REVISTA PERSPECTIVAS

VOL. 8, N°1 / ENERO - JUNIO 2026 / e-ISSN: 266-6688

DOI: https://doi.org/10.47187/perspectivas.8.1.249

RED DE DRONES EN MOVIMIENTO BASADA EN SDN PARA LA TRANSMISIÓN DE VIDEO EN TIEMPO REAL PARA VIGILANCIA ÁREA

Mobile SDN-Based Drone Network for Real-Time Video Streaming in Aerial Surveillance

Anthonny Flores	flrnhn98e242z605v@studenti.unical.it
Sebastian Ruiz	rzjmsb97r14z605w@studenti.unical.it

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica
Università della Calabria
Rende CS, Italy

RESUMEN

Este trabajo presenta y evalúa una red de drones basada en redes definidas por software (Software Defined Networking, SDN) para la transmisión de video en tiempo real orientada a la vigilancia aérea. La arquitectura utiliza un backbone cableado de puntos de acceso (AP) gestionados por el controlador Ryu, empleando el Protocolo de Árbol de Expansión (Spanning Tree Protocol, STP) para evitar bucles, mientras que los drones actúan como nodos inalámbricos que transmiten video en tiempo real hacia una estación base que simula el centro de control. La simulación integra CoppeliaSim y Mininet-WiFi mediante un servidor socket, y el streaming de video se genera con VLC. la escalabilidad se estudia incrementando el número de drones de tres a siete donde se analizan las métricas como: el throughput efectivo pasa de 2,75 a 7,21 Mbit/s, el ancho de banda medio se mantiene entre 6,93 y 7,99 Mbit/s, el jitter permanece por debajo de 1 ms y el tiempo de ida y vuelta (Round-Trip Time, RTT) varía de 8,53 a 8,99 ms, mientras que la pérdida de paquetes aumenta de 21,34 % a 24,43 %. Al comparar distintos exponentes del modelo de propagación para tres drones, el RTT crece de 12,57 ms (exponente 2) a 18,53 ms (exponente 4), mientras que el throughput se mantiene alrededor de 2,75–2,76 Mbit/s y la pérdida de paquetes entre 31 % y 32 %. En conjunto, la arquitectura escala adecuadamente hasta cinco drones y presenta una congestión moderada con siete. Como trabajo futuro se propone extender la arquitectura a redes con múltiples controladores SDN y estudiar protocolos de enrutamiento específicos para drones en la transmisión de video, incorporando el análisis de la Calidad de Experiencia (Quality of Experience, QoE), la Calidad de Servicio (Quality of Service, QoS) y el consumo energético.

Palabras Clave: Redes Definidas por Software, Vehículo Aéreo No Tripulado, Transmisión de video en tiempo real, Vigilancia aérea, Protocolo de Árbol de Expansión, Throughput, Jitter, Pérdida de paquetes, Handover.

ABSTRACT

This paper presents and evaluates a drone network based on Software-Defined Networking (SDN) for real-time video transmission aimed at aerial surveillance. The architecture uses a wired backbone of access points (APs) managed by the Ryu controller, employing the Spanning Tree Protocol (STP) to prevent loops, while the drones act as wireless nodes that transmit real-time video to a base station simulating the control center. The simulation integrates CoppeliaSim and Mininet-WiFi through a socket server, and video streaming is generated using VLC. Scalability is studied by increasing the number of drones from three to seven, analyzing metrics such as: effective throughput, which increases from

Fecha de Recepción: 13/08/2025. **Fecha de Aceptación:** 02/12/2025. **Fecha de Publicación:** 28/01/2026

2.75 to 7.21 Mbit/s; average bandwidth, which remains between 6.93 and 7.99 Mbit/s; jitter, which stays below 1 ms; and round-trip time (RTT), which ranges from 8.53 to 8.99 ms, while packet loss increases from 21.34% to 24.43%. When comparing different propagation model exponents for three drones, RTT increases from 12.57 ms (exponent 2) to 18.53 ms (exponent 4), while throughput remains around 2.75–2.76 Mbit/s and packet loss between 31% and 32%. Overall, the architecture scales adequately up to five drones and shows moderate congestion with seven. As future work, it is proposed to extend the architecture to networks with multiple SDN controllers and to study drone-specific routing protocols for video transmission, incorporating the analysis of Quality of Experience (QoE), Quality of Service (QoS), and energy consumption.

Keywords: Software Defined Networking, Unmanned Aerial Vehicle, Real-Time Video Streaming, Aerial Surveillance, Spanning Tree Protocol, Throughput, Jitter, Packet Loss, Handover.

I. Introduction

Aerial surveillance has advanced considerably due to recent progress in Unmanned Aerial Vehicles (UAVs), whose versatility has enabled a wide range of military and civilian applications. Current literature highlights that UAV systems are increasingly used in domains such as law enforcement, border monitoring, and emergency response due to their ability to operate in areas that are inaccessible or unsafe for ground personnel [1]. In situations involving natural disasters or hazardous environments, UAVs provide a practical means of acquiring timely information while avoiding the risks associated with direct human intervention [2].

An Aerial Surveillance System is usually a remotely piloted or pre-programmed flying device that can transmit data in real time back to control centers to make up an aerial surveillance system. Since they can move beyond road infrastructure, these UAVs have several advantages over terrestrial vehicles, such as faster operating speeds and greater mobility [2][3]. Even though UAV technology is becoming more and more significant, maintaining proper security is still a major worry. Conventional surveillance techniques, like depending only on security guards, frequently have trouble keeping an eye on large urban areas. When compared to traditional Closed-Circuit Television (CCTV) systems, UAV systems greatly improve surveillance capabilities by overcoming these constraints by offering extensive aerial coverage [4].

The use of multiple aerial vehicles instead of a single drone base offers both economic and operational advantages. Deploying several small drones connected through a communication system is more cost-effective than relying on one large drone, while also providing wider coverage and faster task completion. A major benefit of multi-drone systems is their ability to preserve mission continuity even if individual units fail, thanks to cooperative communication. Multiple UAVs can collaborate to form an Unmanned Aerial Vehicle Network (UAVNet), which is inherently more robust and capable of covering larger areas of interest. Such networks are designed for scalability, allowing additional drones to be incorporated as operational demands evolve. Furthermore, the communication links between drones support the formation of aerial relay networks capable of distributing information across extensive geographic regions [5],[6].

