Robust Multi-Path Communications
for UAVs in the Urban IoT

Zoheb Shaikh, Sabur Baidya and Marco Levorato

Computer Science Department, UC Irvine, CA, US

Email: {zsshaikh,sbaidya,levorato}@uci.edu

Abstract—Unmanned Aerial Vehicle (UAV) systems are being increasingly used in a broad range of scenarios and applications. However, their deployment in urban areas poses important technical challenges. One of the most prominent concerns is the robustness of communications between the ground stations and the UAVs in a highly dynamic and crowded spectrum. Indeed, competing data streams may create local or temporary congestion impairing the ground stations to control the UAVs. The main contribution of this paper is a robust multi-path communication framework for UAV systems. The framework continuously probes the performance of multiple wireless multi-hop paths from the ground stations to each UAV, and dynamically selects the path providing the best performance to support timely control. Numerical results, based on a real-world implementation and extensive field experimentation, demonstrate the ability of the proposed framework to provide robust control against exogenous interference and network congestion.

Index Terms—Unmanned Aerial Vehicles, Urban Internet of Things, Congestion Control, Network Selection, Multi-path routing.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAV) are being increasingly used in a broad spectrum of scenarios and applications [1]. Their integration in the Urban Internet of Things (IoT) is attracting a considerable interest, for instance to enhance the ability of the city-wide system to support delicate tasks such as surveillance and monitoring, Virtual Reality, and disaster management, and improve or maintain network coverage.

UAVs are typically controlled by a Ground Control Station (GCS), which wirelessly interconnects with the UAV to build a data-control loop composed of an upstream flow of control messages and a downstream flow of telemetry and sensor data. The urban environment poses several challenges undermining the ability of the GCS to control the UAVs. First, the topological characteristics of the urban environment may severely limit the operating range due to Line of Sight (LoS) obstruction. This issue has been partially addressed in prior work by creating mesh networks of cooperating UAVs. However, another important issue that remains largely unaddressed is the coexistence of UAV-related traffic with competing IoT data streams. Exogenous traffic sharing the same access and/or backbone network may create localized and temporary congestion impairing the ability of the GCS to establish an effective data-control loop with the UAV.

This paper addresses these important problems by proposing a robust multi-hop multi-path framework for the remote control of UAV systems. The data and control links are established using the Wi-Fi infrastructure available in urban environments. In particular, we use the 2.4GHz ISM band, which is shared with other Wi-Fi devices and used by other wireless technologies. The multiple paths from the GCS and UAV are continuously probed to quickly select the best option. Importantly, simple local measurements, such as channel sensing and signal strength, would not protect the GCS-UAV communications against local network congestion.

The framework employs a multi-hop multi-path beacon forwarding technique to continuously monitor the performance of the paths from the GCS to the UAV. The UAV measures beacon delay and loss to migrate control routing from one path to another when the current path falls outside of a predefined Quality of Service (QoS) region.

The main contributions of this paper are: *(a)* A cooperative networking model which establishes multi-hop routes using available Wi-Fi Access Points (AP) to forward control messages from the GCS to the remote UAVs; *(b)* A framework to dynamically adapt the route used to forward control messages from the GCS to the UAVs based on the current QoS of the paths; and *(c)* A real-world implementation and extensive field experimentation of the proposed framework.

Experimental results show a considerable improvement in terms of control messages reliability, which leads to a reduced delay in accomplishing mission objectives.

The rest of the paper is organized as follows. In Section II, we discuss related work and emphasize the main innovations introduced by this paper with respect to existing frameworks. Section III presents the architecture and describes the adaptive communication and control strategy used to dynamically select the best path from the GCS to the UAV. In Section IV, we describe the experimental setup and provide numerical results assessing the performance of the framework. Section V and VI conclude the paper.

