Experiment Report: ”Performance Analysis Framework For Base Station Placement Using IEEE 802.11”

Kukartsev Kirill,Rustam Khakov, Zufar Makhmutov

March 19, 2020

Contents

1 Formal description of the experiment

In the experiment we seek to:

1. Check the functionality of the experimental software system.
   - Check if the framework matches initial functional properties.
   - Test for stability, performance, and usability in real case scenarios.
2. Evaluation of Unmanned Aerial Vehicles (having Wi-Fi Access Points) layout optimization algorithms.
   - Evaluation for the correctness of suggested positions for Unmanned Aerial Vehicles (UAVs). We suppose that the framework is intended to optimize positions for UAV-BS, but in the experiment we use Access Points (APs) placed on the ground instead.
   - Measurement of accuracy, stability, and performance of the optimization algorithms.

1.1 Main information

The main purpose of the experiment is to optimize the location of Access Points in such a way that network throughput and RSS of User Equipments (UEs) is maximized.

For that, we design our experiment in the following way:

1. AP located in the space without any obstacles. They are surrounded by UEs.
2. UEs connect to APs. RSS and throughput are evaluated via specially designed framework (combination of GPS_Tracker, GPS_Android, GPS_Frontend).
3. UEs sends messages to the central server.
4. An operator has access to a control interface, where he can analyze received data and run optimization algorithms to find out the best positions for APs.
5. Once a better-suggested position for AP is found, the AP will be moved there.
6. The experiment goes to the third step and repeats until no significant improvement regarding RSS and speed rate will be observed.

The experiment repeated three times with different initial positions. Test sets:

1. Near-Optimal (APs are equidistant from UEs according to K-Means), described in Section 5.
2. Uniform (APs has the same distance to Command Center), described in Section 6.
3. Sub-Optimal (APs are surrounded by UEs), described in Section 7.

For each test case, we expect that:

1. Minimization of the distance between UEs and APs will lead to Received Signal Strengths (RSSs) gain and throughput increase.
2. The interference effect may be visible - in the sub-optimal placements, it should decrease the throughput;

2 Experiment description

2.1 General perspective

Figure 1 shows default experiment layout. We have 3 laptops: two APs and one Command Centers. Command Center use external Wi-Fi adapter to set up wireless point for APs to access provided services. Each has two Wi-Fi adapters: one used to set up Wi-Fi access point for UEs, the second (internal, built in the vast majority of modern laptops) adapter is used for connection to Command Center.

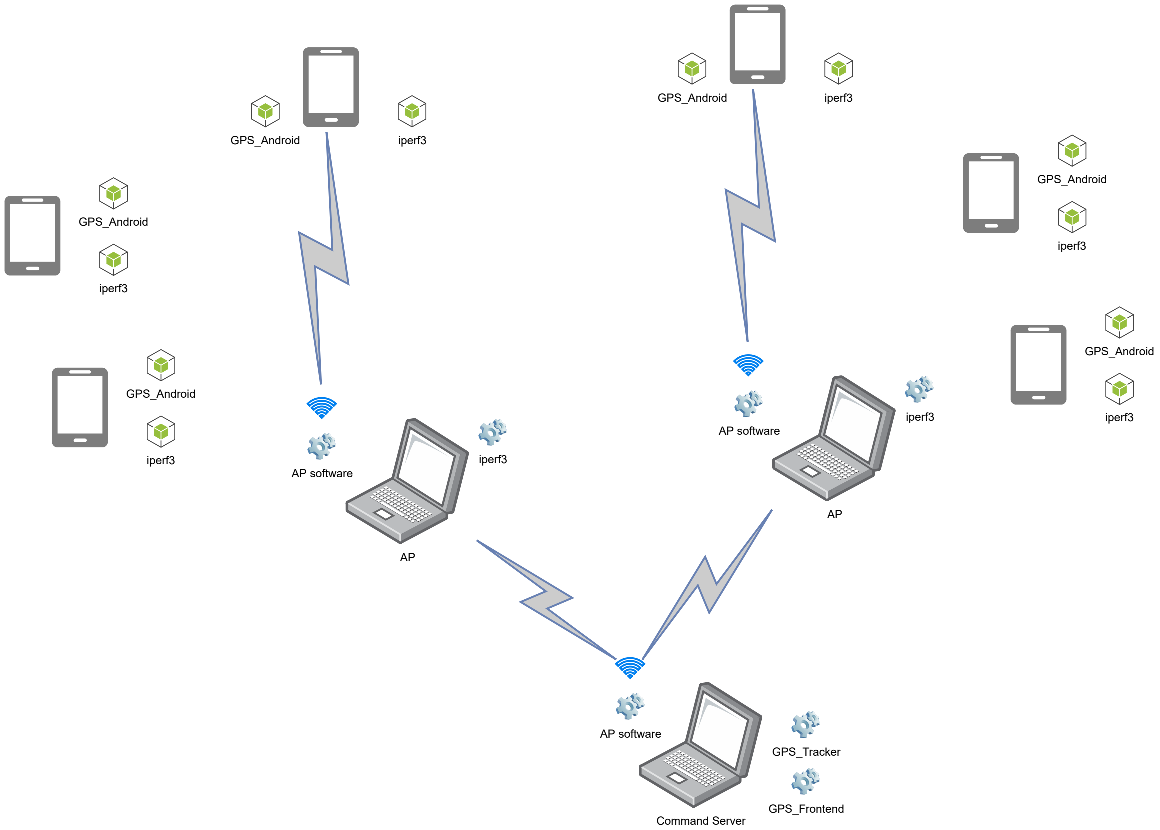

Figure 1: Main scenario of the experiment

From the general system engineering perspective, the experiment is a set of wireless-connected nodes (via Wi-Fi 802.11n (300 Mbps)) which measures the receiving signal strength and measure the throughput of the link to upload and download.

Preliminary tests in default layout showed one restrictions on experiment activity: network bandwidth between UE and measurements with `iperf3` showed about **30 MBit/s** speed rate. On the contrary, speed on the bearer UE - - Command Center showed about **12-15 MBit/s** speed rate. This is significant drop in network rate, probably, because of transmission on the radio channel two-times. The problem here is that radio link has higher error rate due to interference and fading than wired connection, so we should measure the closest radio link to the UEs (bearer UE - AP),thus FTP-server placed in AP.

