1) minimizes the number of active flow tables, indirectly active UAVs, to cover all UEs by having at least one id from each that exists in $\mathcal{U}$. This problem minimizes the number of active UAVs to cover all UEs in flow tables and is analogous to the known set-cover optimization problem that is NP-hard.

$$\begin{aligned}
&\text{minimize} \sum_{i=1}^{N_{UAV}} \omega_i \mathcal{F}_i \\
&\qquad \omega_i : \begin{cases} 1, & UAV_i \in A \\ 0, & UAV_i \in C \end{cases} \\
&\text{subject to} \\
&\qquad id_i \in \mathcal{F}_i \geq 1 \quad \forall \quad id_i \in \mathcal{U} \\
&\qquad \mathcal{F}_i \leq 1 \qquad\quad \forall \quad i \in \{1, 2, 3, ..., N_{UAV}\} \\
&\qquad \mathcal{F}_i \in \mathcal{F}
\end{aligned} \tag{1}$$

In summary, the deep-Q-network (DQN) of the SDN controller that runs inference for each UAV agent is constructed as a multi-layer perceptron (MLP) with four layers. These layers are named input layer, internal layer one, internal layer two, and output layer. The input layer (IL) uses a battery, number of active flows, number of active users, position X, position Y, position Z, neighbor's battery, neighbor's number of active flows, neighbor's number of active users, neighbor's position X, neighbor's position Y, neighbor's position Z, state

TABLE I
AN MESSAGE TYPES.

<table>
<tr><th>AN message ID (1 byte)</th><th>Content</th><th>Direction* (1 byte)</th><th colspan="3">Message payload (254 bytes)</th></tr>
<tr><td>0</td><td>Flow install</td><td>0</td><td colspan="3">OpenFlow payload</td></tr>
<tr><td>1</td><td>Flow remove</td><td>0</td><td colspan="3">OpenFlow payload</td></tr>
<tr><td>2</td><td>Channel status</td><td>0</td><td colspan="2">Active (0) /Deactive (1) flag (1 byte)</td><td>Unused</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">RSSI values</td><td rowspan="2">1</td><td colspan="2">RSSI pair for each channel</td><td>2 bytes × 126 pairs</td></tr>
<tr><td>Channel ID (1 byte)</td><td>RSSI values (1 byte)</td><td>...</td></tr>
<tr><td>4</td><td>Active user number</td><td>1</td><td colspan="2">Unsigned int (2 bytes)</td><td>Unused</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Active flow number</td><td rowspan="2">1</td><td colspan="2">Flow pair for each row</td><td>2 bytes × 126 pairs</td></tr>
<tr><td>Row ID (1 byte)</td><td>Flow ID (1 byte)</td><td>...</td></tr>
</table>

* 0: GCS to UAV, 1: UAV to GCS

of each UAV in the system. The size of the input layer accumulates to (12+ max number of UAVs). This will be referenced as size(IL) in the rest of this paper. The output layer stores the probabilities of each possible outcome: Handover, callback, and hold for each UAV. The overall structure of MLP is size(IL) (input) $\times$ (size(IL))^2 $\times$ (size(IL))^3 $\times$ 3 (output) layers.

The aforementioned MLP inference is run for each UAV, and the result of inference is used as the RL counterpart of ω_i in (1). The RL model includes battery level and the number of users by activating them in the input layer in addition to the core set-cover problem given in (1). The training session is required for the RL model to run inference before deployment. The QoS in (2), UE coverage, and power consumption factor in (4) values used in the reward function given in (3) are used in the feedback loop to define weights for the MLP connections during the training cycle.

The Algorithm 1 performs inference and generates RL counterpart of ω_i in (1) in the first for loop. The output of each round is fed as a reward function given in (3) to generate a fitness function for the training.