II. RELATED WORK

Due to the exponentially increasing diffusion of UAVs, the development of effective communication frameworks supporting their operations has received considerable attention in recent years [2]. The interested reader can find in [3] a detailed survey on the challenges of UAV communications in terms of mobility, fast topology changes, and connectivity. An investigation of IEEE 802.11a applied to UAV-to-ground links can be found in [4]. However, an organic and comprehensive solution to these issues is still missing.
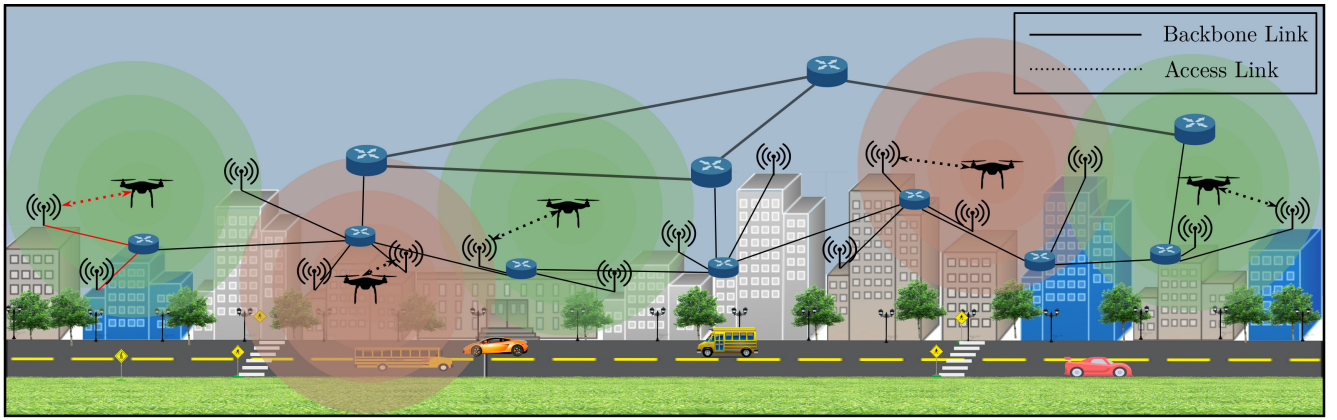
Figure 1. Unmanned Aerial Vehicles (UAVs) operating in an urban environment. The dynamics of traffic created by competing applications and the high mobility of the UAVs make robust control challenging. In this paper, we present a multi-hop multi-path adaptive networking strategy to solve those issues.

Related to the methodology used in this work, [5] presents a study on UAV systems supporting the connectivity of wireless sensor networks. In [6], the authors propose an analytical framework to partition the geographical region and maintain a connected graph of UAV nodes. A framework to make UAV networks self-organizing is presented in [7]. The methodology is based on beacons, whose failure trigger navigation directive to maintain connectivity. Other contributions address the problem of dynamic routing over wireless networks composed of fast moving UAVs, referred to as Flying Ad-hoc Network (FLANET). The solution in [8] extends an existing routing protocol to address as hoc networking scenarios.

Most of these works consider a network scenario where the spectrum is exclusively used by the UAV network. The coexistence of UAV communications with other wireless services has received little attention so far. However, this issue is becoming increasingly important as UAVs are deployed in urban environments. This paper proposes a framework which integrates UAV systems in the Urban IoT using available communication resources to route control messages. A dynamic path selection mechanism ensures robustness against congestion generated by other data streams using the same infrastructure and spectrum. Different from most contributions in this area, we provide a full implementation and experimental investigation.

III. SYSTEM ARCHITECTURE

A. Preliminaries

Fig. 1 illustrates the scenario considered in this paper: a network of UAVs immersed in an urban environment where a multitude of other sensing and communication devices operate and coexist. Typically, UAVs are directly interconnected to the GCS which controls their operations. However, due to the topology of urban environments, a direct link would likely fail to provide a satisfactory communication range, with an inevitable drop in the reliability of control messages' delivery.

Importantly, the UAV incorporates fail-safe mechanisms that are activated when the UAV is disconnected from the GCS, including GPS-based return-to-home function and emergency landing. However, in both cases the UAV fails to accomplish the assigned mission. Therefore, preserving the connectivity

between the UAV and the GCS is of paramount importance. To accomplish such objective, we use the wireless Access Points (AP) available in the city to forward control messages from the GCS to the UAV and telemetry data back from the UAV. The APs are interconnected through the backbone network with established minimum cost paths calculated using either Link State or Distance Vector protocols. In the setup we consider in this paper, we focus on communications in the 2.4 GHz band using Wi-Fi technology. However, the same reasoning can be applied to any, or multiple, technologies depending on the communication capabilities of the UAVs.