Each UE has GPS_Android installed. Access Point create a Wi-Fi access point named “**ap**”. Command Center runs a wireless point named “**cnc**”.

2.2 Deployment diagram

Figure 2 shows components and their communication in the deployed phase. Each building block has a certain level of independence, therefore it possibly fixes a fragile block when the system in operating mode.

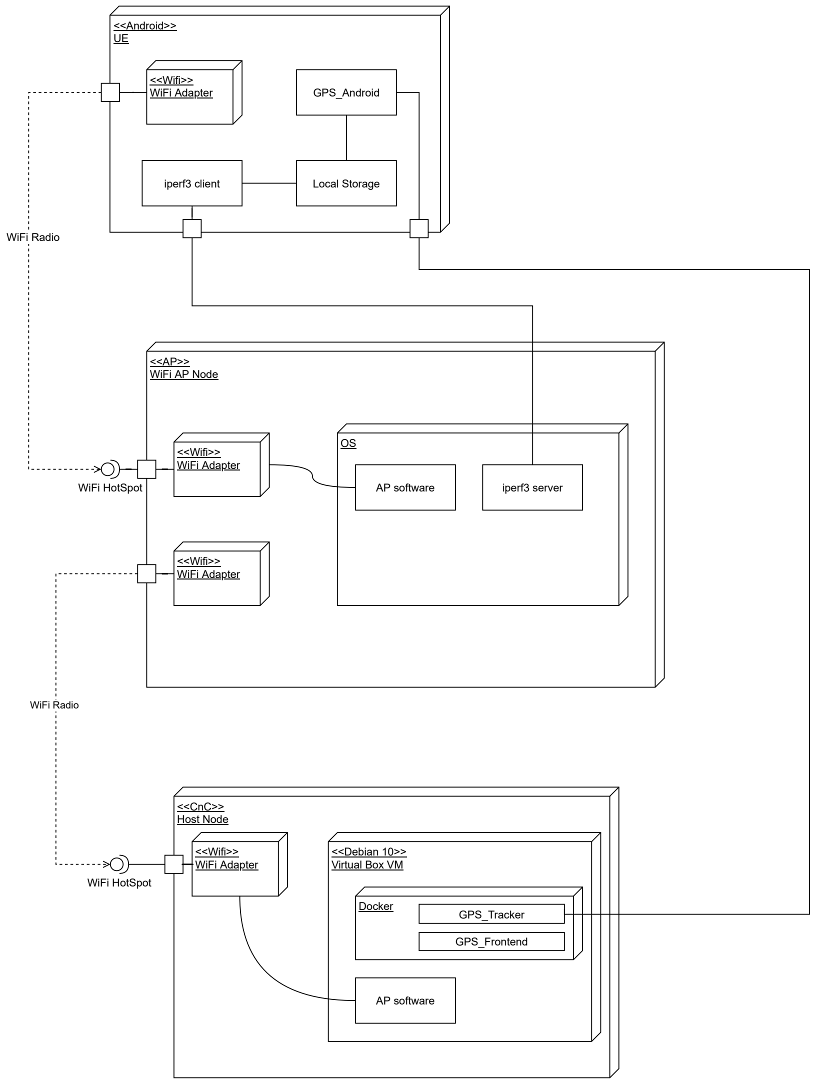

Figure 2: Deployment Diagram

We use virtual machines for running software components running in isolated containers. Virtualization helps to use the same virtual machine image on different hardware platforms. The reason to have container bundles is to simplify development, deployment and maintenance routines. The combination of virtual machines and containers significantly decreases the amount of time required to prepare the experiment environment, also this makes the installation of such a complicated software system easy-to-run.

It is possible to have performance decrease because of virtualization, however, modern processors' capabilities [1, 4] compensates those possible consequences of an additional run-time level. A hypervisor can pass-through access to physical equipment directly into the guest operating system. To automate the deployment of components, we use Ansible.

To start up a Wi-Fi access point, it requires to have two packages in OS:

- `hostapd` - software to manage and run Wi-Fi access points.
- `dnsmasq` - DNS/DCHP server, to provide an IP address, routing and DNS information via DCHP protocol.

2.3 Network diagram

Figure 3 shows network configuration. The APs subnet has identical settings, they have an internal Dynamic Host Configuration Protocol (DCHP) server to provide dynamic addresses for connected UEs. To prevent address collisions and to simplify routing, the APs perform masquerading (SNAT/DNAT) on the output interface (connection to Command Center). Consequently, Command Center cannot access the UEs directly, but UEs will always reach Command Center because DCHP server sets the default gateway IP address.

Command Center has another DCHP server to set APs dynamic addresses. When APs connect to wireless point "**cnc**" they receive a dynamic IP address by Command Center. Because network for Command Center is different from the internal networks of Wi-Fi adapters in the APs, there is no network collision.

Finally, each connected UE can access static Command Center address `192.168.20.1` and local gateway `192.168.10.1`.