Various reviews on UAV networks show that systems consisting of multiple aerial vehicles are particularly suitable for civil monitoring and surveillance applications, as they allow for greater coverage, increased robustness against failures, and greater flexibility in mission design compared to single-drone deployments. In these proposals, UAVs are explicitly treated as network nodes that exchange information and cooperate with each other, giving rise to UAV networks or Flying Ad Hoc Networks (FANETs) capable of providing connectivity and aerial relay functions over large areas [7], [8].

Using Software-Defined Networking (SDN) in UAV networks is a new way to handle a lot of data collection quickly and easily. By separating the network's control and data planes, SDN technology makes data management processes more flexible and gives users more control. The control plane is in charge of network signaling, route calculation, system management, and configuration. It is the part of the network that decides how it will behave. On the other hand, the data plane's job is to send packets to their next destinations. So, you can think of the network as a distributed structure that connects a lot of independent devices. SDN architecture separates the control functions from the hardware infrastructure, putting all control operations in a single programmable controller. This centralized control unit lets network admins change and improve how the network works in real time, quickly adapting to new situations with a specific controller app [7],[9],[10].

In wired SDN systems with a static infrastructure, programmability refers to the ability of the control plane to modify data paths as needed, while the data plane implements these decisions by forwarding packets through the assigned interfaces. In contrast, when SDN is employed in UAV networks (UAVNets), programmability also involves managing the movement of the UAVs to prevent collisions or to enhance the performance of the applications, selecting or updating routing paths, and adjusting transmission parameters such as data rate or transmission power in response to performance or energy constraints, among several other functions [5].

From a communications perspective, UAV networks are characterized by high mobility, predominantly line-of-sight links, and a rapidly and frequently changing topology. Recent tutorials on UAV-assisted communications highlight that this three-dimensional and dynamic nature of the channel requires flexible network control mechanisms capable of adapting to variations in the radio environment and connectivity [11]. In this context, the introduction of a logically centralized and programmable control plane, such as that provided by SDN, is a natural choice for reacting to topology changes and adjusting routes and link configurations in near real time.

One of the major difficulties in these environments is the occurrence of broadcast storms. In traditional networks that contain loops, Spanning Tree Protocol is typically employed to create a loop-free logical topology and thereby prevent uncontrolled broadcast propagation. In contrast, within an SDN environment, the controller leverages its global view of the topology to compute a spanning tree in a centralized manner to mitigate the same issue [12].

In this paper, we propose a simple SDN-based drone network design for Aerial Surveillance. The goal is to provide an architecture that is easy to understand and configure while ensuring reliable backhaul connectivity for real-time video transmission. Because of its simplicity, the network can be deployed rapidly in unforeseen or urgent situations requiring immediate aerial monitoring. The design incorporates the STP to prevent loops in the backbone and maintain stable packet forwarding. In addition, the simulation integrates Mininet-WiFi with CoppeliaSim, enabling a combined evaluation of network behavior and drone mobility. Recent literature on experimentation with UAV swarms emphasizes that, in many cases, a single simulation platform is not sufficient to accurately model both robotic behavior and network aspects. In particular, it has been shown that rigorous experimental design for multi-UAV systems often requires the combination of specialized tools, such as robotic simulators and network emulators, in order to jointly capture mobility, sensory perception, and communication performance [13]. This view supports the use of the hybrid environment adopted in this work, where CoppeliaSim handles the kinematics of the drones and Mininet-WiFi emulates the SDN-based wireless network.

The proposed design enables uninterrupted video streaming over strategic areas, such as regions with elevated crime rates. Key performance metrics are analyzed to evaluate system behavior. The following sections describe the methodology and present the results that characterize the performance and scalability of the network architecture.

II. Background

A. Drones

Drones, also called unmanned aerial vehicles (UAVs), are aircraft that fly without a human pilot on board. They have become very popular due to their mobility, flexibility, and adjustable flight heights. These characteristics allow drones to be used in many areas, such as military operations, surveillance, telecommunications, medical supply deliveries, and rescue missions. In wireless communication systems, drones can act as aerial base stations. Additionally, they can serve as mobile devices, connecting directly to cellular networks for tasks like live video streaming or delivering packages. Another way to classify drones is by their design. Fixed wing: drones like small airplanes fly at high speeds, cannot hover, but have longer flight times. Rotary wing: drones such as quadcopters that can hover in place but typically have shorter flight durations due to higher energy consumption [13].

Compared with ground vehicles and fixed infrastructure, aerial drones benefit from a substantially higher probability of establishing line-of-sight (LoS) links due to their elevated position and reduced obstruction, a behavior extensively characterized in UAV propagation models [14].

B. Software-Defined Networks (SDN)

Software-defined is a network architecture where network control is decoupled from forwarding and is directly programmable. So, SDN is defined by two characteristics, namely decoupling of control and data planes, and programmability on the control plane [15].

SDN separates the routing and forwarding decisions of networking elements (e.g., routers, switches, and access points) from the data plane. Network administration and management become simple because the control plane only deals with the information related to logical network topology, the routing of traffic, and so on. In contrast, the data plane orchestrates the network traffic according to the established configuration in the control plane [16].

C. Ryu Controller

The controller is one of the most relevant elements in Software Defined Networking (SDN), and it is responsible for managing and programming different network applications. There are many controllers with different programming languages, and the protocol versions they support. They are also designed for different environments, like data centers or cloud computing. The RYU controller is an open-source option developed in Python. It supports versions of the OpenFlow (OF) protocol [17].

Fig. 1 illustrates the RYU controller's main components, which help in developing network applications and managing networks. Some of its tools include OFconfig, used to set up the OpenFlow protocol; the Open Virtual Switch Database (OVSDB) library, which manages switch settings and allows users to create, edit, or delete flow table rules; and the NETConfig library, which applies configurations to devices across the network [18].