During training, the rewarding mechanism is constructed using the metrics given below:

- **QoS**: This item represents the ratio of the packets delivered from source to sink successfully. This parameter depends on the availability of the flow table as well as the coverage of the UE by the UAV. The pre-defined service level agreement (SLA) to achieve 90% and above. The

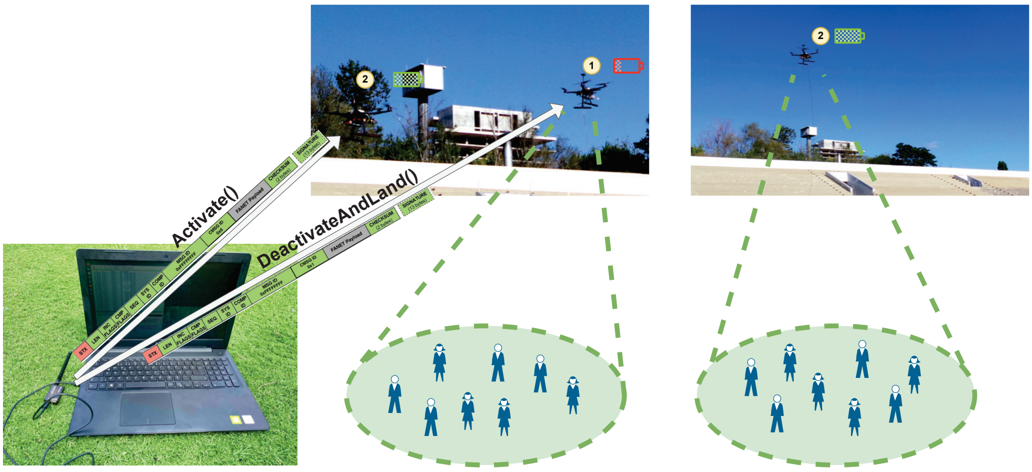


Fig. 5. Handover mechanism.

QoS is the second most significant item in the reward equation. The ratio is normalized to 100 scales in (2) and multiplied by 1.3 in (3). The weight of QoS in the reward equation is derived empirically with respect to coverage and power consumption. This parameter appears as QoS in the reward equation as well. The computation of QoS is explained in (2). The parameters of (2), N_{UE} is the number of UEs, TG_i is the traffic generated by UE_i, $\mathcal{L}$ is the size of the flow table, $\mathcal{S}_{\mathcal{R}_i}$ is the packet statistics of rule i in the flow table, and N_{UAV} is the number of UAVs.

$$QoS^* = 100 \times \frac{\sum_{i=1}^{N_{UAV}} (\sum_{j=1}^{\mathcal{L}} \mathcal{S}_{\mathcal{R}_j})}{\sum_{i=1}^{N_{UE}} TG_i} \quad (2)$$

- **Power consumption**: This is the power consumed for the whole network and measured in KJ/bit. This factor will appear as Pow in the power consumption factor equation. The reward mechanism punishes any rise in power consumption, while drops have been rewarded. Power consumption factors for various intervals are utilized to generate reward functions. These factors are derived by experiment and given in (4).
- **Coverage**: Coverage is defined as the number of users that has at least one flow installed on the UAVs and are capable of running successful transactions. As connectivity for UEs is crucial in disaster areas, this element has the highest impact on the reward equation. This appears as Cov in the reward equation.

The reward equation used for training the network is given below:

$$Reward = (QoS \times 1.3) + (Cov \times 1.5) + PowConFactor, \quad (3)$$

$$PowConFactor = \begin{cases} 110, & \text{if } 0.25 > Pow > 0 \\ 60, & \text{if } 0.30 > Pow \geq 0.25 \\ 40, & \text{if } 0.35 > Pow \geq 0.3 \\ 15, & \text{if } 0.45 > Pow \geq 0.35 \\ 10, & \text{if } 0.60 > Pow \geq 0.45 \\ 1, & \text{if } 1.00 > Pow \geq 0.6 \\ 1/Pow, & \text{otherwise.} \end{cases} \quad (4)$$

The values given in the reward equation and power consumption factor are derived empirically from tests conducted using the aerial testbed developed.

III. Use Cases

We demonstrate two use cases for given network architecture using the new infrastructure. The first is the handover mechanism that happens once an active UAV is about to drain its battery and requires replacement. We give details about this use case in Section III-A. The following use case is new cell deployment which happens when a new flow or user is added to the network. While this happens, the SDN controller extends the network with new UAVs. We discuss this mechanism in Section III-B.