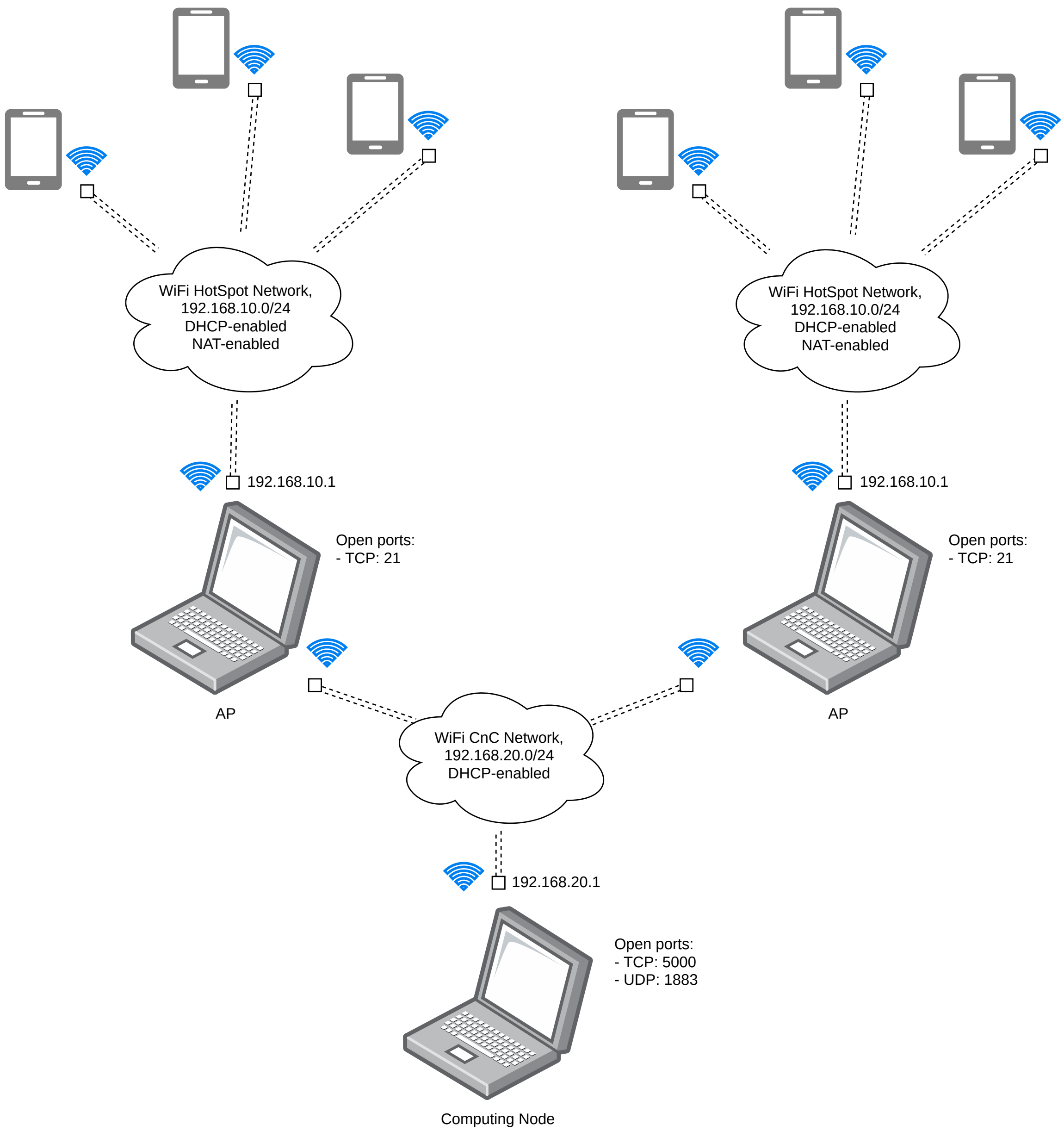

Figure 3: Network Configuration Diagram for experiment

2.4 Experiment steps

The following steps will be executed for each described experimental case:

Table 1: Steps for one experimental case.

No	Step	Description
1	Initialize network connectivity between APs and Command Center	Ensure using ICMP protocol that all nodes are available in the network
2	Start up software	Startup GPS_Tracker, GPS_Frontend
3	Place the APs and UEs according to an experiment case	There are specific predefined positions for each element on the experiment area.
4	Measure RSS, Link quality for the initial layout	Measurements are done via GPS_Android that sends the result to GPS_Tracker
5	Run the APs location optimization for an algorithm in GPS_Tracker	Each method can give different results for the same UE input set.
6	Move the APs to optimized positions.	It is expected that new coordinates for APs would increase our network efficiency.
7	Repeat RSS and link throughput measurements for the optimized APs positions	1-3 iteration per each case.

3 Experiment requirement

3.1 Hardware equipment

- 6 cellphones with Android OS (version 5.0 and above). Phones have a dedicated Global Navigation Satellite System (GNSS) module and an Wi-Fi adapter.
- 3 external Wi-Fi adapters with managed-mode capability.
  - Model: AWUS036NEH
  - Height of antenna for Command Center: 17.2 cm
  - Height of antenna for AP: 11 cm
- 3 Laptops
- One ruler

3.2 Software equipment

3.2.1 For UE

- Android-enabled smartphone (version 5.0 and above).
  - GPS_Android installed.

3.2.2 For Command Center

- The host machine
  - Any Debian-based OS (Debian, Ubuntu, etc.).
  - Oracle VirtualBox hypervisor + VirtualBox Extension Pack latest version installed.
  - Python + Ansible (for convenient configuration) installed.
  - `openssh-server` package installed.

- The virtual Command Center machine
  - `dnsmasq`, `hostapd` packages installed.
  - `openssh-server` package installed.
  - `docker` and `docker-compose` packages installed.
  - `iperf3` package installed.
    * MongoDB container.
    * RabbitMQ container.
    * MQTT Broker container.
    * Python container.
    * nginx container.
    * NodeJS container.

3.2.3 For Access Point

- The host machine
  - Any Debian-based OS (Debian, Ubuntu, etc.).
  - Oracle VirtualBox hypervisor + VirtualBox Extension Pack latest version installed.
  - Python + Ansible (for convenient configuration) installed.
  - `openssh-server`.
- The virtual AP machine
  - Any Debian-based OS (Debian, Ubuntu, etc.).
  - `dnsmasq`, `hostapd` installed.
  - `iperf3` installed.
  - `openssh-server` installed.
  - `docker` and `docker-compose` installed.
    * FTP-server container.

4 Experiment preparation procedures

These steps influence experiment parameters. We aim to take into account environmental conditions and to minimize noise components.

4.1 Preparation steps

4.1.1 Prepare hardware and software bundles

Early preparation will solve possible problems during the real experiment.

4.1.2 Install GPS_Android on UE

Before the experiment, the software must be already installed on Android phones to ensure compatibility.

4.1.3 Check the weather condition

Specific environment condition (such as snow or rain) may damage equipment and disrupt experiment results.

4.1.4 Check the RSS attenuation

This parameter will influence the size of the area used to place APs and UEs.

4.1.5 Measure the experiment area and place UE on the field

In case if RSS is high enough (level of highness evaluated by expert method) there is no reason to use a smaller area because the radio link quality would remain the same.