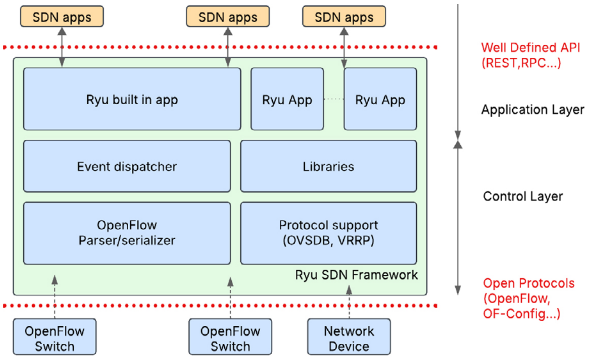


Fig. 1. *Architecture of RYU controller. Source: Adapted from [18]*

D. Mininet WiFi

Mininet is a tool used for emulating networks. Allows create virtual hosts, switches, links, and controllers, all within one machine. This is possible due to container-based virtualization, which allows the system to behave like a real network. It's a cost-effective and reliable option for building and testing applications that use OpenFlow. With Mininet, there's no need to set up physical hardware to try out different network setups, since it can build custom and complex topologies virtually. It also comes with a simple Python interface that makes it easy to design and test networks. [19].

Mininet-WiFi adds wireless functionality by creating virtual Wi-Fi stations (STAs) and access points (APs) that use the mac80211/SoftMAC driver. The driver stack is found in most current Linux wireless cards. Since the majority of Linux wireless drivers rely on mac80211/SoftMAC, Mininet-WiFi can access almost all the features of real Wi-Fi adapters and gives users very fine, low-level control over each wireless packet [20].

E. Coppelia Sim

Coppelia Sim is used for algorithm development, factory automation simulations, fast prototyping and verification, robotics-related education, and remote monitoring. Coppelia Sim is based on a distributed control architecture; each object/model can be individually controlled via a remote API client (Python, Lua, Java, MATLAB, Octave, C, C++, Rust) [21].

F. Propagation Model

Propagation path models represent a set of algorithms and mathematical equations that are used for signal strength estimation in a particular terrain profile. Propagation models can be classified into three types of models. Empirical models, Deterministic models, and Statistical models. Empirical models use a set of equations obtained from the results of several measurements. Deterministic Models use reflection & diffraction laws, which govern electromagnetic signal propagation. Statistical models model the terrain profile as a series of random variables and depend on probability analysis to predict path loss. These models need the least information about the terrain profile and are the least accurate. There are three different area types, namely, Rural, Suburban, and Urban. Rural Area [22].

G. Handover Effect

Handover or handoff is the procedure by which a mobile node transfers its wireless link from one cell to another, reassigning radio resources such as frequency, time slot, spreading code, or a combination thereof without interrupting the ongoing communication. The transfer is typically triggered when the device crosses the coverage boundary of a cell or when the signal quality drops below a set threshold, thereby preserving session continuity and maintaining quality of service in mobile systems [23].

H. Spanning Tree Protocol

Spanning Tree Protocol (STP) is a protocol that operates in the data link layer (Layer 2) of the OSI model. STP allows for defining a loop-free topology by preventing broadcast storms that occur when there are loops. This protocol operates by exchanging messages between switches to determine a root bridge, which is the central point of the spanning tree. Then, the switches calculate the shortest path to the root bridge and disable any redundant links that could create loops. When a link that is part of the active links is disabled, the protocol searches for an alternative link in the network [24].

The root bridge is identified by a unique Bridge ID, which consists of two parts: a configurable priority value and the bridge's MAC address.

In STP, bridges exchange Bridge Protocol Data Units (BPDUS) to evaluate and share information about the configuration of bridges and ports, which determines whether ports should be forwarded or blocked. Therefore, STP defines three types of roles that ports can take.

To establish these roles, the following considerations are as follows:

- The switch with the lowest ID is designated as the root bridge. This switch sets all its ports as designated ports.

- The port on each switch with the lowest cost to the root bridge will be determined as the root port, and the remaining ports will be configured as designated ports. If the cost is the same, the designated port is chosen based on the lowest port ID.

- Designated ports will be set as forwarding ports, while the other ports will be blocked ports.

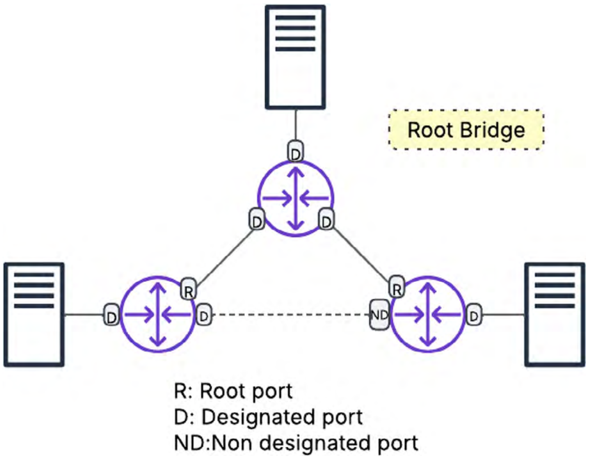


Fig. 2. *STP Port roles*

Fig.2 shows the different roles of each port. The Root Port receives BPDU packets originating from the Root Bridge. The Designated Port forwards BPDU packets received from the Root Bridge to other ports, and Non-designated Ports block frame transmission to prevent network loops. Besides, STP protocol defines four main states for each port:

- **Blocking:** This port is blocked to prevent loops in the network.

- **Listening:** The port processes BPDUs and waits for new information that could cause it to return to the blocking state.

- **Forwarding:** Operates normally by forwarding and receiving frames.

- **Disabled:** This port neither forwards frames nor participates in the spanning tree configuration

Although STP is a protocol used in Ethernet networks, it can also be used in modern networks such as Software-Defined Networks (SDN). However, it has certain limitations due to its lack of suitability for these types of networks. In large-scale networks, this protocol presents some limitations in loop prevention [24].

III. Methodology

A. Equipment and Materials

1) Hardware: The hardware components used to simulate this study include the following:

- DELL Inspiron 15 laptop with 16 GB of RAM and a 512 GB SSD, Intel Core i5 7th 2.5GHz

2) Software: The Software and operating system used in this project are as follows:

- Ubuntu 20.04.6
- Coppelia Sim
- Mininet Wi-Fi
- Ryu Controller
- VLC media player
- Wireshark

B. Network Topology

Fig.3 shows the SDN-based drone network topology, consisting of a wired backbone with four strategically placed Access Points (APs), forming a robust infrastructure that covers the area of interest for aerial surveillance. Within this coverage area, three drones were deployed to analyze the network metrics. The drones used in the simulation are quadcopters that feature four vertical rotors which provide lift and control. An example of such a drone is the Parrot AR.Drone.

