Investigate and Compare Software-Defined Network Controllers for UAV Networks Management

Hussein Ali Al-Gboury$^1$ and Sahar Abdulaziz Al-Talib$^1$

$^1$Computer and Information Engineering Department, Electronics Engineering College, Ninevah University, Mosul, Iraq

Corresponding author’s e-mail: hussein.mohammed@uoninevah.edu.iq

Abstract. Unmanned aerial vehicles that abbreviates (UAVs) are flying platforms, known as drones, which have features such as mobility, adaptive altitude and flexibility. UAVs admit numerous applications that can be used as aerial base stations to enhance coverage, capacity, and energy efficiency of wireless networks. On the other hand, UAVs can operate as flying mobile terminals within a cellular network. Such cellular-connected UAVs can enable several applications ranging from real-time video streaming to item delivery. A Software Defined Network (SDN) Controller is the application that acts as a strategic control point in a software-defined network. It is the “brain” of the network. Controller manages flow control to the routers/switches 'under' (via southbound APIs), the business logic and applications 'above' (via northbound APIs) to deploy intelligent networks. Wireless networking with software defined (SDWN) is the use of SDN conceptions in wireless networks by using a controller in the control plane. SDWN facilitates the creation of new adaptive mechanisms according to various applications and user requirements, such as mobility, handover, security and quality of service (QoS). In this paper, simulation work has been conducted to compare and investigate four SDN controllers (Pox, Ryu, Floodlight and OpenDaylight) in order to see which one is suitable to be used. Mininet-Wifi has been selected as the simulation tool to do the experiments and Python script for programming. The results obtained reveals that Ryu controller is the best selection in terms of latency and packet loss.

Keywords: Mininet-Wifi emulator, SDN UAV Networks, SDN Controllers, SDWMN

1. Introduction
Unmanned aerial vehicles (UAVs) known as drones, are aircrafts without a pilot aboard to control its motion, which can either be remotely or self-controlled. They are one of the emerging technologies that attract a lot of industrial and academic interests. UAV has been well stared as ground-breaking technology to revolutionize future wireless network architectures thanks to their high mobility. UAVs can be deployed as mobile access points or relay nodes response to real-time data traffic changes. Recently, UAVs have been widely applied as aerial access points to assist terrestrial wireless networks [1],[2],[3],[4],[5].

According to Business Insider Intelligence’s reports, there will be more than 29 million UAVs are estimated to be put into use in 2021 [6]. It is predictable that millions of (UAVs), become active in daily life in the recent future and provide wide services. A large number of latest advancements in the technology of UAVs have facilitated them to be very valuable and effective in numerous applications in today’s society. They can be used for military, public, and civil applications. Military usage of...
UAVs is more than 25 years old mainly consisting of border country surveillance and discovery. Public usage is by public organizations such as police, transportation management, and public safety. UAVs can provide timely disaster cautions and assist in acceleration rescue and recovery processes when the public communication networks are disconnected. They can carry medical supplies to unreachable areas. In situations like poisonous gas infiltration, wildfires, and wild animal tracing, UAVs could be used to quickly cover a large area without risking the safety of the personnel involved. Watching and covering large geographical areas is a main requirement to increase safety and improve the operation of future cyber-physical systems, as smart cities, smart oil fields, and smart farms. Furthermore, using manned aerial vehicles in risky missions exposures pilots to high risks and life-threatening conditions; in contrast, using UAVs will eliminate this risk [7],[8],[9],[10]. For over 40 years, UAVs have been a part of NASA's fleet and range from full-scale solar-powered versions to those using electric motors or propellers. Uses of UAVs included remote sensing for earth sciences studies, hyperspectral imaging for agriculture monitoring, tracking of severe storms, and serving telecommunications relay platforms [11]. Figure 1 shows a notional scenario for the Unmanned Aerial System (UAS) traffic management © NASA.

![NOTIONAL SCENARIO](image)

Figure 1. A notional scenario for UAS Traffic Management © NASA [12].

2. Related works
The Researchers have applied many approaches using unmanned aerial vehicles. Various studies are done on using UAV in a wide variety of applications.

Guillen-Perez, et al. [13] use the Intel Galileo development board, easily designed and fitted to function as a Wi-Fi node (either as an Infrastructure Mode access point or as an ad hoc intermediate jump) on a drone, they compared two experimental Wi-Fi operating methods in terms of coverage, reliability, and energy consumption. Results showed better network status with respect to the obtained signal intensity and bandwidth but the worst behavior with respect to current consumption compared to ad-hoc mode.

Zhang et al. [14] developed a software-defined network (SDN) architecture for UAV backbone networks based on the SDN. At the ground control station is mounted an SDN controller. They showed that an SDN deployment can extend the battery life of the UAV due to an integrated load balance algorithm within the SDN controller.
Wang et al. [15] suggested a traffic-aware adaptive UAV deployment scheme in a UAV-assisted communication network where the UAV efficiently adapts its position within the sector with the highest number of users and built the optimum displacement distance to optimize the average throughput of a typical random user in the uplink. They showed that the ideal displacement distance decreases as the density of the consumer increases, and showed that the planned adaptive deployment scheme outperforms solely that of the non-adaptive scheme.

Zacarias et al. [16] demonstrated an implementation of (SDN) to enhance the efficiency of the video streaming experience in the context of mobile military networks. The aim of the approach is to connect a heterogeneous network consisting of UAVs and ground vehicles focusing on applications for surveillance and recognition in military operations. Since the study considers military use, it is possible to use the scenarios in the civil domain too.