5 Near-Optimal layout

Figure 4 shows Near-optimal case. Here Access Points are placed in the ground so that both of them are in the center of the corresponding quarter (2nd and 3rd) 25x25 meters and far from all UEs to approximately 25m.

UEs located in two groups of three elements that fill the other two quarters (1st and 4th) respectively.

At the same time, they keep a distance from each other to limit interference and hold the same conditions.

Finally, Command Center is set in the middle of whole 50x50 meters an experimental field, so that 2 AP and 2 groups of UEs mentioned above are on the same distance.

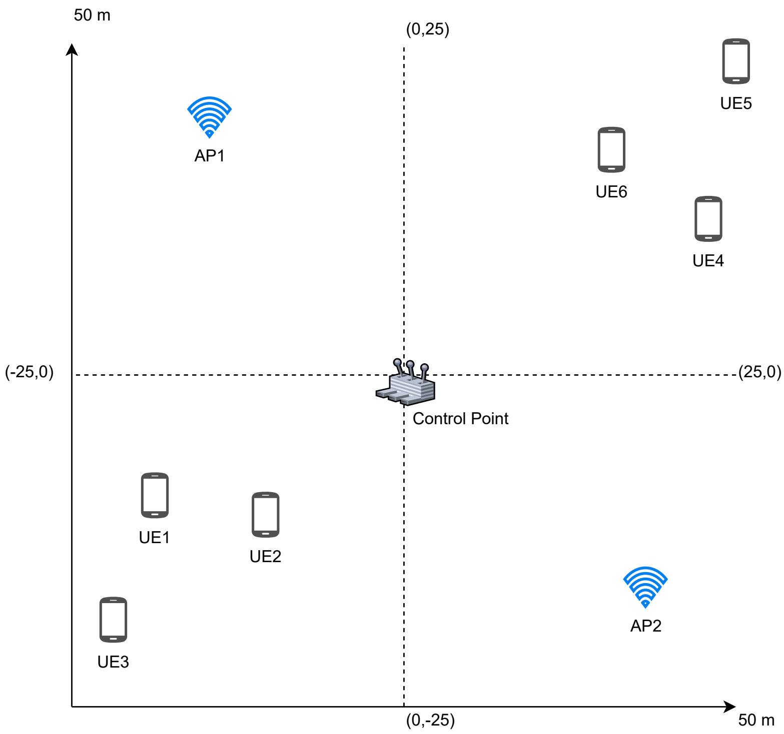

Figure 4: Near-Optimal layout

6 Uniform layout

In the uniform case layout, presented in Figure 5, we want to put APs at the same distance and line from Command Center.

Here, both APs are set right on the line between the 1st and 3rd quarters and between the 2nd and 4th quarters.

Signal quality and transmission rate in this configuration expected to increase compared with the Near-Optimal case.

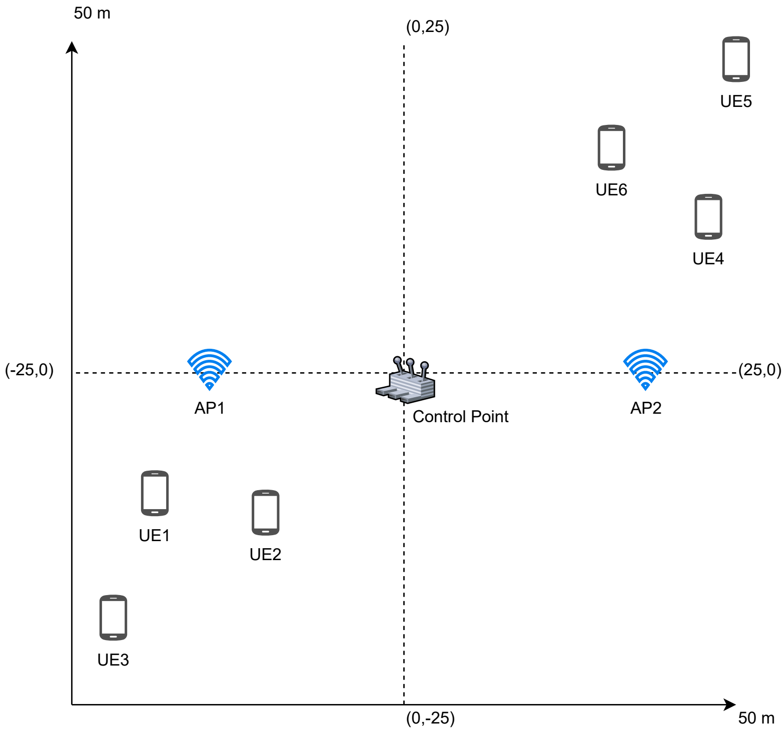

Figure 5: Uniform case layout

7 Sub-Optimal layout

Figure 6 shows Sub-Optimal layout. Each AP is in the middle of each group, therefore they have the same distance to cluster centers.

This case should have the highest signal quality and transmission rate.

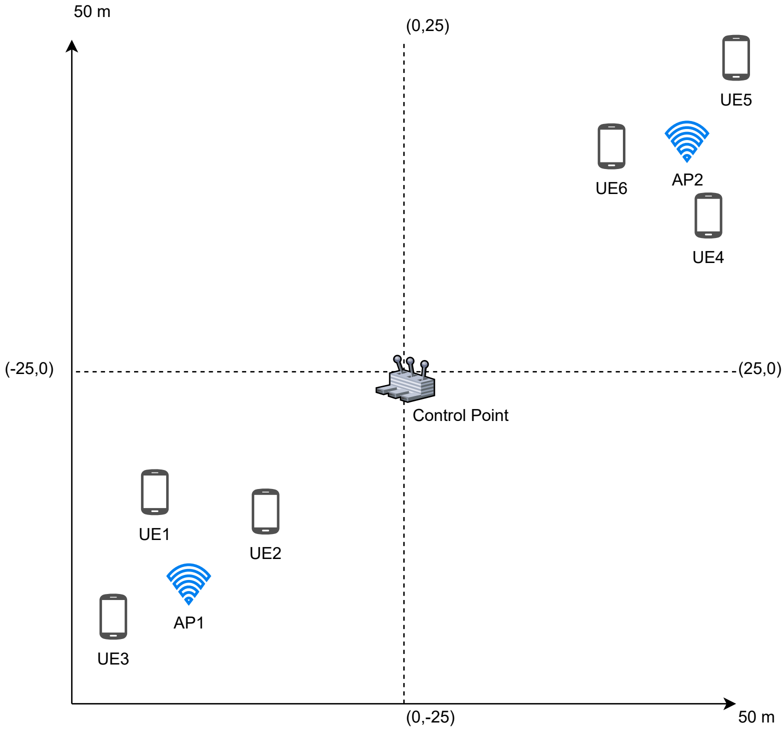

Figure 6: Sub-Optimal layout

8 Experimental phase

8.1 Experiment-1. 12.02.2020

The first attempt took place on 12.02.2020. 6 UEs took part ranging by Android version from 4 to 9.