A. Handover Mechanism

In this section, we describe the handover mechanism within the proposed network architecture in detail. The handover mechanism is changing the UAV that provides network access to UE to another UAV. Using our testbed implementation; we illustrate the handover process in Fig. 5. Additionally, Fig. 6, uses a sequence diagram to show this handover process in more detail. The GCS collects the UAV info: Location, load information of the flow table, UE coverage, and battery information. Then the inference is run to decide the action for the current state. Suppose a swap action is triggered for a particular UAV. In that case, an exact copy with the same configuration in terms of flow table is prepared and deployed to the nearby location. Furthermore, once a duplicate UAV reaches its position activation signal is generated for the duplicate, and the original UAV is deactivated to complete the sequence.

SDN controller uses custom MavLink messages to control UAV's behavior and monitor UAV flight data. Our handover process embraces a handover mechanism that does not disrupt communication during the handover period. SDN controller uses a neural network (NN) to make various decisions using the mission data of the UAVs, such as battery level and traffic demand of UEs on the site. If a UAV's battery level is low, the SDN controller initiates the handover process to keep

Algorithm 1 Action generation and execution

```
Input: UAVs
Decisions = new List < Action > ()
for UAV in UAVs do
    Decisions(UAV) = AN.runUavInference(
    Battery,
    NumOfActiveFlows,
    NumOfActiveUsers,
    Location.X,
    Location.Y,
    Location.Z,
    Neighbor.Battery,
    Neighbor.NumOfActiveFlows,
    Neighbor.NumOfActiveUsers,
    Neighbor.Location.X,
    Neighbor.Location.Y,
    Neighbor.Location.Z,
    Foreach UAV in UAVs UAV[UavIdx].IsActive)
end for
for Decision in Decisions do
    if Handover then
        Dup = SDN.Duplicate(UAV)
        SDN.SendAndWait(Dup)
        SDN.Activate(Dup)
        SDN.DeactivateAndLand(UAV)
    else if Callback then
        SDN.DistributeFlows(UAV)
        SDN.DeactivateAndLand(UAV)
    else if Hold then
        Donothing
    end if
end for
WiFiChannel = AN.SelectChannel(UAVs.RssiList)
SDN.DistributeChannel(Channel)
```

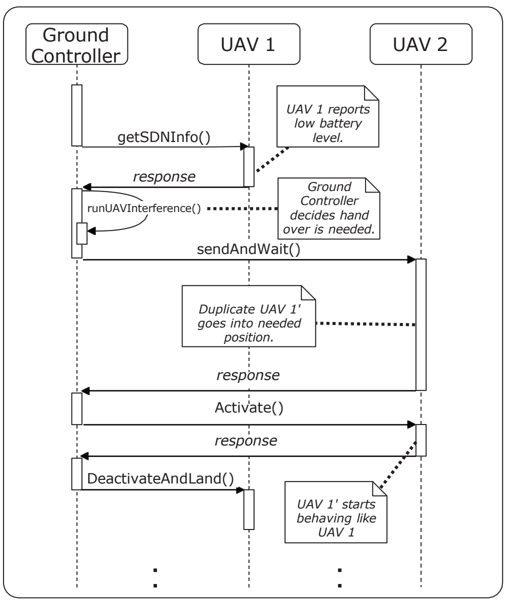


Fig. 6. Sample handover mechanism process with low battery.

providing service to existing UEs. Once the SDN controller decides to switch the active UAV, it prepares the replica of the active UAV with pre-installed flow rules and sends it to the 2-meter perimeter of the active UAV. When the replica reaches its mission position, as depicted in Fig. 5, the GCS sends an activation message to the replica UAV to start the handover. Then, GCS sends a deactivation message to the active UAV with a low battery to stop duplicate messages from circulating in the network. Finally, a deactivated UAV is commanded to return home to complete the handover with the handover mechanism for re-charging purposes.