Jehad et al. [17] have strictly analyzed and compared the performance of two SDN controllers: Ryu and POX. Experiments were performed over single, linear, tree, dumbbell and Data Center Networks (DCN) topologies in SDN environment. They have also extended the performance comparison tests to satellite communication systems (SATCOM) i.e SDN-SAT system topology for naval tactical networks. The experiments have indicated that the percentage decrease in latency and the TCP throughput increased for different SDN topologies while using Ryu is high. Experimental findings have demonstrated the effectiveness of Ryu controller with hierarchical structure topologies i.e. DCN and tactical networks (SDN-SAT).

Mohannad Alharthi, et al. [18] presented an architecture for software-defined drone network that utilizes SDN to implement different drone mission scenarios. They described use cases that demonstrate the configurability and reusability benefits of the architecture.

3. UAV SERVICES IN COMMUNICATIONS
Drone based communications supply many advantages over present terrestrial wireless communications, such as flexible reconfiguration, flexible deployment, and better channel conditions for user equipment’s [19].

The use of UAVs as flying wireless communication platforms has received substantial attention recently. On the one side, UAV can be implemented as a wireless relay for improving the connectivity of ground wireless devices and spreading network coverage. On the other side, UAV can act as an aerial mobile base station to provide wireless coverage to a geographical area out of the reach and reliable uplink and downlink communications for users on the ground, and enhancing the wireless networks capacity. Compared to the terrestrial base station, the benefit of using UAV as aerial base station is its ability to move quickly and easily. Additionally, the high altitude of UAV can allow a line of sight (LOS) communication links to the ground users [20],[21].

In recent years, aerial communications has been recognized as an interesting alternative for providing wireless services to terrestrial users. Facebook Aquila Drone and Google Loon are two well-known projects that exploit aerial platforms to fill coverage gaps and provide internet access for users in remote locations. The ABSOLUTE project is another important enterprise that uses aerial platforms to improve public safety and accomplish capacity enhancement. The quick deployment of UAVs as aerial platforms, independent of terrestrial infrastructures, makes UAVs effectively appropriate for many operations [22], [23].

The use of aerial wireless communication platforms carried by UAVs is seen as a promising approach to enhance the coverage and capacity of future wireless networks. UAVs can have three key tasks: aerial base stations, aerial relays, and user equipment. Due to their flying nature, UAVs can be harnessed as aerial base stations to provide line of-sight networks toward ground users and, thus, potentially improving the rate, delay, and overall performance of wireless networks. Furthermore, UAVs can be used as user equipment's (UE) connected to the cellular network for information dissemination and data collection. In addition, UAVs can play as relays between a source and a destination for situations in which a LoS link does not exist or for the backhaul connectivity of base stations. However, deploying UAVs for wireless communication features faces numerous challenges
including the modeling air to ground channels, energy efficiency, optimal deployment, resource management, path planning, and security [24][25].

4. SOFTWARE-DEFINED NETWORKING FOR MANAGING UAV NETWORKS

Recent suggestions for future wireless architectures are aimed at creating a flexible network with enhanced agility and resilience. Software-defined networking (SDN) has been introduced in 2008 to program the network via a logically software-defined controller, which can decouple the control plane and data plane to simplify network reconfiguration. This is helpful to manage the infrastructure and resources of wireless networks compared to conventional networking. SDN has better controllability and visibility of network components which enables better management by using the controller. In real-world applications of drone-cells, wireless networks must be configured efficiently for seamless integration/disintegration of UAVs, such as changing protocols and creating new paths. Based on the SDN architecture, UAVs can do as SDN switches on data plane for gathering context information in a distributed way, while the ground BSs are controllers gathering data and making control decisions on network functions and resource allocation. Helped by SDN, network reconfiguration and resource allocation among a swarm of UAVs can be conducted in a more flexible way [26].

The basic principles of SDN can be summarized in two principles. The first is that SDN functionally enables the network to be accessed programmatically to provide a high level of automation in the field of network management technologies. The second principle is to decouple the management and control planes from the data transmission plane, where the policies that command the forwarding rules are centralized, while the actual processing of the forwarding rules is distributed among different network devices. The centralized point view about the whole network through using controller enhances the functionality and performance of network devices. As a result of enabling the fastest possible protection in the incident of a failure, the highest resiliency, and the ability to place new services into a network in one command via using Application Programming Interfaces (APIs). SDN could facilitate flexible deployment and management of novel services and help reduce cost, increase security, and availability in networks [7].

Recently UAV based communications systems have attracted increasing research and commercial interest. In such networks, UAVs carry the equipment, such as a Wi-Fi access point, to provide communication service. UAV-based communication networks are very useful in emergency scenarios, such as post disaster scenario. Unlike most other application domains, such as search and rescue, the mobility of a UAV network offering communication services is minimal, resulting in a relatively stable network topology. On the network infrastructure side, SDN has emerged as a promising approach to conventional IP networks, separating network logic control (control plane) from the data plane. SDN consists of a controller in the central and forwarding elements (switches and routers). The centralized control makes SDN ideal for a UAV network because UAV networks are usually of small sizes. Moreover, all the nodes in a UAV network cooperate towards a common goal; therefore, every node (UAV) is willing to share control information with the SDN controller [27].

In SDNs, the controller collects the network statistics information from switches, then installs flows on switches, and sends administrative policies to switches. The controller also requires knowing about the latest network topology. Therefore, connectivity between the SDN controller and the UAVs is tremendously important. A key feature of a deployment method is the placing of the SDN controller. Figure 2 shows a typical scenario of an SDN-based UAV network. UAVs act as forwarding elements, and the network is managed by the SDN-controller. Each UAV is providing communication to users within its transmission range [27].
There are three types of communication objects in the SDN-based network: UAV, UE (User Equipment), and SDN controller, which are illustrated in Figure 3. The UAVs equipped with Wi-Fi access points (APs) form a mesh network with IEEE 802.11 wireless local area network (WLAN) access technology in the sky.