B. Architecture

Current approaches addressing connectivity in urban environments primarily use Received Signal Strength Indicator (RSSI) to perform AP selection. However, each individual AP and the router involved in the path to the AP may be also supporting other data streams, which may create localized congestion and affect the performance of a subset of the possible paths. Intuitively, messages from the GCS have stringent delay requirements, where excessive delay may affect controllability, or trigger fail-safe mechanisms as mentioned earlier. The architecture we propose is specifically designed to be robust against congestion and traffic dynamics. To accomplish this objective, we integrate RSSI with performance metrics evaluated in real-time indicating the current state of entire forwarding paths. Informed by the computed metrics, the framework, then, implements a flexible make-before-break handover mechanisms which dynamically selects the best path.

The performance of each path from the GCS to the UAV is measured using beacon messages. Specifically, the GCS periodically generates beacons: small packets containing the generation timestamp and the destination AP information. These beacons are forwarded to all the APs that the GCS can reach through the backbone network. The UAV monitors all the WiFi channels and capture the broadcast beacons from all the APs in its vicinity.

As illustrated in Fig.2, the framework we proposed consist of different functional blocks at the GCS and UAV. The functional blocks at the GCS are: *Control Generator*, *Bea-*
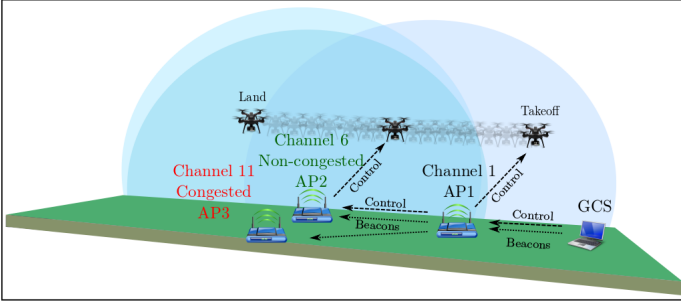
Figure 2. Proposed flexible and robust communication architecture.

con Generator and *Handover Manager*. The UAV functional blocks are: *Deep Packet Inspector*, *Performance Analyzer*, *Decision Manager* and *Handover Manager*. In the following, we describe each of these blocks in detail.

□ **GCS - Control Generator**: This block generates and handles the control messages to be forwarded to the UAV. In the considered case, control messages belong to two classes: heartbeat messages and navigation messages. The former are small messages that are periodically generated so that the UAV can monitor the connection with the GCS. Navigation messages instruct the UAV to navigate to the indicated GPS coordinates and the speed. These messages are defined by the mission control block, which is not explicitly included in the proposed architecture. The Control Generator block adds a generation timestamp and a sequence number to all the control messages. This information is used by the UAV to monitor the quality of the path used to communicate with the GCS in terms of absolute delay and message loss rate.

□ **GCS - Beacon Generator**: Note that control messages are routed only through the path currently used to interconnect the GCS to the UAV. Thus, the timestamps and sequence numbers do not provide any information on all the other possible path options. In fact, broadcasting the control messages over the entire backbone network may increase congestion, especially in scenarios with a large number of UAVs. To address this issue, the Beacon Generator periodically generates small messages – only containing a timestamp and a sequence number – that are broadcast to all the APs and the UAV does not need to be associated with any specific AP to receive the beacons.

□ **UAV - Deep Packet Inspector (DPI)**: This block, implemented at the UAV side, captures all the beacons and control packets. The beacons are collected from all the APs and channels the UAV can receive from, whereas control messages are received only from the currently used path. The block inspects each received packets and creates a data point including the message type, the reception time, the sequence number and the originating AP (MAC address). This information is forwarded to the Performance Analyzer.

□ **UAV - Performance Analyzer**: This block receives the data

points from the DPI block and determines per-message class packet loss rate and average absolute delay. These performance metrics are measured over a moving time window of duration equal to Δ seconds. The metrics are forwarded to the Decision Manager, where the moving average measures are used to trigger handover events (control messages) and select the best path (beacons). In addition to message-related measures, the Performance Analyzer also measures the average RSSI associated with the various interconnected APs.

Note that the duration of the moving window influences the response time and frequency of the framework. On the one hand, a long window better smooths "noise", removing small delay and loss peaks, and avoids frequent handover. On the other hand, a short window allows a faster reaction of the framework to congestion. A thorough study on the effect of this parameter on the performance of control delivery and UAV navigation is not included here due to space constraints, and is deferred to future studies.