B. New Cell Deployment

The following use case we inspected is when the demand increases around certain spots in the network. When this happens, GCS initiates a new cell deployment process. As briefly explained in Section II-A2, GCS keeps track of the active flows, active UEs, UAV nodes with their battery status and locations, WiFi channel status, and QoS. Using this data, Algorithm 1 makes decisions throughout the network. GCS makes these decisions by using its deep RL-based controller. Reduced QoS, increased demand, and the addition of new users are capable of generating a new cell deployment action. The result of such action with the addition of new UE is demonstrated in Figs. 7(a) and 7(b). Additionally, if the demand decreases, the GCS can call back UAVs similarly. Combination of these yields, energy-efficient operation of UAV network, and better QoS.

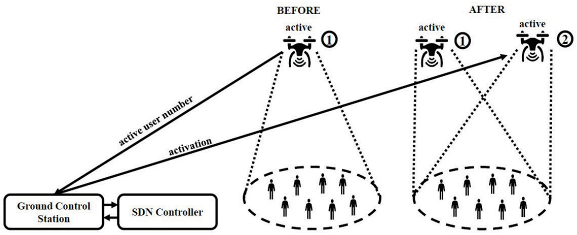
IV. PERFORMANCE EVALUATION

This section contains the performance evaluation of the proposed model and a walk-through of the results to verify the validity of the solution.

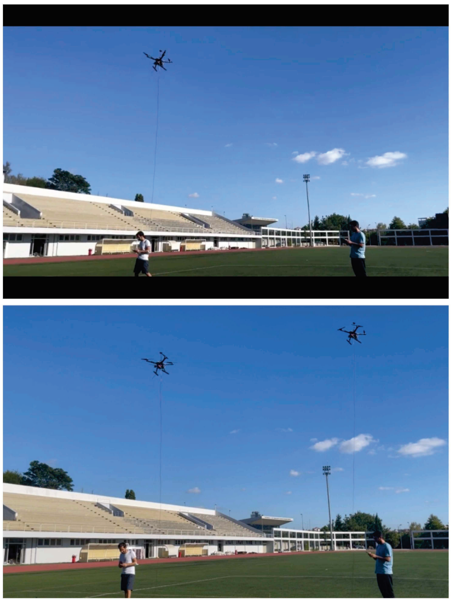
A. Test-bed Setup

Our testbed consists of 4 NXP HoverGames [18] drones equipped with three cell Lithium Polymer (LiPo) batteries that can provide up to 25 minutes of flight time depending on the air conditions. Also, each drone has Raspberry Pi 4 Computer Model B [19] with 16 GB Micro SD card as seen in Fig. 8. We employ Ubuntu 21.04 as an operating system and utilize Open vSwitch 2.15.0 [20] on each Raspberry Pi to create a data plane. Raspberry Pi is powerful, supports WiFi, has 4 GB memory, and a Quad-Core 1.5 GHz ARM-v8 processor to efficiently run an inference with MLP. It runs in battery-powered environments without performance drawbacks. Additionally, we use MAVSDK Library [21] to each Raspberry Pi to create an interface and use MavLink protocol [22] between the Raspberry Pi and the FMU of a drone, which has FMUK66-based PixHawk autopilot firmware [23]. We also use the MavLink protocol between the GCS and the drone. Furthermore, we use Packet Sender [24] software to create 3 to 25 flows from each ground Raspberry Pi to drone Raspberry Pi. We classify these flows as TCP, UDP, and VoIP, which are randomly distributed.

GCS consists of a general-purpose laptop with Intel(R)



(a) In theory



(b) In action

Fig. 7. New cell deployment scenario.

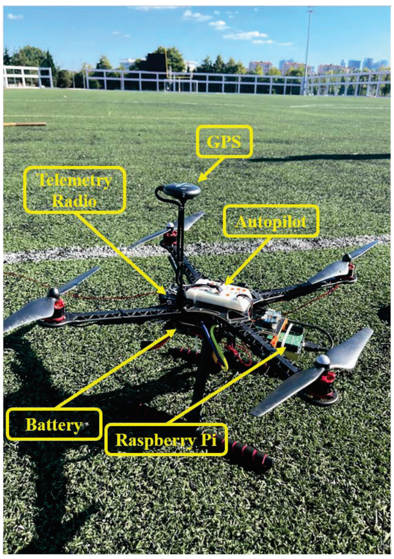


Fig. 8. Test-bed setup on the drone.