The UEs can be any on-ground terminal devices, such as a sensor, a mobile phone, and so on, which use the aerial UAV-based backbone network to transmit information. The UEs are distributed on the ground but each of them should be served by at least one UAV. The SDN controller can be either ground-based or sky-based (for example, satellites, high-altitude platforms, or larger UAVs) that communicates with all UAVs to send control packets and receive UAV packets. At the same moment, the UAVs can communicate with each other directly or through multi-hop forwarding [14].

Managing SDN networks require the SDN controller to have updated information related to the condition of the network, especially its topology. Therefore, having a reliable mechanism is the main concern to discover network topology. The SDN controller should have information concerning each network device and the connection channels that link them. A mandatory issue in an SDN network, especially for the large networks, is to keep the service running almost real-time with a vision about the topology of the network [28].

There are many different available SDN controllers such as: NOX [29], POX [30], RYU [31], Pyretic [32], Trema [32], FloodLight [33], ONOS [34], Beacon [34], Maestro[35], MuL [36], and OpenDaylight [37]. In this paper, the Ryu controller has been chosen because it is widely used, provides well document, and defines API for creating various SDN applications. It is also a very fast
python-based OpenFlow controller as compared to the performance of application with Pyretic and POX controllers [38]. Simulation results of different types conducted in this work prove that the Ryu controller is the best selection.

Four open source controllers that have been simulated and compared in this paper will be summarized as follows:

5.1. POX CONTROLLER
POX is a Python-based OpenFlow controller. It is widely used in the research community because it is easy to program. It provides a platform for rapid development of applications and prototyping for controlling network devices in an OpenFlow network. POX can be connected remotely with Mininet and different applications such as firewalls, intrusion detection, and prevention system, load balancer, switching and routing. POX controller can mostly run on Python v2.7, and can officially support Windows, Mac OS, and Linux. Moreover, the POX controller is capable of converting unusual OpenFlow devices to operate as switches, routers, load balancers, firewall devices, etc., and provides better performance to SDN architecture [17], [39].

5.2. RYU CONTROLLER
Ryu is a Japanese word means flow. It is pronounced as "ree-yooh". It is also a python-based OpenFlow controller, which has a strong API through which developers can program, control, and manage applications. Network devices can be configured using various protocols such as Netconf, OpenFlow, and OF-conf, based on the applications running on Ryu. Ryu is an open source and its Apache licensed code. Ryu can be used with OpenFlow to collect statistical information from the switches. So it can be set up as a traffic monitor, firewall, router, and switch [17].

5.3. Floodlight CONTROLLER
The Floodlight controller is an open source, Java language-based OpenFlow controller. It is Apache-licensed and is supported by the largest SDN controller developer community that includes a number of BigSwitch network researchers and engineers. The Floodlight controller is designed as heavily concurrent systems for gaining the throughput required by multiple data centers and enterprise class networks. Since OpenFlow specifies protocols through an OpenFlow-enabled switch, a controller activated remotely can modify the behavior of core networking devices through a well-defined data forwarding instruction set. The Floodlight controller is designed to operate in various routers, switches, access points, etc. which supports standards of OpenFlow. The Floodlight controller can also support and work with hybrid networks where OpenFlow-enabled switches that connected during an existing conventional switches [39].

5.4. OpenDayLight (ODL) CONTROLLER
OpenDaylight, an open source controller, is designed for automating and customizing networks of any scale and size focusing on network programmability. The ODL platform is designed to cover a lot of use cases such as Automated Service Delivery, Network Resources Optimization (NRO), Network Functions Virtualization (NFV), Cloud, and others. Using ODL services such as on-demand services, dynamic network optimization, (i.e. dynamic Virtual Private Network VPN services, bandwidth, etc.), and agile service delivery on cloud infrastructure is provided. ODL may also be used to achieve a centralized administration of the network [40]. Table 1 shows a comparison of different controller models in terms of interfaces, APIs, GUIs, Operating System (OS) support, programming languages, Partners, etc.
6. SOFTWARE-DEFINED WIRELESS NETWORKS (SDWNs)

Software-defined wireless network (SDWN) is a programmable network. It is the networking paradigm that provides an effective solution to cost effective high speed network services when compared with the conventional wireless network. SDWN which is often denoted as a revolutionary new idea in computer networking, has the capability to radically simplify the network configuration, control, and enable revolution through network programming [42].

Wireless Mesh Networks (WMNs) have been considered as one of the best promising wireless technologies to build very scalable wireless backhaul networks [43].

WMN is a wireless network composed of mesh clients and routers, which can achieve wide coverage, high capacity at a low cost. The routers are in charge of creating the communication infrastructure, hence they have limited mobility, and energy consumption is not a critical factor. On the other hand, mesh clients are usually mobile stations and energy consumption becomes relevant. WMN can be viewed as a network operates in both infrastructure and ad-hoc mode, where users communicate with one another using multi hop wireless transmission. This mechanism extends the coverage, without losing the channel capacity, allowing more efficient reuse of frequencies [44].