□ **UAV - Decision Manager**: The Decision Manager block uses the moving average performance metrics derived by the Performance Analyzer to perform two functionalities: *(a)* Trigger handover to a different AP; and *(b)* Select the best path when a handover event is triggered. In the former functionality, only metrics relative to control messages are used, as handover is necessary only when the QoS of the current path suffers a degradation sufficient to impair the ability of the GCS to control the UAV. The latter functionality considers metrics relative to beacon reception from all the APs, as path selection requires the evaluation of all the feasible paths.

At time instant t the Decision Manager receives moving average beacon delays $D_{\mathrm{b}}^i(t)$, RSSI $R_{\mathrm{b}}^i(t)$ and loss rate $L_{\mathrm{b}}^i(t)$ corresponding to AP i, with $i = 1, \ldots, N$, moving average control delay $D_{\mathrm{c}}(t)$, RSSI $R_{\mathrm{c}}(t)$ and loss rate $L_{\mathrm{c}}(t)$. A handover request is issued at time t if one of the following conditions is satisfied:

$$\lambda_1 D_{\mathrm{c}}(t) + \lambda_2 L_{\mathrm{c}}(t) + \lambda_3 (R_{\max} - R_{\mathrm{c}}(t)) > \Theta; \quad L_{\mathrm{c}}(t) > \Phi, \quad (1)$$

where λ_1, λ_2 and λ_3 are positive weights, with $\lambda_1 + \lambda_2 + \lambda_3 = 1$, and $R_{\max}$ is the maximum RSSI index. Θ and Φ are positive thresholds. The first condition corresponds to a general degradation of the current path. In addition to the first condition, we include in the framework an urgent handover mechanism to recover from harsh events in which the connection with the current AP is abruptly severed. Specifically, is the number of received heartbeats received in the window is below a certain threshold, the handover manager is immediately notified. This event corresponds to the second condition.

If a handover request is issued, the Decision Manager computes the metric

$$W_i(t) = \gamma_1 D_{\mathrm{b}}^i(t) + \gamma_2 L_{\mathrm{b}}^i(t) + \gamma_3 (R_{\max} - R_{\mathrm{b}}^i(t)), \quad (2)$$

for all the APs $i = 1, 2, ..., N$, where γ_1, γ_2 and γ_3 are positive weights, with $\gamma_1 + \gamma_2 + \gamma_3 = 1$ and Θ is a positive threshold defining the minimum accepted performance. The path is selected as

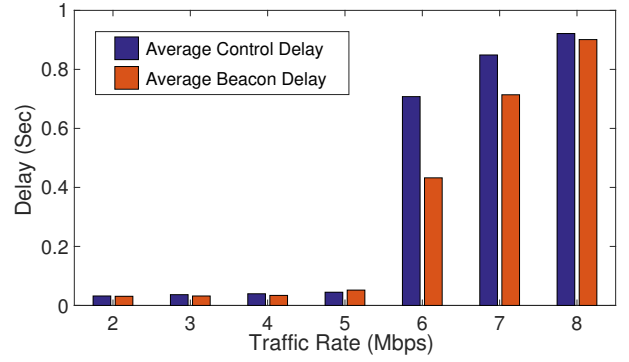$$k = \arg\min_i \{W_i(t)\}. \quad (3)$$

Figure 3. Topology of the experimental setup.

Figure 4. Impact of congestion on beacon and control messages delay. The overall injected traffic volume is equal to 7 Mbps.

Thus, the decision manager selects the kth AP as the new control path if a handover request is triggered. In this case, the Decision Manager forwards to the UAV Handover Manager the handover request and the index of the new selected AP.

□ **Handover Manager**: The handover manager block is located both at the GCS and UAV sides, and implements a 3-way handshake mechanism. The GCS maintains a data structure thats maps the connected UAVs to its corresponding IP address. Each UAV keeps track of the GCS's IP address which we assume to be fixed for the duration of the mission. If a handover request is triggered by decision manager, the handover manager at the UAV associates itself with the AP provided by the decision manager. The Dynamic Host Config-uration Protocol (DHCP) server at the new AP provides an IP address to the UAV. To ensure make-before-break handover, at this point the UAV doesn't disassociate itself from the old AP and keep receiving the control messages through that. Now, the UAV initiate handover by sending a handover request message to the GCS via both the APs to maximize the reception probability at GCS. The handover request message contains the UAV's new IP address and the information of new AP. Upon receiving this handover request message, the GCS sends an approve message and note the information received by the request message. Upon reception of approve message, the UAV complete the 3-way handshake by sending ACK message. After the successful handover, the GCS station forwards the control messages over the new path and the UAV disassociates itself from the old AP to save energy.