8.2 Weather conditions

- no precipitation
- cloudy sky
- a thin layer of snow on the ground
- temperature of -1..+1°

8.3 Procedure

All items of the experiment placed on carton boxes on the ground. Figure 7 shows a picture of Command Center position during the first trial of experiment.

Figure 7: Layout of Command Center server

8.3.1 Case 1

We started with Sub-Optimal case.

One AP functioned correctly - one group of APs connection to one AP was made successfully. Here was a problem with the second group of APs - due to weather conditions we encountered Wi-Fi module issues of AP (the laptop freeze).

After some time of data collection, it was clear the second AP was not sending data to Command Center at all, so we decided to locate devices according to Near-Optimal layout scheme.

8.3.2 Case 2

During Near-Optimal layout case we observed increase of RSS level -82 and -84 dBm for the left AP to -76 and -80 dBm for the other. Link measurement messages from UEs connected to the second AP dropped by an unknown reason.

Moreover, GPS_Frontend showed that some UEs are close to each other (having approximately the same coordinate), whereas the other 2 were detected much further. It indicates that GNSSs position measurements are biased.

After, the second Access Point stopped working at some time. Restart and re-connection of UEs did not help to obtain data.

8.3.3 Case 3

Because of the problems encountered in the previous case, we decided to stop the experiment and figure out possible solutions.

8.4 Outcome

The first trial showed:

- Further development and bug fixes of GPS_Tracker and GPS_Android required.

- Tests of placement algorithms can be performed, but due to problems with message sending, we could not prove that the suggested optimal positions would lead to better signal conditions.

As a result, we decided to:

- Find out and fix the failure reasons for the second Access Point.
- Analyze the outliers in GNSS measured positions.

8.5 Experiment-2. 22.02.2020

The second attempt took place on 22.02.2020.

This time the aim was to test changes in the system components:

- A new way to estimate uplink and downlink speed added in GPS_Android using File Transport Protocol (FTP) (ability to specify IP address for Command Center (CnC) in the app)
- Logging capabilities in GPS_Android
- Improvements in GPS_Frontend (the uplink/downlink throughput measurements plot, figures became more informative).

The goal is to perform an reduced experiment with 1 Command Center, 1 AP, and 3 UEs with the same settings from the previous experiment but with modified components.

The second AP did not participate because of troubles with Wi-Fi Access Point installation drivers (later the update to Debian 11 Testing fixed this problem).

8.6 Weather conditions

- No precipitation
- Clear sky
- No snow
- Strong winds

8.7 Procedure

Only Near-Optimal case was performed.

To exclude possible bottleneck because of additional wireless bearer between AP and Command Center, we tried to measure the direct connection to `cnc` in short distance (shown in Figure 8).

Figure 8: Direct connect to Command Center

All UEs can connect to Command Center successfully:

- deviceId assigned
- UE coordinates displayed

During the first half of the experiment, after pressing the ‘push once’ button we received **uplink/downlink** throughput measurements, however, the values of uplink seemed to be too high Wi-Fi standard (300 000 kBit/s) compared to downlink (1000-2000 kBit/s).

Later, pressing ’push once’ button again did not cause speed re-estimation, the messages from UEs did not reach Command Center. The log journal did not contain any error messages.

8.8 Outcome

The second attempt was also not successful. We had not managed to solve the problem in measured data sending. For the next iteration, we proposed to implement some design improvements.

As a result, we decided to:

- Add logging capabilities to APs.
- Implement direct Hyper Text Transfer Protocol (HTTP) requests and retire Message Queuing Telemetry Transport (MQTT) broker architecture.

9 Experiment 3. 07.02.2020

It took place on 27.02.2020. We aimed to check how **2 major updates** of the Android app behave.

The *first* update concerns the usage of HTTP requests instead of MQTT. We suspected that MQTT-protocol is the reason of previously detected issues.

The *second* update is to refactor the source code. It includes switching to Model-View-ViewModel (MVVM) architectural pattern, removing of insignificant interface components.

9.1 Procedure

Due to the bad weather conditions (heavy snowfall), we decided to perform experiments indoor (Mensa). Mensa is a 2-floor building in TU Ilmenau, it has a large area inside.

The layout included 1 Command Center, 1 AP, and 3 UEs took part.

All UEs can connect to Command Center successfully:

- 'Push once' button pressed
- deviceId assigned
- UE coordinates displayed

9.2 Consideration during the experiment

As for 'Push continuously', it should be known in advance the coordinates updated on the display **only in the case of moving to some minimal delta** (10 centimeters). This is insured based on GNSS values passed by the Android device.

To make sure the connection is still alive and the values are transferred, we checked the log journal periodically.

9.3 Outcome

To sum up, the system finally started working without failures and the meaningful set of data is collected, as shown in the following figures.

Figure 9: Movement trajectory of connected phones

Figure 9 shows movement trajectory of connected clients.

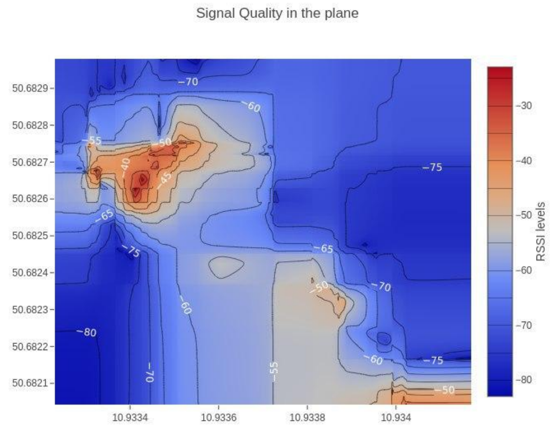

Figure 10: Signal quality map

Figure 10 shows how signal changes in different location inside Mensa. Due to obstacles (concrete walls, metal objects), the signal rapidly decreases in the farthest corners and on the second floor.