WMNs have gained significant popularity during the last few years and have evolved from the relatively simple single-radio single channel networks to more powerful multi-radio multi-channel wireless networks with significantly more advanced capabilities. These networks are being deployed throughout the world and offer powerful and processing capabilities. They are being used in military scenarios, disaster communications, community communications, and medical facilitation. In fact, mesh networks have evolved from the traditional community networking to being at the top of several

Table 1. Comparison of SDN controllers [41]

| Software-Defined Interfaces | Ryu | FloodLight | OpenDayLight | ONOS |
|-----------------------------|-----|------------|--------------|------|
| OF1.0, 1.2, 1.3, 1.4, NETCONF, OVSDB | OF1.0, 1.2, 1.3, 1.4, NETCONF, YANG, OVSDB, PCEP, BGP-LS, LISP, SNMP, OVSDB | OF1.0, 1.3, 1.4, 1.5 NETCONF, YANG, OVSDB, PCEP, BGP-LS, LISP, SNMP, OVSDB | OF1.0, 1.3, 1.4, 1.5 NETCONF, YANG, OVSDB, PCEP, BGP-LS, LISP, SNMP, OVSDB |
| REST API | Yes (For SB only) | Yes | Yes | Yes |
| GUI | Yes (Initial phase) | Web-based | Web-based | Web-based |
| Modularity | Medium | High | High | High |
| Orchestration Support | Yes | Yes | Yes | No |
| OS Support | Most supported on Linux, Windows, and MAC | Linux, Windows, and MAC | Linux Windows, and MAC | Linux, Windows, and MAC |
| Partner | Nippe Telegraph and Telephone Corporation (NTT) | Big Switch Networks | Linux Foundation With Memberships Covering Over 40 Companies, like Cisco, IBM, | ON.LAB, Sk Telecom, Cisco, Ericsson, Fujitsu, Huawei, Intel, AT&T, Nortel, Ciena, Ntl, Ntt Communication |
| Documentation | Medium | Good | Very good | Good |
| Programming Language | Python | Java | Java | Java |
| Multi-threading support | Yes | Yes | Yes | Yes |
| TLS Support | Yes | Yes | Yes | Yes |
| Virtualization | Mininet and OVS | Mininet and OVS | Mininet and OVS | Mininet and OVS |
| Application Domain | Campus | Campus | Data center and Transport-SDN WAN | Data center and Transport-SDN WAN |
| Distributed/Centralized | Centralized | Centralized | Distributed | Distributed |
technological innovations [45]. The software-defined networking paradigm (Figure 4) is a novel technique of managing network routing. The basic idea is that instead of routers running distributed routing protocols and deciding the end-to-end route among themselves, a central authority i.e. the controller has complete knowledge about the network, makes optimal routing decisions, and then installs the resulting routing configuration onto the routers. In other words, the routers become more like switches which are only there to forward the packets per the switching table built on them by the controller. This can be summarized as separation of the control plane i.e. routing and the data plane, i.e. user data [45].

Figure 4. Software defined wireless mesh network [45]

WMN is considered to be one of the promising wireless technologies to create a wireless multi-hop backbone network. It provides a coverage service for a varied range of applications in both rural areas and urban, like emergency response, transportation, public safety, enterprise networks, and mining fields. That is because of its quick and relatively low-cost deployment. WMN is described by a dynamic and distributed network control (protocols) and challenged by scalability, management complexity, and reliability issues [46].

A typical wireless mesh network consists of a number of nodes that can be connected to one another in a multi-hop way via wireless connections. The network nodes can be static wireless routers, or mobile devices, e.g., smart phones or laptops, which can join or leave the network at any moment. Due to the high flexibility of wireless mesh networks, it can be simply set up without a fixed infrastructure and has been extensively deployed in numerous applications. For instance, a number of wireless routers can be simply deployed to provide Internet services to mobile users in a large coverage area without the support of dedicated cellular base stations. On the other hand, wireless mesh networks are hard to manage and administer because of dynamic network topology and device diversity. The network may consist of different devices, e.g., wireless routers, laptops, and smart phones, with distinguished processing and communication capabilities. Network topology changes due to user mobility impose a great challenge in optimizing network resource usage. Besides, WMN are subject to various attacks due to their structure-less nature and lack of centralized monitoring points [47].
7. SOFTWARE-DEFINED WIRELESS MESH NETWORKS (SDWMNs) as UAV NETWORK

SDN principles can also be included in WMN produced by OpenFlow switches. WMN can benefit from the flexibility and the simple management provided by the SDN concept. The shortcoming in SDN is that it does not have language interpretability [48].

Wireless networks need specific features such as mobility management, rapid client re-association, and dynamic channel configuration. In cases which require some extension to an existing wired topology, and there is little interest to make fundamental changes to the existing topology, Software-Defined Wireless Networks (SDWN) can be used [49].

The combination of SDN paradigm and WMN architecture is called Software-Defined Wireless Mesh Networking (SDWMN). This approach is challenging because it combines different control natures (i.e. the distributed nature of WMN with centralized nature of SDN). Figure 5 demonstrates the basic logical vision of the SDWMN, where the infrastructure layer exists in at the bottom, it consists of a set of SDN-enabled routers. Routers interconnect to provide the functions of the network data plane.

![Figure 5. SDWMN General Architecture [46]](image)

In the middle, there is the control layer, consisting of software-based SDN controller/s. The controller provides and keeps a global view of the network to the application layer.

It abstracts the complexity of configuring, discovering and managing the distributed network nodes, and preserving connectivity among them. The communication between the controller and the infrastructure layer happens through the southbound interface. OpenFlow which is the dominant southbound standard, founded by Open Network Foundation (ONF).

On the upper of the SDWMN architecture is the application layer, which is responsible for defining and implementing network policy requirements. The northbound interface is the interface between the control layer and the application layer. There is no standard defined for northbound till now [46].

With like flexible architecture carried by SDWMN, programmability and innovation are better prepared at a higher rate. New network applications services and policies can, therefore, be implemented simply via applications such as load balancing, routing, firewalling, etc. running at the upper of the controller. The controller then translates these requests into instructions for the elements in the data plane (mesh routers) [46].
Each UAV functions as an SDN forwarding element (switch) and provides wireless communication services to its associated users. Together, the UAVs form a mesh network that is connected to the SDN controller and the Internet via a gateway [50].