Core (TM) i7-1065G7 CPU, 16 GB RAM, and Windows 10 Pro operating system. Furthermore, two virtual machines are deployed on GCS, one of which acts as an Air-Ground bridge while the other one handles the SDN Controller role. Virtual

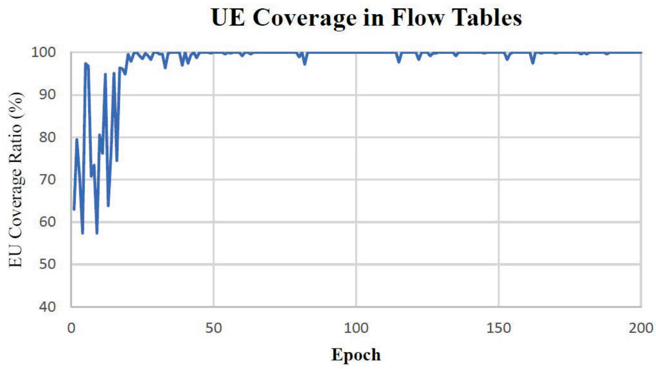


Fig. 9. UE Coverage trend in testbed-based training of ML model.

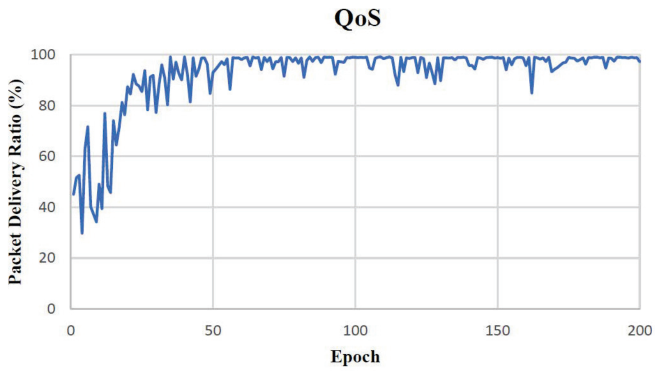


Fig. 10. QoS, packet delivery ratio, the trend in testbed-based training of ML model.

machines are equipped with Ubuntu 20.04.2 LTS operating system. The QGroundControl is installed on the Air-Ground bridge to exchange information with UAVs using the MavLink protocol on GCS. In addition, Open network operating system (ONOS) [25] is installed on the SDN Controller entity virtual machine to handle control plane activities in the architecture given in Fig. 2 and explained in Section II.

This testbed is utilized for training the model proposed in Section II-A2. During the training loop, each episode is divided into 128 decision cycles in which the inference of DQN generates one of the three decisions available for each UAV; hold, handover, or callback, as explained in Algorithm 1. The result of the decision on the testbed is fed back to the model for tracing the training trend to observe the model's validity. Furthermore, the status of the topology and the link utilization is monitored through ONOS. The result of the experiment is presented in Section IV-B.

B. Performance Results

Performance results presented in this section prove that the proposed model is valid as DQN intends to maximize the UE coverage as prioritized by the reward function. The rationale behind this is to provide a fair share of the flow table for each UE during disaster times. The proposed model converges to full coverage within 40 episodes, as depicted in Fig. 9. Fig. 9 corresponds to the analogous of $id_i \in \mathcal{F}_i$ in (1). Then the model works on optimizing QoS to reach 90% packet delivery

TABLE II
PERFORMANCE COMPARISON.