Figure 11 describes changes in RSS while UEs were moving inside the building. It is clear to observe that some phones started transmitting later than others. If a UE fails to send a measurement message, it stores in a local database and resends once the connection is back.

Figure 11: Signal quality changes

10 Experiment 4. 03.03.2020

It took place on 03.03.2020. The weather conditions were appropriate for the experiment, although there was raining lightly.

For the fourth trial, we implement sending of measurement messages only with HTTP protcol from UE to Command Center.

We aimed to find optimized positions for the UEs in three cases (Sub-Optimal, Uniform, Near-Optimal).

We decided to reduce the area layout to 25x25 meters.

We use one smartphone with a special program that can precisely locate the position. Then we placed the phone where UEs and APs were, which helped us to find out the initial positions. These coordinates for UEs are shown in Table 2. We use `decimal degrees` for coordinates.

Table 2: Initial coordinates for UEs

Unique ID	Coordinates
f072f812f48ce468	50.6823458, 10.94051806
eb0b54819c69cf0c	50.6824625, 10.9403367
27349a2cde6592df	50.6822786, 10.94057
51336504999bc1ca	50.6822861, 10.940590
b1c225280d0ed13f	50.6824403, 10.940343889
abb76773bdee4fa0	50.6824789, 10.940261944

The initial coordinates for APs are shown in Table 3.

Table 3: Initial coordinates for APs

AP	Coordinates
AP1	50.6823778, 10.94054861
AP2	50.6824281, 10.940313611

10.1 Results

The HTTP protocol helped to receive messages reliably, despite there were connection failures, we observed that the farther UE is located from AP, the less is probable reception of the message. Retransmission of messages implemented in GPS_Android partly mitigated losses.

We found out that network speed measurements were not significant, because there was markedly seen the difference between `uplink` and `downlink` speed, uplink tests threw timeout exception in case of the larger distance between a UE and a Access Point, because the communication took longer and the session terminated.

The experiment is divided into four parts:

- Before 11:05 - Sub-Optimal case
- 11:05 - 11:08 - Uniform case
- 11:08 - 11:12 - Near-Optimal case
- 11:12 - 11:15 - Sub-Optimal to compare the suggested positions

10.1.1 Sub-Optimal case

The first case is the most profitable from the signal quality point of view. The APs are implicitly located at the same distance from the connected UEs. Signal changes can be seen in Figure 12 - despite this case is expected to have the best link parameters, RSS is unstable for UEs, possible reason for this - differences in phone generations, e.g. modern phones also include more advanced Wi-Fi model.

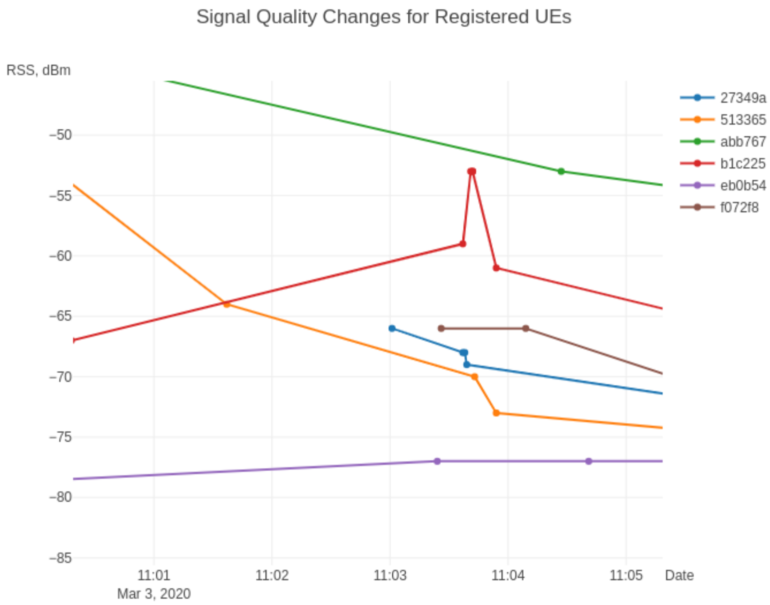

Figure 12: Signal Quality changes in Sub-Optimal case

Despite the APs were located close to UEs, the RSS level varies noticeably. However, only in this case the speed and signal quality measurement were the most stable among all cases.

10.1.2 Uniform case

In this case, the APs are located with equal distance from the Command Center.

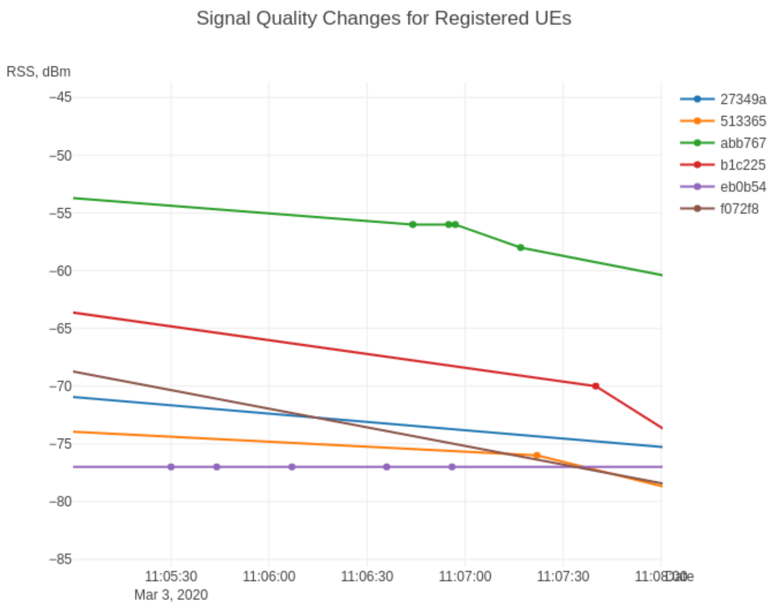

Figure 13: Signal Quality changes in Uniform case

Figure 13 depict RSS changes in Uniform case. Link measurement is more steady and keeps on the same level for the majority of UE, however, we encountered speed test failed.