In this work, it is assumed that the SDN controller has up-to-date information about the network topology, positions of UAVs, positions of users, and data rate demand and path of each flow. The SDN controller maintains this information by receiving updates from all forwarding elements (UAVs in the proposed scenario). A control agent running on each UAV gathers information about its neighbors, the positions of its associated users, and the data traffic demands and it sends this information to the SDN controller after regular intervals, the length of which depends on the network dynamics. Similarly, whenever a change in the network conditions occurs, such as the arrival of a new user, the UAV sends an update to the SDN controller immediately. Furthermore, when a new flow needs to be set up, a request is sent to the SDN controller, which computes a path based on the topology and traffic information and installs forwarding rules on every forwarding element along the path. Similarly, the proposed algorithm is executed on the SDN controller. After computing the desired positions for all the UAVs, the SDN controller sends out control packets to each UAV instructing it to move to its desired position [50].

8. Implementation and Simulation Parameters

8.1. Mininet-Wifi Emulator

Mininet-Wifi is an SDN network emulator. It is a branch of the Mininet that expanded the service of Mininet through adding virtualized access points and WiFi stations based on the 80211_hwsim wireless emulation driver and wireless drivers of the standard Linux. They append classes to enable the addition of these wireless devices in a Mininet network scenario and to emulate a mobile station’s attributes such as position and motion relative to the access points [51].

Mininet-Wifi [52],[53] is a wireless network emulator as well that supports SDWMN by expanding the popular Mininet with wireless channel emulation and WiFi APIs support. The emulation of wireless networks has to implement node mobility, signal propagation, among other features, to allow experiments with environments that have interference, signal attenuation, etc [54]. We can choose among many wireless mobility models and propagation as well as arbitrary topologies and wireless network scenarios, including the emulation of (i) ad hoc and (ii) infrastructure wireless modes. At the essential part of the emulator is a virtualization of 802.11 Linux drivers using mac80211_hwsim, the software simulator of 802.11 radio(s). The hostapd daemon is integrated in Mininet-Wifi to allow user space software access points able of making common Network Card Interface into access points. The wireless channel emulation is performed by using the parameters (e.g. node distance) of the chosen propagation model to re-configure in real-time. The Linux Traffic Control (TC) parameters (e.g., bandwidth, packet loss, and delay) of the virtual interfaces [55],[51].

To emulate Software-Defined Wireless Networks, the base code of Mininet should be extended by adjusting scripts and classes to support wireless services and functionalities whereas keeping all Software Defined Networking abilities from the standard Mininet network emulator [56].

8.2. Test machine properties

The laptop that has been used to perform the tests is of the specifications shown in Table 2.

| Operating system | CPU              | Frequency | RAM  | Hard disk     |
|------------------|------------------|-----------|------|---------------|
| Ubuntu 18.04.4   | Intel® CoreTM i5-4210M | 2.60-GHz  | 4 GB | 256 GB SSD    |
|                  |                  |           |      | 500 GB        |
8.3. Design UAV Network using Mininet-WiFi

Mininet-WiFi and python script have been used to create SDWMN as UAV network that consists of five access points mounted on five UAVs at 200-meter height in a square area of (3km x 3km) and fifteen stations as user equipment’s (UE) shown in Figure 6.

A mapping between UAV network elements and Mininet-WiFi elements is done in Table 3. The UAVs and UEs were assumed static for testing the controller. After that run the python script and connect the topology with the controller, run pingall command to check the connectivity of nodes in the network tests, and obtain the results.

| Table 3. Mapping between UAV network and Mininet-WiFi |
|-----------------------------------------------|
| **Mininet-WiFi** | **UAV Network** | **Height (meter)** |
| AP1              | UAV1            | 200               |
| AP2              | UAV2            | 200               |
| AP3              | UAV3            | 200               |
| AP4              | UAV4            | 200               |
| AP5              | UAV5            | 200               |
| Stations 1-15    | UEs             | Ground            |

| Table 4. Network parameters for the network in Figure 7 |
|-----------------------------------------------|
| **Parameter** | **Value** |
| Mode          | IEEE 802.11g |
| Frequency     | 2.4 GHz |
| Propagation mode | Log distance, exp=2.5 |
| Tx power of mesh | 25 dBm |
| Tx power of Wlan | 20 dBm |
| Mesh channel  | 5 |
| Wlan channel  | 1 |
8.4. NETWORK SIMULATION TOPOLOGY
The network simulation topology used as UAV network is demonstrated in Figure 7.

![Figure 7. Network simulation topology](image)

8.5. UAV NETWORK EMULATION
The procedure to do the emulation includes:

- Program the network in Python language that simulates UAV network,
- Run the Python script in Mininet-WiFi that emulates the UAV network,
- Connect it with (Pox, Ryu, OpenDaylight, or Floodlight) controllers each at a time which operates as a simple forwarding switch that creates a forwarding table in UAVs based on mac address table to forward the packets like a routing table.

8.6. STEPS OF THE EXPERIMENT
1- Run python script as a UAV network in Mininet-WiFi and view the network topology in Mininet-WiFi GUI as shown in Figure 8. There are five UAVs, fifteen UEs, all are proposed as stationary for testing the controllers as shown in Figure 8.
Table 5. Positions of nodes distributed in area of (3km x 3km)

| Nodes   | X-axis (meter) | Y-axis (meter) | Z-axis (meter) |
|---------|----------------|----------------|----------------|
| UAV1    | 500            | 500            | 200            |
| UAV2    | 500            | 2500           | 200            |
| UAV3    | 2500           | 2500           | 200            |
| UAV4    | 2500           | 500            | 200            |
| UAV5    | 1500           | 1500           | 200            |
| UE1-UE15|                |                | Ground         |

2- Run Pox controller as simple forwarding switch as shown in Figure 9 that illustrates the forwarding.L2 learning model is running which makes the Pox work as a simple switch and the controller is connected with UAVs.
3- Check to see the UAVs are connected and administrated by the controller as the steps shown in Figure 10.