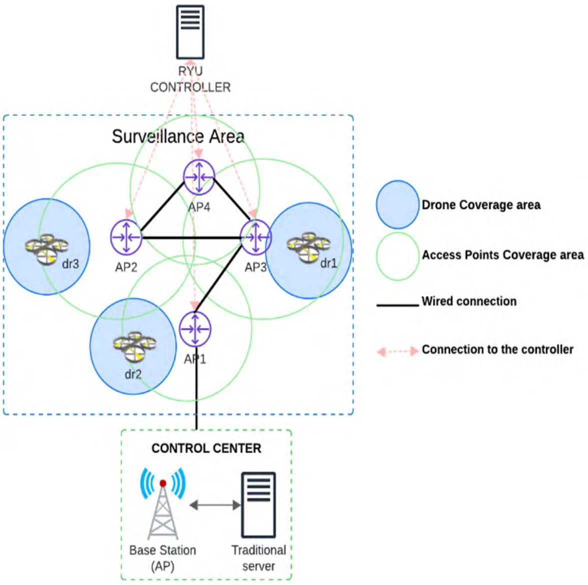


Fig. 3. *Topology of SDN-based drone*

Each drone operates as a wireless AP with a coverage of 20 meters, that continuously sends live video data through the AP backbone network that has a coverage area of 50 meters by each AP. In this case both the AP and drone use WIFI for communication allowing handover when the drone is far from the AP. Therefore, if the drone is far and the coverage is insufficient, it changes the AP and continues sending the information.

The APs create a coordinated network, centrally managed by an SDN controller (Ryu), which can be used to optimize, route and improve network traffic. Therefore, the controller allows maintains communication links and adjusts the data route in real time, ensuring efficient use of network resources to maintain video data quality. The video data transmitted by the drones is sent through the backbone network to the base station, which is represented by an AP that has two interfaces, one wireless and one wired, allowing communication with the server and other APs.

C. Performance Metrics

This project uses Coppelia Sim in conjunction with Mininet-WiFi and the Ryu controller to simulate the behavior of drones navigating the surveillance area, as shown in Fig. 4. This simulation does not consider aspects such as obstacles or wind, which facilitates the evaluation of network performance under optimal operating conditions. The metrics are analyzed at the base station, which is where the data arrives, to understand the behavior of the network.

The metrics evaluated are throughput, packet loss, jitter, round-trip time (RTT), and bandwidth.

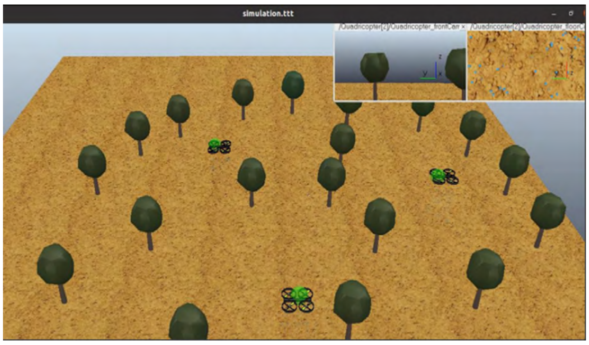

simulation.ttt
/Quadricopter[2]/Quadricopter_frontCam ×
/Quadricopter[2]/Quadricopter_floorC
z
y
x

Fig. 4. *Coppelia Sim simulation of the drone behavior.*

D. Drone Movement and Positioning

For the implementation of the topology, specific parameters were established in each drone for its movement and the camera coverage. The camera used in the simulation is configured with a FOV (Field of View) of 60 degrees vertically, meaning it has a conical aperture of 60 degrees with a field of view from top to bottom. This FOV is the same for all drones.

The movement of each drone is done diagonally up and down, considering movements from the left and right, until it forms a square. Each drone motor is configured to move 100 steps, considering a displacement value of 0.005. Therefore, the drone travels 0.5 meters along the X and Y axes. The total displacement of the configured drone will be 10 meters on each side. Therefore, the maximum coverage area of each drone is 100 square meters as shown in Fig.5.

Related work on disaster management with multi-UAV systems and FANET networks use predefined trajectories and geometric coverage models to study how flight paths and field of view influence the monitored area and network connectivity. These studies show that trajectory design and coordination between multiple UAVs have a direct impact on both coverage quality and wireless link stability [6], [25]. This justifies the use of simple movement patterns, such as square or systematic sweep trajectories, to analyze the effect of mobility on network coverage and performance in a controlled manner.

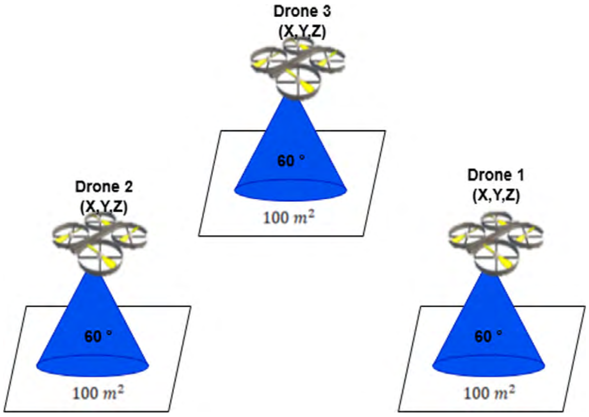


Fig. 5. *Scanning area of the drone.*

To move the drone in the (X, Y) plane, the current position of the drone is obtained, and the relative position concerning its initial position is calculated. This allows for left and right movement, enabling horizontal control of the drone's position. Fig.6 shows a diagram of the drone's movement. The initial position is defined as P0, from which the first displacement moves the drone above the Y-axis and to the right along the X-axis by 10 meters. In the second displacement, P1, the Y- coordinate remains the same, and the drone moves only along the X-axis to the right. In the third displacement, P2, the drone moves below the Y-axis and to the left along the X- axis. Finally, in P3, it maintains the same Y-coordinate and moves to the left along the X-axis, forming a square as shown in Fig. 6.