10.1.3 Near-optimal case

The third case simulates the situation where the APs are placed uniformly far from centers of UEs clusters.

Figure 14 demostrates that measured RSS lower than in Uniform case because of larger distance between APs and UEs approximately on 10-15 dBm.

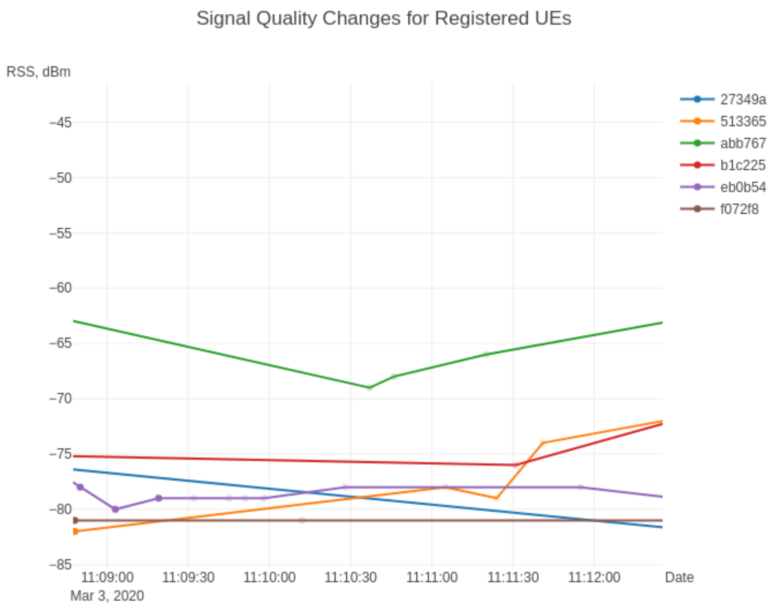

Figure 14: Signal Quality changes in Near-optimal case

10.1.4 Comparison between real coordinates in Sub-Optimal case layout and real positions positions

To check the validity, we place the APs in a Sub-Optimal case position. Figure 15 represents the most recent positions of UEs as input data used for optimization tasks. The exact coordinates are presented in Table 4.

Table 4: The last received coordinates for UEs by GPS_Android

Unique ID	Coordinates
27349a2cde6592df*	50.68206, 10.94044
51336504999bc1ca*	50.68206, 10.94044
f072f812f48ce468*	50.6823, 10.9406
b1c225280d0ed13f*	50.68243, 10.94035
abb76773bdee4fa0*	50.6823, 10.93987
eb0b54819c69cf0c*	50.68253, 10.93984

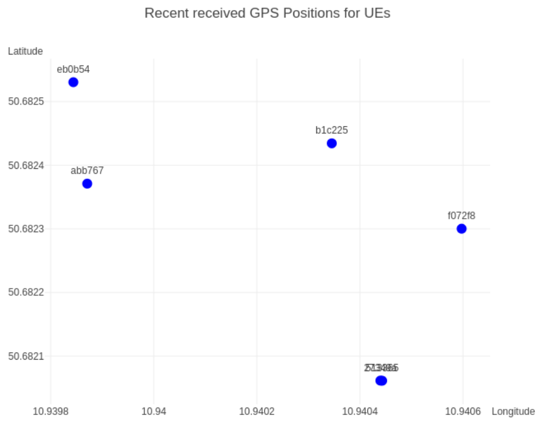

Figure 15: The UEs coordinates used to optimize APs positions

The coordinates for **27349a2cde6592df** and **51336504999bc1ca** overlap.

The most recent received coordinates for UEs used to schedule an optimization task with the following parameters:

- Number of clusters: 2
- Estimation method: “clustering”

The “simplex” method against two clusters is not possible. Instead we use ’clustering’ method which uses K-means clusterization algorithm. Figure 16 demonstrate positions for UEs as blue dots and suggested coordinates for two APs as black box.

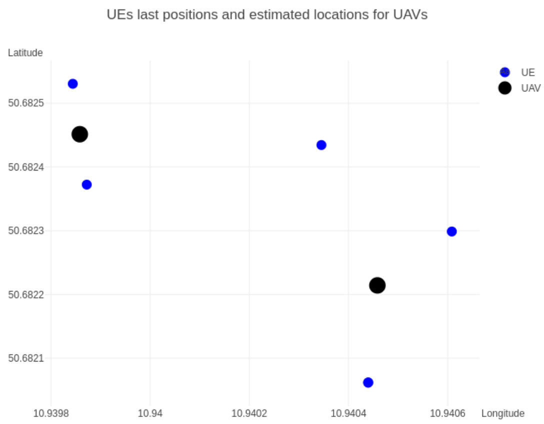

Figure 16: Result of optimization task to obtain optimal positions for APs

The suggested optimal coordinates for APs are shown in Table 5.

Table 5: Suggested optimal positions for UAVs

AP	Real coordinates	Suggested coordinates
AP1	50.6823778, 10.94054861	50.68221, 10.94046
AP2	50.6824281, 10.940313611	50.68245, 10.93986

Figure 17 shows on the map real and suggested optimal coordinates for AP. These new positions can drop out connected clients due to large distance because at this space connection became unstable, their error rate is higher.

This figure contains four types of elements:

1. Green circles - represent initial coordinates for UEs.
2. Red circles - represent coordinates for UEs received by GPS_Android.
3. Yellow tags - represent initial coordinates for APs.
4. Blue tags - represent suggested optimal coordinates for APs.