IV. EXPERIMENTAL SETUP AND NUMERICAL RESULTS

We assess the performance of the proposed architecture and framework by means of real-world experiments.

A. Experimental Setup

In the considered setup, the backbone network is composed of three APs connected to a GCS. The topology is illustrated in Fig. 3: the GCS is connected with AP1, and AP2 and AP3 are at two hop distance from the GCS. All the three APs operate on non-overlapping channels.

We use a Raspberry Pi (RPi) implementing hostapd to create the APs and all the APs operates according to the IEEE 802.11b standard. The APs communicate with each other via static routing. The GCS, which runs on a laptop, generates beacons every 200 ms and heartbeat message every 500 ms. UDP is used as transport layer for both beacons and control messages. To synchronize the clocks among the UAV and GCS, we use the Network Time Protocol (NTP) with the GCS set as the NTP server.

The UAV is a 3DR solo quad-copter connected to an on-board RPi via a serial link. The RPi is enclosed in a custom 3D printed case. We used the dronekit helper library to communicate with the Pixhawk 2.0 fight controller embedded in the UAV. The RPi is connected with 5 external wireless dongles: 3 dongles are used in monitor mode to capture the beacons in WiFi channel 1, 6, and 11, and the remaining 2 dongles are used to support the make-before-break handover. tcpdump is used to capture the beacons.

The UAV operates in Guided mode, which uses GPS to navigate to way-points (latitude and longitude coordinates). The GCS transmits a predefined series of messages instructing the UAV to navigate to checkpoints. We consider two conges-tion scenarios using the Iperf utility: *Scenario 1:* a continuous stream of traffic is routed through AP3; and *Scenario 2:* the competing traffic stream is alternated between AP2 and AP3.

B. Numerical Results

Fig. 4 shows the average beacon and control delay for different levels of traffic injected at AP3, with which the UAV is connected. The maximum achievable traffic volume traversing an individual AP is equal to 8 Mbps. It is apparent how congestion affects delay as it approaches the maximum supported rate. We observe that in the congestion region, control delay suffers a larger impact of congestion. This is most likely due to the larger size of control packets with respect to beacons.

Fig. 5 depicts the average delay of control messages achieved by different handover strategies in Scenario 1 and 2. The overall injected traffic volume is equal to 7 Mbps. We test an RSSI-based handover strategy against our adaptive handover framework. It can be observed the considerable reduction in delay granted by the proposed framework. Note that the delay in the RSSI-based handover strategy halves in Scenario 2 with respect to Scenario 1. In fact, in the former the congestion is equally spread through the APs, with the UAV connected to one of them in periods uncorrelated with respect to the congestion level. The delay obtained using the proposed technique increases in Scenario 2, where the UAV is
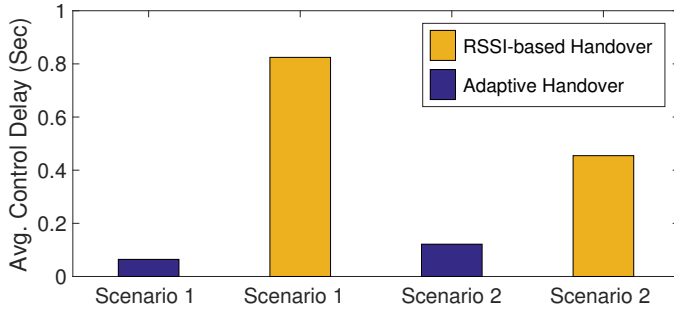
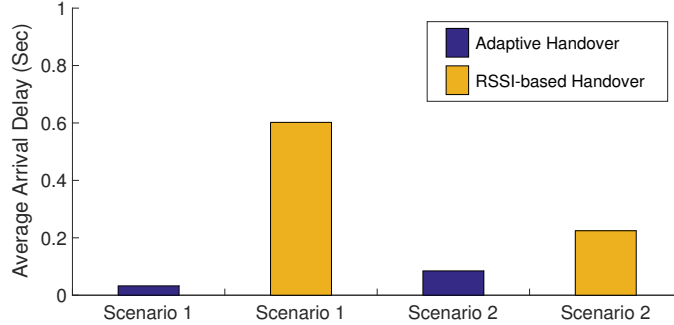Figure 5. Average control delay obtained by the handover strategies in Scenario 1 and 2.