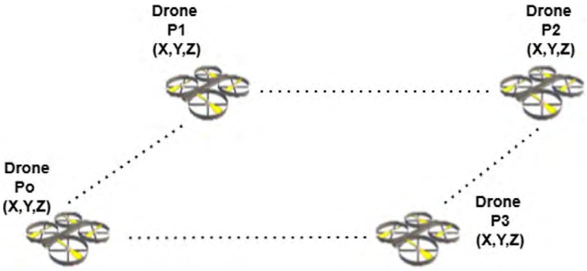


Fig. 6. *Drone movement around the area.*

The Table I shows the main configured parameters for each drone and AP.

Table. I. *Main Network Parameters*

Configuration	Parameter
Height Drone	5m
Conical Aperture (Camera)	60°
APs Coverage Area	50m^2
Drone Coverage Area	20m^2
Dron motor step	100
Displacement Value	0.005

E. Mininet WiFi and Coppelia Sim

To enable communication between Mininet WiFi and Coppelia Sim, a socket server was configured to create a communication channel between them for sending data. Table II shows the configured parameters for establishing communication between Mininet WiFi and Coppelia Sim and the propagation model configured.

Table. II. *Communication Parameters*

Configuration	Parameter
IP Address Server	127.0.0.1
Port	12345
Propagation Model	long-distance
Exponential	3

Due to the simulation being wireless, different propagation models based on the log-distance model can be configured. Some models that Mininet-WiFi allow configuration are specified in Table III with their different exponents that can be configured.

Table. III. *Table of Log Distance Model Exponents*

Exp	Environment	Description
2	Free-Space	Ideal model with no obstacles or reflections
3	Standard Urban City	Urban areas with scattered buildings
4	Densely Built Urban Area	High density of buildings and obstructions

To simulate video acquisition by the drones, video streaming was performed using the VLC media player with the Real-Time Streaming Protocol (RTSP), which operates in conjunction with the Real-Time Transport Protocol (RTP) for multimedia data delivery. To verify that the network was correctly transmitting and receiving video traffic, packets at the base station were analyzed using Wireshark. The captured RTP packets had a payload size of 475 bytes.

The use of real-time video streams as traffic load is consistent with previous studies on UAV communications, where video streaming is used as a representative application due to its high bandwidth consumption and sensitivity to channel variations. It has been shown that, in IEEE 802.11 links used by drones, throughput fluctuations, latency, and jitter have a direct impact on service continuity and perceived video quality [27], [28]. For this reason, the evaluation of metrics such as throughput, RTT, jitter, and packet loss is essential when analyzing communication architectures for real-time aerial surveillance.

General topology can be observed using the Mininet-WiFi Graph tool, as shown in Fig. 7, to verify that the topology was configured correctly, where the Base Station is represented by ap5, where the metrics were collected to evaluate network performance.

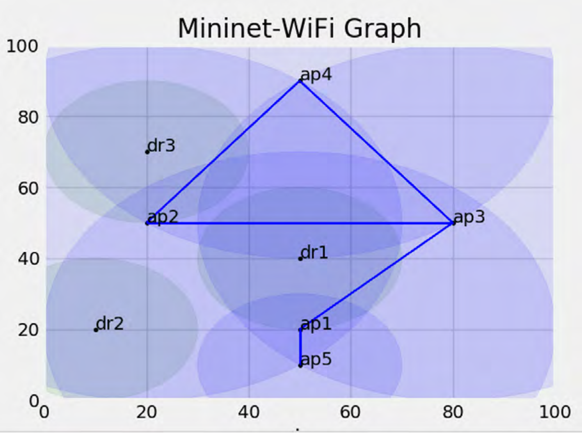


Fig. 7. *Mininet-WiFi of the topology.*

F. Spanning Tree Protocol Configuration

The STP protocol avoids looping between APs, allowing redundancy-free communication on the links. This protocol is implemented within the RYU controller. The AP with the lowest priority is defined as the root bridge, which is the central point to build the loop-free tree. Each bridge has a unique identifier called the bridge ID, which is made up of the MAC address and a previously configured priority value. STP has three types of ports, which are root port, designated port, and non-designated port. Through these ports, the APs will be able to send, receive, or block packets, avoiding loops.

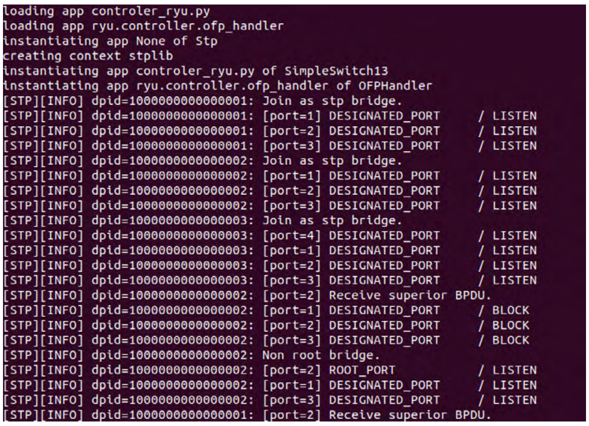
Table IV shows the STP configuration, setting the priority of each AP. The assigned value is entered in hexadecimal format, where the lowest value configured is 0x5000. Once the priorities have been established, a table is defined that contains the MAC addresses of each port, avoiding unnecessary flooding. When the topology changes, the STP protocol updates the port status on each AP and changes the port type, thus deleting the flows previously installed on the AP and cleaning up the entries in its MAC table. If a packet arrives and the destination MAC address is known to the AP, the packet is forwarded directly. If the MAC address is not known, the AP floods by broadcast until it learns the new MAC address and can send the packet.

Table. IV. *Table of Log Distance Model Exponents*

AP	MAC Address	Status	Priority
1	0x1	Bridge	0x7000
2	0x2	Bridge	0x5000
3	0x3	Bridge	0x6000
4	0x4	Bridge	0x8000
5	0x5	Bridge	0xa000

Fig. 8 presents a real-time console capture from the Ryu controller during STP convergence. The bridge with DPID 1000000000000002 (AP2) is not the root once superior BPDUs arrive. Its ports 2 and 3 are first listed as DESIGNATED_PORT/BLOCK, temporarily stopping traffic while the algorithm recalculates the tree.