Method	Coverage	QoS	Power
Proposed method	97%	$\geq$90%	0.28 KJ/bit
[16]	98%	$\geq$70%	0.42 KJ/bit

ratio as seen in Fig. 10. The QoS convergence is later than UE coverage as the reward factor for coverage is higher within (3). The proposed model can reach a packet delivery ratio higher than 90% within 80 episodes. Figs. 9, 10, and 11 RL model first explores the search space for an optimal solution. This results in low reward and less performance for the warm-up period. Then the model exploits its discoveries and ramps up the reward and performance in later episodes. However, there are still oscillations in reward and performance, even in late episodes. This is expected and proves that the training model applies random walks to prevent over-fitting and expands the search space and knowledge instead of getting stuck in local peaks. The model then achieves stable power consumption output measured in KJ/bit for maintaining a scalable solution in terms of data processed and UE covered. As depicted in Fig. 12, the proposed model is capable of generating stable per-bit consumption as opposed to [3], which generates power consumption result, which highly varies depending on data processed by the network. Though the packets delivered within the network and the UEs covered increase through time, the model keeps the power consumption within the target range, which is 0.28 KJ/Bit, as depicted in Fig. 12.

The learning trend and capability of the model with our testbed in action are depicted in Fig. 11 in which the model reaches 94% of the maximum reward available through time.

Finally, the proposed method is compared with [16]. The comparison results are represented in Table II. [16] tries to increase the area coverage for serving the UEs. To this end, the power consumption of [16] is slightly higher than the method proposed within the context of this paper. Though [16] provides good SINR for the UE in terms of coverage, it does not increase the intensity of the UAVs in areas where UEs are denser. Moreover, it fails to provide high QoS compared to the proposed method. In terms of coverage, the proposed method demonstrates comparable performance compared to [16]. The network lifetime of the [16] tends to be lower over time as it consumes more energy and tends to keep more UAVs in the air.

V. CONCLUSIONS

In this work, we have presented a novel infrastructure for disaster areas based on SDN and orchestrated the underlying aerial network to maximize UE coverage and keep QoS at the target level while optimizing the energy efficiency measured in KJ/bit. As opposed to [3], the proposed method provides a scalable power consumption scheme as it stabilizes per bit consumption for a different amount of data processed.

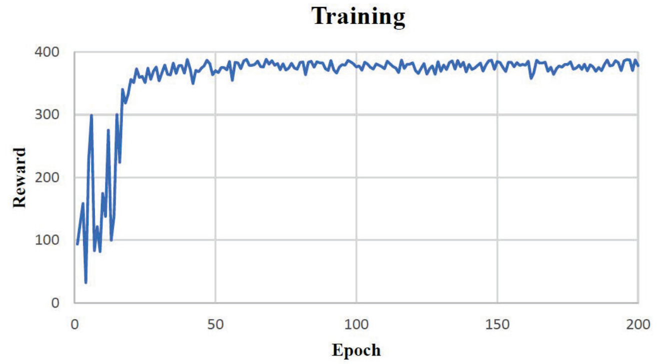


Fig. 11. Reward trend of testbed-based training of ML model.

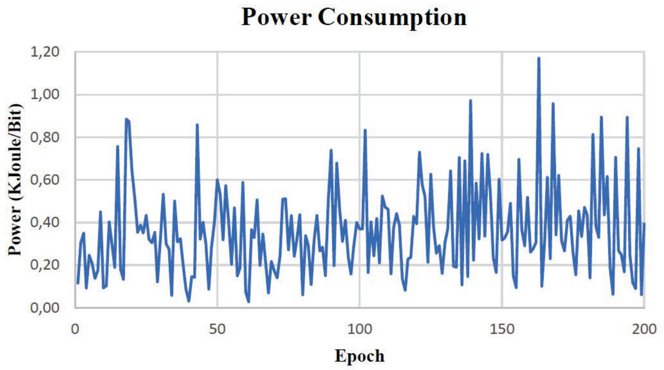


Fig. 12. Power consumption trend in testbed-based training of ML model.

ACKNOWLEDGEMENTS

The paper is supported by the Scientific Technical Research Council of Turkey (TUBITAK) 1001 The Scientific and Technological Research Projects Funding Program, with project number 119E434.