Figure 6. Average arrival delay to meet the predefined checkpoints in Scenario 1 and 2.

forced to shift between AP2 and 3, suffering a delay penalty due to congestion detection and the establishment of the new forwarding connection for control messages.

We observed that RSSI and delay are largely uncorrelated in the considered scenario. In fact, although RSSI influences the maximum transmission rate of the direct wireless link between the APs and the UAV, beacon (and control) messages are small messages with small transmission time. Congestion of the APs buffer or intermediate router results in a delayed forwarding of the packets. Thus handover necessarily needs to use additional information collected by routing packets through the possible paths connecting the GCS to the UAV. Note that RSSI may play a bigger role in determining the overall delay when heavier data streams, *e.g.*, telemetry, are considered.

In addition to the measurement of network performance metric, we illustrate the beneficial impact of the proposed technique on UAV control. In this experiment, we define a sequence of instructions that guide the UAV through a series of waypoints (GPS coordinates). Fig. 6 depicts the average delay in reaching each individual checkpoint granted by the handover techniques with respect to a case with no congestion in Scenario 1 and 2. The reduced time needed to delivery the control messages from the GCS to the UAV granted by the proposed technique results in a reduced delay in reaching the waypoints with respect to RSSI-based handover. Again, we not the same trend where Scenario 2 mitigates congestion in RSSI-based handover and penalizes the proposed technique due to the more frequent handover events triggered by the alternated traffic injection.

Fig. 7 shows the temporal traces of the relative delay. It can be observed that RSSI-based handover incurs periods of large
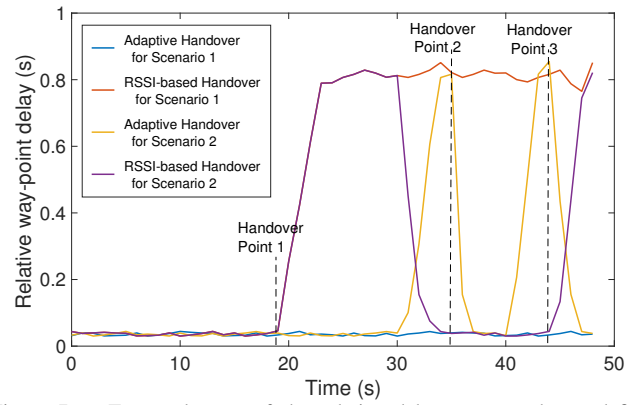

Figure 7. Temporal trace of the relative delay to meet the predefined checkpoints in Scenario 1 and 2.

delay when congestion affects the AP used to communicate with the GCS. The proposed technique has short delay peaks corresponding to handover events.

V. ACKNOWLEDGMENTS

VI. CONCLUSIONS

In this paper, we developed a robust communication framework for UAVs operating in congested urban environments. The framework builds a multi-hop multi-path infrastructure used to connect the Ground Control Station to the UAVs. The paths are dynamically selected based on beacon messages periodically broadcast over all the paths. Experimental results shows that the proposed framework considerably improves the reliability and effectiveness of control against local congestion.

REFERENCES

[1] M. Rodrigues, D. F. Pigatto, J. V. Fontes, A. S. Pinto, J.-P. Diguet, and K. R. Branco, "Uav integration into ioit: Opportunities and challenges," *ICAS 2017*, p. 95, 2017.

[2] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (fanets): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.

[3] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[4] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Channel measurements over 802.11 a-based uav-to-ground links," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 1280–1284.

[5] E. P. De Freitas, T. Heimfarth, I. F. Netto, C. E. Lino, C. E. Pereira, A. M. Ferreira, F. R. Wagner, and T. Larsson, "Uav relay network to support wsn connectivity," in *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2010, pp. 309–314.

[6] T. Ha, "The uav continuous coverage problem," AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH DEPT OF OPERATIONAL SCIENCES, Tech. Rep., 2010.

[7] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military uav relay networks," *IEEE Communications Letters*, vol. 20, no. 4, pp. 804–807, 2016.

[8] S. Rosati, K. Krużelecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic routing for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1690–1700, 2016.