Subsequent lines show AP2 designating port 2 as ROOT_PORT and moving it through the LISTEN to LEARN to FORWARD sequence, whereas port 3 remains NON_DESIGNATED_PORT / LISTEN, preventing loops on that segment.



```
loading app controler_ryu.py
loading app ryu.controller.ofp_handler
instantiating app None of Stp
creating context stplib
instantiating app controler_ryu.py of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
[STP][INFO] dpid=1000000000000001: Join as stp bridge.
[STP][INFO] dpid=1000000000000001: [port=1] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000001: [port=2] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000001: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000002: Join as stp bridge.
[STP][INFO] dpid=1000000000000002: [port=1] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000002: [port=2] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000002: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000003: Join as stp bridge.
[STP][INFO] dpid=1000000000000003: [port=4] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000003: [port=1] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000003: [port=2] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000003: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000002: [port=2] Receive superior BPDU.
[STP][INFO] dpid=1000000000000002: [port=1] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=1000000000000002: [port=2] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=1000000000000002: [port=3] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=1000000000000002: Non root bridge.
[STP][INFO] dpid=1000000000000002: [port=2] ROOT_PORT           / LISTEN
[STP][INFO] dpid=1000000000000002: [port=1] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000002: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=1000000000000001: [port=2] Receive superior BPDU.
```

Fig. 8. *Ryu controller STP convergence.*

Other bridges display similar transition ports with the lowest path cost becoming ROOT_PORT or DESIGNATED_PORT, advance to the LEARN state for MAC table population, and finally reach FORWARD once the topology stabilizes, while higher-cost ports stay blocked. Overall, the log illustrates how STP systematically suppresses redundant paths, elects forwarding interfaces, and restores full connectivity without broadcast storms.

IV. Results

For this section Iperf tool was used to generate UDP traffic with a fixed bandwidth of 10 Mbps.

This value represents the maximum transmission rate, which means the sender will attempt to transmit packets at that rate. The metrics that were considered for our project are throughput, packet loss, and jitter. Table II presents a summary of all metrics obtained by increasing the number of drones. Round-trip time (RTT) delay, however, will be measured separately using TCP flows to provide an accurate characterization of end-to-end latency. To obtain the averages, a total of twenty measurements were taken for each case presented in this project.

A. Metrics

1) Round-Trip Time (RTT), commonly referred to as network delay or latency, is the total time it takes for a data packet to travel from a source host to a destination host and back again.

RTT is typically measured in milliseconds (ms) and includes all delays encountered along the network path, such as processing delays at intermediate nodes, queuing delays, propagation delays, and transmission delays.

2) Bandwidth: Refers to the maximum data-carrying capacity of the wireless link, expressed in megabits per second (Mbits/s). It represents the theoretical upper limit on how much information can be transmitted over the channel in one second.

3) Jitter: refers to the variation in latency (packet delay) experienced by data packets traveling across a network. Unlike RTT, which measures the average round-trip delay, jitter specifically captures fluctuations and inconsistencies in packet delivery times.

4) Packet Loss: refers to the comparison between the total packets received in comparison to the packets sent using a UDP protocol.

5) Throughput is a key performance metric in networking, defined and represents how data can be successfully transmitted from one node to another in a given amount of time.

B. Comparison of metrics between different numbers of drones

Table V summarizes the metrics obtained for different numbers of drones. The data in Table V shows two distinct behaviors. First, the average bandwidth remains virtually constant: it increases slightly when going from three to five drones and returns to a very similar value with seven drones, so that no significant variation is observed. The same is true for jitter and RTT, which remain within a narrow range without significant degradation. In contrast, effective throughput grows almost linearly as the number of drones increases, reflecting the expected aggregation of throughputs while the links do not reach saturation.

Table. V. *Metrics For Different Numbers of Drones*

Metric	3 Drone Avg	5 Drone Avg	7 Drone Avg
Bandwidth (Mbits/s)	6.93	7.99	7.75
Jitter (ms)	0.55	0.61	0.58
Packet Loss (%)	21.34	22.20	24.43
Throughput (Mbits/s)	2.75	5.20	7.21
RTT delay (ms)	8.53	8.62	8.99

As shown in Fig. 9, the effective throughput grows almost linearly as the number of drones increases, reflecting the expected aggregation of throughputs while the links don’t reach saturation.

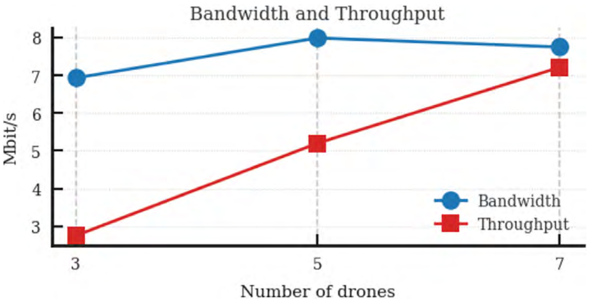


Fig. 9. *Average bandwidth and throughput versus number of drones.*

As shown in Fig. 10, both Jitter and RTT remain highly stable as the number of drones increases. Jitter stays below 1 ms in all cases, indicating consistent packet timing even as additional nodes are introduced. RTT also exhibits only a slight increase, rising from 8.53 ms with three drones to 8.99 ms with seven drones. This steady behavior suggests that the network maintains low latency and does not encounter congestion effects that would significantly impact temporal performance.

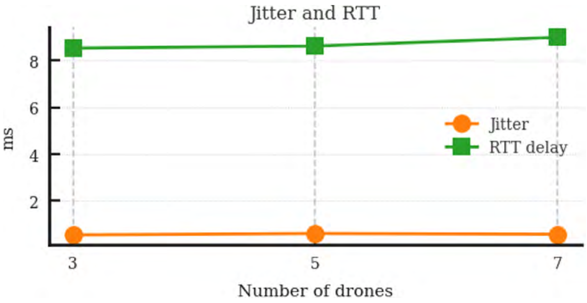


Fig. 10. *Average Jitter and RTT versus number of drones*

Fig. 11 shows that packet loss increases steadily as the number of drones grows. The loss rate rises from 21.34% with three drones to 22.20% with five drones and reaches 24.43% with seven drones. This upward trend suggests a gradual increase in channel contention and interference, which results in a higher probability of packet drops as additional nodes share the wireless medium.