4- Using the Pingall command to check connectivity between all nodes, as shown in Figure 11. The pingall command has been executed and it shows that the connection between all nodes is ok and all packets are received with 0% dropped.
5- After running the UAV network script and connect it with the controller, the controller installs the OpenFlow table in the UAVs to be used to forward the packets and working as a routing table. In this case, the controller is running as simple forwarding switch which contains the source and destination mac addresses. Figure 12 shows the flow table of UAV1.
9. Testing the controllers
In order to select which controller to implement in this work, this test has been performed. The four controllers are tested using the ping tool. Mininet-Wifi offers tools for evaluating network efficiency, including the ping tool which is used to measure the transmission time of a specified number of packets between nodes in the programmatic network. This tool is a packet generator for testing network connectivity between two nodes and gives values of round trip time (RTT) by sending echo messages within the network using the Internet Control Messaging Protocol (ICMP). For each controller ping of 100 packets is executed between (UE1 and UE10). The distance between the two elements is measured as shown in Figure 13 which is 3405.88 meters.

![Figure 13. Distance between UE1 and UE10](image)

It is the longest distance in the network. The result of execution is summarized in Table 6 and plotted in Figure 14.

| RTT (msec) | Pox | Ryu | Floodlight | OpenDaylight |
|-----------|-----|-----|------------|--------------|
| Min       | 1.607 | 1.386 | 1.728 | 1.502 |
| Avg.      | 5.351 | 3.846 | 4.635 | 6.371 |
| Max       | 63.115 | 17.475 | 20.507 | 125.434 |
| Mdev      | 9.891 | 3.082 | 2.543 | 14.235 |
| Packet loss | 1% | 0% | 12% | 4% |

Where Mdev (moving standard deviation) is the default deviation, basically an average of how far each RTT ping is from the mean RTT. The higher the Mdev, the more variable the RTT (overtime) will be.

![Figure 14. Comparison among SDN controllers (Pox, Ryu, Floodlight, and OpenDaylight in terms of latency in (msec).](image)

According to the results listed in Table 6 and plotted in Figure 14, the Ryu controller has an average RTT (3.846 msec) and (0% packet loss). Compared with other controllers, Ryu has the lowest...
latency among all the controllers and that is a good factor to choose the controller to control the UAV network.

10. Conclusions
This paper highlights a comprehensive study on UAV networks, its service and applications. SDN paradigm, controllers and SDWMN have been discussed in details that include their characteristics with deep analysis in the literature. Mininet-Wifi Emulation tool are used to investigate four controllers (Pox, Ryu, Floodlight, and OpenDaylight). The simulation results emphasize that Ryu controller outperform other controllers in many aspects like packet loss and latency. In our future work, the aim is to emulate complex UAV networks with dense UEs and the use of Long-Term Evolution (LTE) 4G technology. Further work will be conducted to compare the performance of networks with SDN by running different applications over the SDN controller.

References
[1] P. Zhao, Y. Yang, Y. Zhang, K. Bian, L. Song, P. Qiao, and Z. Li, “Optimal Trajectory Planning of Drones for 3D Mobile Sensing,” 2018 IEEE Glob. Commun. Conf. GLOBECOM 2018 - Proc., pp. 1–6, 2018.
[2] S. Hayat, C. Bettstetter, A. Fakhreddine, R. Muzaffar, and D. Emini, “Handover challenges for cellular-connected drones,” in DroNet 2019 - Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, co-located with MobiSys 2019, 2019, pp. 9–14.
[3] O. Bekkouche, T. Taleb, and M. Bagaa, “UAVs Traffic Control Based on Multi-Access Edge Computing,” 2018 IEEE Glob. Commun. Conf. GLOBECOM 2018 - Proc., pp. 1–6, 2018.
[4] R. Lyu, J. Liu, M. Sheng, and J. Li, “Modeling and Analysis of UAV Assisted Cellular Network,” 2018 IEEE Glob. Commun. Conf. GLOBECOM 2018 - Proc., pp. 1–7, 2018.
[5] M. A. Messous, S. M. Senouci, and H. Sedjelmaci, “Network connectivity and area coverage for UAV fleet mobility model with energy constraint,” in IEEE Wireless Communications and Networking Conference, WCNC, 2016.
[6] S. Zhang, H. Zhang, B. Di, and L. Song, “Cellular UAV-To-X Communications: Design and Optimization for Multi-UAV Networks,” in IEEE Transactions on Wireless Communications, 2019, vol. 18, no. 2, pp. 1346–1359.
[7] L. Gupta, R. Jain, and G. Vaszkun, “Survey of Important Issues in UAV Communication Networks,” IEEE Commun. Surv. Tutorials, vol. 18, no. 2, pp. 1123–1152, Apr. 2016.
[8] F. Mohammed and I. Jawhar, “Efficient Data Communication in Unmanned Aerial Vehicles,” United Arab Emirates University, Al Ain, United Arab Emirates, 2015.
[9] M. Monwar, O. Semiari, and W. Saad, “Optimized Path Planning for Inspection by Unmanned Aerial Vehicles Swarm with Energy Constraints,” 2018 IEEE Glob. Commun. Conf. GLOBECOM 2018 - Proc., pp. 1–6, 2018.
[10] A. T. Albu-Salih and S. A. H. Seno, “Optimal UAV deployment for data collection in deadline-based IoT applications,” Baghdad Science Journal, vol. 15, no. 4, pp. 484–491, 2018.
[11] Yvonne Gibbs and B. Dunbar, “Unmanned Aerial Vehicles,” 2019. [Online]. Available: https://www.nasa.gov/centers/armstrong/images/UAV/index.html. [Accessed: 15-Dec-2019].
[12] RussiaToday, “Google, NASA work together to design drone air-traffic-control system,” 2015. [Online]. Available: https://www.rt.com/usa/310705-drone-air-traffic-control/. [Accessed: 17-Nov-2019].
[13] A. Guillen-Perez, R. Sanchez-Iborra, M. D. Cano, J. C. Sanchez-Aarnoutse, and J. Garcia-Haro, “WiFi networks on drones,” in Proceedings of the 2016 ITU Kaleidoscope Academic Conference: ICTs for a Sustainable World, ITU WT 2016, 2017.
[14] X. Zhang, H. Wang, and H. Zhao, “An SDN framework for UAV backbone network towards knowledge centric networking,” INFOCOM 2018 - IEEE Conf. Comput. Commun. Work., no. 61471376, pp. 456–461, Jul. 2018.
[15] H. Huang et al., “Traffic-Aware Adaptive Deployment for UAV-Aided Communication Networks,” 2018 IEEE Glob. Commun. Conf. GLOBECOM 2018 - Proc., vol. 2018-Janua, no. 3, pp. 1–6, Sep. 2018.