REFERENCES

[1] "Internet health report." https://ihr.iijlab.net (accessed Aug. 31, 2021).
[2] "Ripe labs, after the quake in croatia." https://labs.ripe.net/author/becha/after-the-quake-in-croatia/ (accessed Aug. 31, 2021).
[3] Z. Kaleem *et al.*, "UAV-empowered disaster-resilient edge architecture for delay-sensitive communication," *IEEE Netw.*, vol. 33, no. 6, pp. 124–132, 2019.
[4] N. Hu *et al.*, "Building agile and resilient UAV networks based on SDN and blockchain," *IEEE Netw.*, vol. 35, no. 1, pp. 57–63, 2021.
[5] S. Rosati, K. Kruželecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic routing for flying ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1690–1700, 2016.
[6] B. Nogales, V. Sanchez-Aguero, I. Vidal, and F. Valera, "Adaptable and automated small UAV deployments via virtualization," *Sensors*, vol. 18, no. 12, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/12/4116
[7] M. A. Ali, Y. Zeng, and A. Jamalipour, "Software-defined coexisting UAV and wifi: Delay-oriented traffic offloading and UAV placement," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 988–998, 2020.
[8] A. Islam, A. Al Amin, and S. Y. Shin, "FBI: A federated learning-based blockchain-embedded data accumulation scheme using drones for Internet of things," *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 972–976, 2022.

[9] A. Islam, T. Rahim, M. Masuduzzaman, and S. Y. Shin, "A blockchain-based artificial intelligence-empowered contagious pandemic situation supervision scheme using Internet of drone things," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 166–173, 2021.

[10] G. Asaamoning, P. Mendes, and N. Magaia, "A dynamic clustering mechanism with load-balancing for flying ad hoc networks," *IEEE Access*, vol. 9, pp. 158574–158586, 2021.

[11] R. M. Shukla, S. Sengupta, and A. N. Patra, "Software-defined network based resource allocation in distributed servers for unmanned aerial vehicles," in *Proc. IEEE CCWC*, 2018.

[12] A. A. Ashraf Ateya, A. Muthanna, R. Kirichek, M. Hammoudeh, and A. Koucheryavy, "Energy- and latency-aware hybrid offloading algorithm for UAVs," *IEEE Access*, vol. 7, pp. 37587–37600, 2019.

[13] P. Cumino *et al.*, "Cooperative UAV scheme for enhancing video transmission and global network energy efficiency," *Sensors*, vol. 18, no. 12, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/12/4155

[14] Z. Zhao *et al.*, "Software-defined unmanned aerial vehicles networking for video dissemination services," *Ad Hoc Netw.*, vol. 83, pp. 68–77, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870518306231

[15] T. Dapper e Silva *et al.*, "Stfanet: SDN-based topology management for flying ad hoc network," *IEEE Access*, vol. 7, pp. 173499–173514, 2019.

[16] H. Huang, C. Huang, and D. Ma, "A method for deploying the minimal number of UAV base stations in cellular networks," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 559–567, 2020.

[17] D. Project, "Mavlink packet serialization." https://mavlink.io/en/guide/serialization.html (accessed Aug. 19, 2021).

[18] "Nxp hovergames." https://nxp.gitbook.io/hovergames/ (accessed Aug. 31, 2021).

[19] "Raspberry pi 4." https://www.raspberrypi.org/ (accessed Aug.31, 2021).

[20] "Open vswitch." https://www.openvswitch.org/ (accessed Aug. 31, 2021).

[21] "Mavsdk library." https://mavsdk.mavlink.io/main/en/index.html (accessed Aug. 31, 2021).

[22] "Mavlink protocol." https://mavlink.io/en/ (accessed Aug. 31, 2021).

[23] "Fmuk66 based pixhawk autopilot firmware." https://github.com/PX4/PX4-Autopilot (accessed Aug. 31, 2021).

[24] "Packet sender." https://packetsender.com/ (accessed Sep. 6, 2021).

[25] "Open network operating system." https://opennetworking.org/onos/ (accessed Aug. 31, 2021).