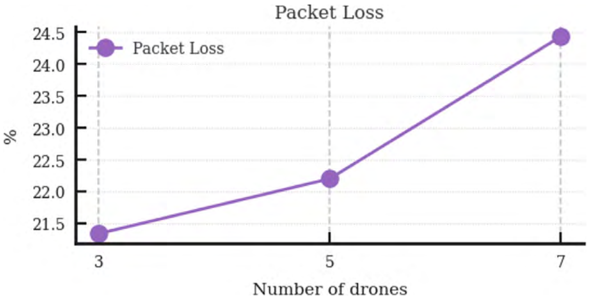


Fig. 11. *Average Packet Loss versus number of drones*

C. Comparison of metrics between different propagation exponents

The Log-Distance model has different exponents that refer to different environments, which were specified in Table III. In this way, a comparative analysis of the metrics in different environments was performed, as shown in Table VI.

Table. VI. *Metrics Average for 3 Drones for Different Propagation Exponents*

Propagation Exponent	Bandwidth (Mbits/s)	Jitter(ms)	Packet Loss (%)	Throughput (Mbits/s)	RTT delay (ms)
exp = 2	7.11	0.507	31.00	2.76	12.57
exp = 3	6.93	0.55	32.00	2.75	14.30
exp = 4	7.45	0.748	31.33	2.76	18.53

Table VI also summarizes the metrics obtained for the three-drone scenario under different propagation exponent values (i.e., exp = 2, 3, 4). The average bandwidth remains around 7 Mbit/s (7.11, 6.93, and 7.45 Mbit/s). The channel's conditions mainly affect latency and temporal stability, while aggregate throughput and loss rate remain virtually constant in this range of exponents, as we can observe in Fig. 12.

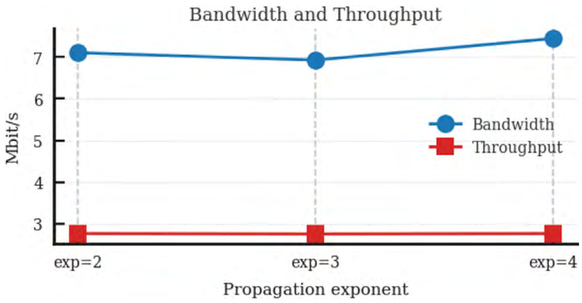


Fig. 12. *Average bandwidth and throughput versus propagation exponent*

The channel conditions mainly affect latency and temporal stability, while aggregate throughput and loss rate remain virtually constant in this range of exponents.

As shown in Fig. 13, RTT increases significantly as the propagation exponent grows, rising from 12.57 ms (exp = 2) to 14.30 ms (exp = 3) and reaching 18.53 ms at exp = 4. Jitter also increases, although to a lesser extent, going from 0.507 ms to 0.55 ms and 0.748 ms, respectively. These trends indicate a noisier channel with greater attenuation, whose clearest impact is reflected in the delays and temporal variability of the link.

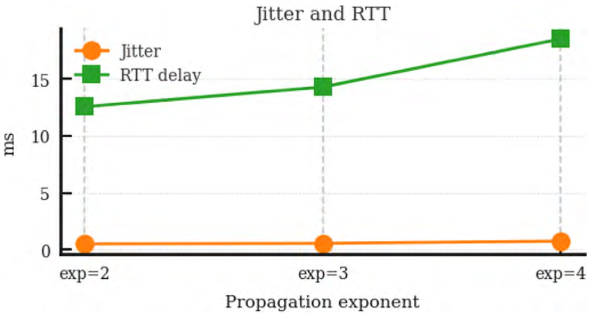


Fig. 13. *Average Jitter and RTT versus number of drones*

As shown in Fig. 14, packet loss presents only slight variations as the propagation exponent changes. The loss rate increases from 31.00% at exp = 2 to 32.00% at exp = 3 and then decreases to 31.33% at exp = 4. Overall, the variations remain small across the evaluated exponents.

Similar behavior is described in studies of wireless communications with UAVs, which show that the trajectory loss exponent and line-of-sight (LoS) or non-line-of-sight (NLoS) conditions directly influence signal attenuation, signal-to-noise ratio, and link coverage probability [11], [14].

These studies show that as the environment becomes more obstructed or the propagation exponent increases, the channel degrades and it becomes more difficult to maintain reliable links with comparable quality levels, resulting in an overall degradation of communication performance. In our scenario, this effect is reflected in the increase in average RTT values and in the greater temporal variability observed when higher propagation exponents are used.

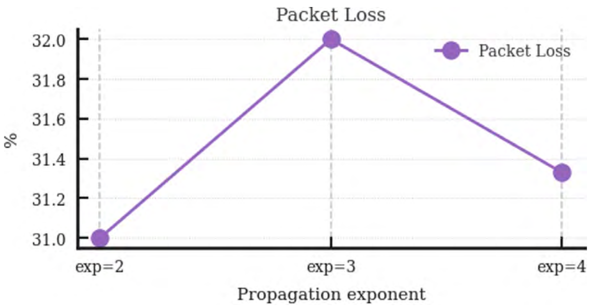


Fig. 14. *Average Packet Loss versus number of drones*

V. Conclusions

This work analyzes the performance of an Unmanned Aerial Vehicle (UAV) network for real-time video transmission based on SDN. A network topology was implemented using Mininet-WiFi, Coppelia Sim, a Ryu controller responsible for managing packet forwarding in the network, and the STP protocol to prevent loops within the backbone network, which consists of four strategically placed Access Points (APs).

The main metrics obtained are throughput, packet loss, jitter, and RTT (Round-Trip Time). The results show that as the number of drones increases, the network consumes more bandwidth, starting with an initial bandwidth of 6.93 Mbps with 3 drones and increasing to 7.75 Mbps with 7 drones. As the number of drones increases, this value will continue to rise, which could saturate the links and cause information loss, and the controller would start to fail at managing the network. When increasing the number of drones, both the throughput and the packet loss increase, resulting in a percentage of 21.43% when considering only 3 drones, rising to 24.43% with 7 drones. Meanwhile, the RTT parameter has a value of 8.53 ms with three drones compared to having seven drones where the delay value is 8.99 ms; in this case, there is a small change that does not affect the transmission; this value depends on the distance at which each drone is positioned, so its value could increase if the distance is very far.