[16] I. Zacarias, J. Schwarzrock, L. P. Gaspar, A. Kohl, R. Q. A. Fernandes, J. M. Stocchero, and E. P. de Freitas, “Enhancing Mobile Military Surveillance Based on Video Streaming by Employing Software Defined Networks,” Wirel. Commun. Mob. Comput., vol. 2018, pp. 1–12, Oct. 2018.

[17] J. Ali, S. Lee, and B. H. Roh, “Performance analysis of POX and Ryu with different SDN topologies,” ACM International Conference Proceeding Series. pp. 244–249, 2018.

[18] M. Alharthi, A.-E. M. Taha, and H. S. Hassanein, “An Architecture for Software Defined Drone Networks,” ICC 2019 - 2019 IEEE Int. Conf. Commun., pp. 1–5, 2019.

[19] Q. Fan and N. Ansari, “Towards traffic load balancing in drone-assisted communications for IoT,” IEEE Internet Things J., vol. 6, no. 2, pp. 3633–3640, 2019.

[20] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Mobile internet of things: Can UAVs provide an energy-efficient mobile architecture?,” in 2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings, 2016.

[21] X. Zhang and L. Duan, “Optimization of Emergency UAV Deployment for Providing Wireless Coverage,” 2017 IEEE Glob. Commun. Conf. GLOBECOM 2017 - Proc., vol. 2018-Janua, pp. 1–6, 2018.

[22] M. M. Azari, F. Rosas, K. C. Chen, and S. Pollin, “Optimal UAV positioning for terrestrial-aerial communication in presence of fading,” in 2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings, 2016.

[23] A. Al-Hourani, S. Chandrasekharan, A. Jamalipour, L. Reynaud, and S. Kandepan, “Optimal cluster head spacing for energy-efficient communication in aerial-backhauled networks,” IEEE Glob. Commun. Conf. GLOBECOM 2015, 2015.

[24] M. Chen, W. Saad, and C. Yin, “Liquified state machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks,” IEEE Trans. Wirel. Commun., vol. 18, no. 3, pp. 1504–1517, 2019.

[25] M. Mozaffari, W. Saad, M. Bennis, and M. M. Debbah, Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage, vol. 20, no. 8. Institute of Electrical and Electronics Engineers Inc., 2016, pp. 1647–1650.

[26] B. Li, Z. Fei, and Y. Zhang, “UAV communications for 5G and beyond: Recent advances and future trends,” IEEE Internet Things J., vol. 6, no. 2, pp. 2241–2263, 2019.

[27] S. Ur Rahman, G. H. Kim, Y. Z. Cho, and A. Khan, “Deployment of an SDN-based UAV network: Controller placement and tradeoff between control overhead and delay,” in International Conference on Information and Communication Technology Convergence: ICT Convergence Technologies Leading the Fourth Industrial Revolution, ICTC 2017, 2017, vol. 2017-Decem, pp. 1290–1292.

[28] D. Hasan and M. Othman, “Efficient Topology Discovery in Software Defined Networks: Revisited,” Procedia Computer Science, vol. 116, pp. 539–547, 2017.

[29] “GitHub - noxrepo/nox: The NOX Controller.” [Online]. Available: https://github.com/noxrepo/nox. [Accessed: 10-Jan-2020].

[30] “Installing POX — POX Manual Current documentation.” [Online]. Available: https://noxrepo.github.io/pox-doc/html/. [Accessed: 10-Jan-2020].

[31] “Ryu SDN Framework.” [Online]. Available: https://osrg.github.io/ryu/. [Accessed: 10-Jan-2020].

[32] “pyretic/controller.py at master · frenetic-lang/pyretic · GitHub.” [Online]. Available: https://github.com/frenetic-lang/pyretic/blob/master/pyretic/vendor/ryu/ryu/controller/controller.py. [Accessed: 10-Jan-2020].
[33] “Download «Floodlight OpenFlow ControllerProject Floodlight.” [Online]. Available: http://www.projectfloodlight.org/download/. [Accessed: 10-Jan-2020].

[34] “Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions.” [Online]. Available: https://www.opennetworking.org/onos/?utm_referrer=https%3A%2F%2Fwww.google.com%2F. [Accessed: 10-Jan-2020].

[35] “Maestro - Software-Defined Networking with OpenFlow - Second Edition [Book].” [Online]. Available: https://www.oreilly.com/library/view/software-defined-networking-with/9781783984282/9d18d4a9-fa4b-40aa-8c3e-efc344812c16.xhtml. [Accessed: 10-Jan-2020].

[36] “Open MUL Foundation Home - HOME.” [Online]. Available: http://www.openmul.org/. [Accessed: 10-Jan-2020].

[37] “Home - OpenDaylight.” [Online]. Available: https://www.opendaylight.org/. [Accessed: 10-Jan-2020].