Mehmet Ariman is a PhD candidate within the Department of Computer Engineering at Istanbul Technical University. He holds a BSc and MSc within Computer Engineering from the same university. His research focus is software defined networking. He has studied wireless access management during his MSc and extended his work to incorporate Ad-Hoc networks such as UAVs during his PhD. Mehmet acts as Reviewer for Elsevier Computer Networks, Elsevier Computer Communications. In addition, Mehmet works on Security on his professional life and works at Analog Devices as Senior Manager, SW Systems Design Engineering and leads a mid-size engineering team to research and tackle state-of-the-art and next generation security problems. The topics studied by Mehmet includes, but not limited to side-channel-attack, post-quantum cryptography, lightweight cryptography and efficient implementation of crypto algorithms using hardware/software co-operative solutions.

Mertkan Akkoç received his B.Sc. in Electronics and Communication Engineering and M.Sc. in Computer Engineering from Istanbul Technical University (ITU), Turkey, in 2016 and 2020, respectively. He is currently a Ph.D. student in the Computer Engineering Program of ITU. His research areas include software-defined networking (SDN), data plane programming, intelligent aerial networks, and network management systems.

T. Tolga Sarı is a Research Assistant and Ph.D. student in the Computer Engineering Department at Istanbul Technical University. He received his MSc ITU in June 2021. He received best paper award in 6th IEEE International Workshop on Wireless Communications and Networking in Extreme Environments (WCNEE) and 2022 ComSoc Turkey Master Thesis Competition Special Achievement Award. His research interests are next generation ad-hoc networks and MAC protocols.

Muhammed Raşit Erol received her BSc degree in computer engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 2020. He is currently an MSc candidate in the Computer Engineering Program of ITU. He interned in Baykar Makina from August 2018 to September 2018, where he developed a project with C# and .NET framework. After that, he interned at ASELSAN, Turkey, in 2019. During this internship, he developed software for the Avionics Systems Design Division using C#. In 2019, he also interned at Doğuş Teknoloji, Turkey. In Doğuş Teknoloji, he worked in the Network & Security System Department, and he was able to learn how to manage a company network. He has worked as a Backend Developer at Hitit Computer Services since July 2022. His current research areas include AI-enabled Digital Twins, IoT Communication, and Smart Wireless Networks.

Gökhan Seçinti (M'17) is an Assistant Professor in the Computer Engineering Department at İstanbul Technical University (ITU). He worked as a Postdoctoral Research Associate in Next Generation Networks and Systems (GENESYS) Laboratory at Northeastern University between Nov. 2017 - April 2019. He is the recipient of the distinguished PhD thesis award in 2018 from ITU. He also received best paper award from the 21st IEEE International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (IEEE CAMAD) in 2016, and best poster paper award from IEEE International Conference on Computer Communication (IEEE INFOCOM) in 2015. He currently serves as a Reviewer in several IEEE journals including IEEE Transactions on Wireless Communication, IEEE Transactions on Vehicular Systems and IEEE Internet of Things Journal. His research interests include Software-defined Networks, Ad-hoc Communication Systems and Aerial Networks.

Berk Canberk (Senior Member of IEEE) is a professor within the School of Computing, Engineering and The Build Environment, Edinburgh Napier University, and he is also a professor within the Department of Artificial Intelligence and Data Engineering, Istanbul Technical University. He serves as an Editor in IEEE Transactions on Vehicular Technology, Elsevier Computer Networks and Elsevier Computer Communications. He is the recipient of IEEE Turkey Research Incentive Award (2018), IEEE INFOCOM Best Paper Award (2018), The British Council (UK) Researcher Link Award (2017), IEEE CAMAD Best Paper Award (2016), Royal Society International Exchanges (UK) (2021), Royal Academy of Engineering (UK) NEWTON Research Collaboration Award (2015), IEEE INFOCOM Best Poster Paper Award (2015), ITU Successful Faculty Member Award (2015) and Turkish Telecom Collaborative Research Award (2013). His current research areas are AI- enabled Digital Twins, IoT Communication, and Smart Wireless Networks.