If the metrics are analyzed considering different propagation model exponent values, the RTT and packet loss values vary depending on the environment analyzed, showing a considerable increase from working with an exponent of 2 with an RTT of 12.57ms to a value of 18.53ms when considering an exponent of 4. However, the value of the throughput is the same for all exponent values; this is because this metric is not affected by the propagation environment.

The behavior of the analyzed parameters depends on the efficiency of the management of the RYU controller and the STP protocol, which are responsible for avoiding loops and managing packet transmission between the drones and the APs. However, this study is limited to having a single controller; as future work, it is suggested to incorporate multiple controllers for better traffic management and network optimization.

V References

[1] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. B. Yagoubi, "A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs)," Veh. Commun., vol. 10, pp. 29–56, 2017.

[2] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. B. Yagoubi, "A survey on position-based routing protocols for Flying Ad hoc Networks

(FANETs)," Veh. Commun., vol. 10, pp. 29–56, 2017.

[3] A. Agbeyangi, J. Odiete, and A. Olorunlomerue, "Review on UAVs used for Aerial Surveillance," Journal of Multidisciplinary Engineering Science and Technology (JMEST), vol. 3, 2016, pp. 2458–9403.

[4] N. Dilshad, J. Hwang, J. Song, and N. Sung, "Applications and challenges in video surveillance via drone: A brief survey," in 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020, pp. 728–732.

[5] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint," IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2624–2661, 2016, doi: 10.1109/COMST.2016.2560343.

[6] İ. Bekmezci, O. K. Şahingöz, and Ş. Temel, "Flying Ad-Hoc Networks (FANETs): A Survey," Ad Hoc Networks, vol. 11, pp. 1254–1270, 2013, doi: 10.1016/j.adhoc.2012.12.004.

[7] M. Alharthi, A.-E. M. Taha, and H. S. Hassanein, "An architecture for software defined drone networks", in ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–5.

[8] Z. Zhao et al., "Software-defined unmanned aerial vehicles networking for video dissemination services", Ad Hoc Netw., vol. 83, pp. 68–77, 2019.

[9] S. Tomovic, M. Pejanovic-Djurisic, and I. Radusinovic, "SDN-based mobile networks: Concepts and benefits," Wirel. Pers. Commun., vol. 78, no. 3, pp. 1629–1644, 2014.

[10] A. Kumar, R. Krishnamurthi, A. Nayyar, A. K. Luhach, M. S. Khan, and A. Singh, "A novel Software-Defined Drone Network (SDDN)-based collision avoidance strategies for on-road traffic monitoring and management," Veh. Commun., vol. 28, no. 100313, p. 100313, 2021.

[11] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges," IEEE Communications Magazine, vol. 54, no. 5, pp. 36–42, 2016, doi: 10.1109/MCOM.2016.7470933.

[12] S.-Y. Wang, C.-C. Wu, and C.-L. Chou, "Constructing an optimal spanning tree over a hybrid network with SDN and legacy switches," in 2015 IEEE Symposium on Computers and Communication (ISCC), 2015, pp. 502–507.

[13] A. Phadke, F. A. Medrano, C. N. Sekharan, and T. Chu, "Designing UAV Swarm Experiments: A Simulator Selection and Experiment Design Process," Sensors, vol. 23, no. 17, Art. no. 7359, 2023, doi: 10.3390/s23177359.

[14] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," arXiv [cs.IT], 2018.

[15] Bor-Yaliniz and H. Yanikomeroglu, "The New Frontier in RAN Heterogeneity: Multi-Tier Drone-Cells," IEEE Commun. Mag., vol. 54, no. 11, 2016, pp. 48–55.

[16] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," IEEE Commun. Surv. Tutor, vol. 17, no. 1, pp. 27–51, 2015.

[17] K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, "Software-defined networking (SDN): a survey: Software-defined networking: a survey," Secur. Commun. Netw., vol. 9, no. 18, pp. 5803–5833, 2016.

[18] A. T. Albu-Salih, "Performance evaluation of Ryu controller in software defined networks," J. Al-Qadisiyah Comput. Sci. Math., vol. 14, no. 1, p. Page 1-7, 2022.

[19] K. Kaur, J. Singh, and N. S. Ghumman, "Mininet as software defined networking testing platform," in Proc. Int. Conf. Commun., Comput. & Syst. (ICCCS), Aug. 2014, pp. 139–142.

[20] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-WiFi: Emulating software-defined wireless networks," in 2015 11th International Conference on Network and Service Management (CNSM), 2015, pp. 384–389.

[21] "Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics," Coppeliarobotics.com. [Online]. Available: https://www.coppeliarobotics.com/. [Accessed: 25-Jul-2025].

[22] V. S. Anusha, G. K. Nithya, and S. N. Rao, "A comprehensive survey of electromagnetic propagation models", in 2017 International

Conference on Communication and Signal Processing (ICCSP), 2017, pp. 1457–1462.

[23] "Effective management of handover process in mobile communication network", J. Adv. Technol. Eng. Res., vol. 2, núm. 6, 2016.

[24] E. Amiri y R. Javidan, "A new method for layer 2 loop prevention in software defined networks", Telecommunication Syst., vol. 73, n.º 1, pp. 47–57, julio de 2019. Accedido el 31 de julio de 2025. [En línea]. Disponible: https://doi.org/10.1007/s11235-019-00594-4

[25] M. Erdelj, M. Król, and E. Natalizio, "Wireless Sensor Networks and Multi-UAV Systems for Natural Disaster Management," Computer Networks, vol. 124, pp. 72–86, 2017, doi: 10.1016/j.comnet.2017.05.021

[26] B. Van den Bergh, A. Chiumento, and S. Pollin, "Ultra-Reliable IEEE 802.11 for UAV Video Streaming: From Network to Application," in Advances in Ubiquitous Networking 2, UNet 2016, Lecture Notes in Electrical Engineering, vol. 397, Springer, 2016, pp. 637–647, doi: 10.1007/978-981-10-1627-1_50.

[27] S. Kacianka and H. Hellwagner, "Adaptive Video Streaming for UAV Networks," 2015, doi: 10.1145/2727040.2727043.