[38] K. Tantayakul, R. Dhau, B. Paillassa, and W. Panichpattanakul, “Experimental analysis in SDN open source environment,” in ECTI-CON 2017 - 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2017, pp. 334–337.

[39] I. Z. Bholebawa and U. D. Dalal, “Performance analysis of SDN/openflow controllers: POX versus floodlight,” Wirel. Pers. Commun., vol. 98, no. 2, pp. 1679–1699, Jan. 2018.

[40] C. Kalaivani, “An overview of SDN Controllers,” Int. J. Comput. Sci. Eng., vol. 7, no. 1, pp. 592–594, Jan. 2019.

[41] L. Mamushiane, A. Lysko, and S. Dlamini, “A comparative evaluation of the performance of popular SDN controllers,” in IFIP Wireless Days, 2018.

[42] T. A. Assegie, “Performance analysis of emulated software defined wireless network,” Indones. J. Electr. Eng. Comput. Sci., vol. 16, no. 1, pp. 311–318, Oct. 2019.

[43] P. Patil, A. Hakiri, Y. Barve, and A. Gokhale, “Enabling Software-Defined Networking for Wireless Mesh Networks in smart environments,” in Proceedings - 2016 IEEE 15th International Symposium on Network Computing and Applications, NCA 2016, 2016, pp. 153–157.

[44] J. F. G. Orrego and J. P. U. Duque, “Throughput and delay evaluation framework integrating SDN and IEEE 802.11s WMN,” in 2017 IEEE 9th Latin-American Conference on Communications, LATINCOM 2017, 2017, vol. 2017-Janua, pp. 1–6.

[45] U. Ashraf, “Placing controllers in software-defined wireless mesh networks,” in 2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development, iCoMET 2018 - Proceedings, 2018, vol. 2018-January, pp. 1–4.

[46] H. Elzain and W. Yang, “Decentralizing software-defined wireless mesh networking (D-SDWMN) control plane,” in Lecture Notes in Engineering and Computer Science, 2018, vol. 2235.

[47] H. Huang, P. Li, S. Guo, and W. Zhuang, “Software-defined wireless mesh networks: Architecture and traffic orchestration,” IEEE Netw., vol. 29, no. 4, pp. 24–30, Jul. 2015.

[48] S. Sadiq, F. Shaukat, and T. Mahboob, “A Survey on Wireless Software Defined Networks,” 2015.

[49] H. Mohseni and B. Eslamnour, “Handover Management for Delay-sensitive IoT Services on Wireless Software-defined Network Platforms,” in Proceedings of 3rd International Conference on Internet of Things and Applications, IoT 2019, 2019.

[50] S. Ur Rahman, G. H. Kim, Y. Z. Cho, and A. Khan, “Positioning of UAVs for throughput maximization in software-defined disaster area UAV communication networks,” Journal of Communications and Networks, vol. 20, no. 5, pp. 452–463, 2018.

[51] R. D. R. Fontes and C. E. Rothenberg, mininet-wifi-draft-manual. 2019.
[52] “intrig-unicamp/mininet-wifi: Emulator for Software-Defined Wireless Networks.” [Online]. Available: https://github.com/intrig-unicamp/mininet-wifi. [Accessed: 21-Feb-2020].

[53] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. S. Santos, and C. E. Rothenberg, “Mininet-WiFi: Emulating software-defined wireless networks,” Proc. 11th Int. Conf. Netw. Serv. Manag. CNSM 2015, pp. 384–389, Dec. 2015.

[54] R. D. R. Fontes and C. E. Rothenberg, Wireless Network Emulation with Mininet-WiFi, 1st ed. 2019.

[55] R. Dos Reis Fontes and C. E. Rothenberg, “Mininet-WiFi: A platform for hybrid physical-virtual software-defined wireless networking research,” in SIGCOMM 2016 - Proceedings of the 2016 ACM Conference on Special Interest Group on Data Communication, 2016, pp. 607–608.

[56] L. L. Zulu, K. A. Ogudo, and P. O. Umenne, Simulating Software Defined Networking Using Mininet to Optimize Host Communication in a Realistic Programmable Network. Institute of Electrical and Electronics Engineers Inc., 2018.

BIOGRAPHIES OF AUTHORS

Hussein Ali Al-Gboury Received the B.S. degree in Computer and Information Engineering from Mosul University, Mosul, Iraq in 2010. He is currently working toward the Master’s degree Computer and Information Engineering, Electronics Engineering, Ninevah University, Mosul, Iraq. His current research interests include SDN and UAV Networks. He can be contacted at hussein.mohammed@uoninevah.edu.iq

Sahar Abdul Aziz Al-Talib received her BSc. in Electrical and Communication Engineering from University of Mosul in 1976, her MSc, and PhD in Computer and Communications Engineering from University Putra Malaysia (UPM), in 2003 and 2006 respectively. She served at MIMOS Berhad, Technology Park Malaysia, for about 4 years as Staff Researcher where she enrolled in infrastructure mesh network project (P000091) adopting IEEE 802.16j/e/d wireless technology. She became a lecturer at the Department of Computer and Information Engineering, Electronics Engineering, University of Mosul, Iraq in 2011. She was the main inventor of 10 patents and published more than 20 papers in local, regional, and international journals and conferences. She certified CISCO-CCNA, WiMAX RF Network Engineer, WiMAX CORE Network Engineer, IBM “Basic Hardware Training Course”- UK, Cii-Honeywell Bull “Software Training Course on Time Sharing System and Communications Concept”- France, Six-Sigma white belt, Harvard Blended Leadership Program by Harvard Business School Publishing – USA, Protocol Design and Development for Mesh Networks by ComNets – Germany. She can be contacted at sahar.alitalib@uoninevah.edu.iq