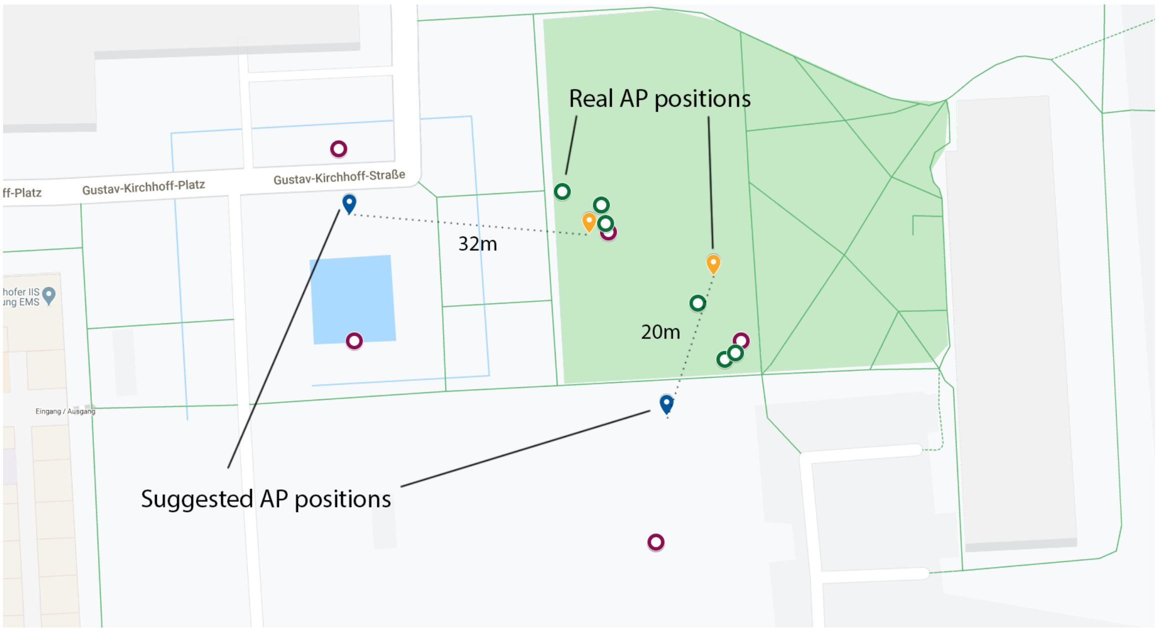

Figure 17: Original and Optimized Coordinates: A map with tags.

Distance between real and suggested positions for AP1 is 32m and 20m for AP2 accordingly. We find out two possible reasons for this difference:

1. Fragmentation of used Android phones - in the experiment we used various gadgets running different Operating System (OS) version. They present several generations of Android evolution. The older generations can have less accurate GNSS module, therefore provide a wrong position estimate.

2. Location provider optimization in GPS_Android - GPS_Android was designed to choose between network-assisted and GPS-assisted location estimation. The exact option depends on what provider has better connection status. First, it tries to set up GPS provider, if it fails, the phone fallback to network-provided location estimation which is less accurate.

Figure 18 depict these coordinates on the map from a satellite point of view.

Figure 18: Original and Optimized Coordinates for UAVs: A map from satellites.

"Clustering" algorithm performs simple K-means cluster calculation based on GNSS coordinates for UEs. The result provides insights about that layout optimization algorithms relying solely on the coordinate input data can result in a biased solution.

10.2 Outcome

Finally, we have managed to test the framework and estimate an optimized position for UAVs which were represented by Wi-Fi access points. The results showed that the framework match requirements, however, should be redesigned and evaluated on updated data.

11 Final Results

We have completed 4 experimental trials. We encountered problems in each experiment attempt. To sum up, they all relate to:

- Wi-Fi connection: radio link may suffer serious collisions that influence data transmission.
- Architecture flaws: the initial proposed design did not match the given functional requirement and worked incorrectly. Further, we redesign the architecture. It allowed us to run an experiment correctly, however, there are still gaps to fill in.
- Optimization algorithms: they should go through a formal evaluation to check their correctness on test data.

Despite these problems, we can conclude that the developed framework can be applied for performing UAVs layout optimization, the current version has enough capabilities to provide access to investigate and analyze the measured data. The algorithms included in the current version of the framework have restrictions, simplified and produce biased results, but the contrary can be highly modified to obtain more reliable results. The framework extensibility allows us to include more algorithms to optimize UAV positions.

Glossary

Access Point A laptop with an external Wi-Fi adapter.. 4, 7, 8, 10, 12, 16, 17, 23, 31

Android A modern operation system for mobile phones developed by Google. 5, 6, 15, 19

Ansible Ansible is a simple, but powerful, server and configuration management tool [2]. 5, 6, 9

Broker A broker component coordinates communication of requests from client to server and also coordinates returning the results from server to client [3]. 5

Command Center A central server where the system core is running. All measurement analysis and UI interactions perform here.. 4, 5, 7, 8, 10–13, 16–19, 22, 24, 33

GPS_Android An Android application to send the telemetry information about the current mobile network from the users' phones to GPS_Tracker.. 5, 8, 12, 17, 23

GPS_Frontend A web-based application to provide user interface to interact with GPS_Tracker.. 11, 16, 17

GPS_Tracker A name of the framework and as well a name for the backend software running for processing and storing telemetry data. 11, 12, 17, 31

MongoDB A NoSQL database storing data in BSON (Binary JSON) format [5]. 5

nginx is an HTTP and reverse proxy server, a mail proxy server, and a generic TCP/UDP proxy server [6]. 5

NodeJS Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine [7]Facc. 5

. 5, 6

RabbitMQ An in-memory queuing message broker [8]. 5

Unmanned Aerial Vehicle Is an aircraft without a human pilot on board and a type of unmanned vehicle.. 4, 32

User Equipment A mobile terminal which transmits data via the radio link.. 4, 32

Wi-Fi Wi-Fi a family of wireless networking technologies, based on the IEEE 802.11 family of standards, which are commonly used for local area networking of devices and Internet access. [9]. 4, 5, 7–10, 16–18, 23, 29, 30

Acronyms

AP Access Point. 4–8, 10–14, 16–19, 23–27, 33

DCHP Dynamic Host Configuration Protocol. 10

DNAT Destination Network Address Translation (NAT). 10

DNS Domain Name System. 10

FTP File Transport Protocol. 6, 8, 17

GNSS Global Navigation Satellite System. 5, 16, 17, 19, 29

HTTP Hyper Text Transfer Protocol. 18, 19, 22, 23

ICMP Internet Control Message Protocol. 11

IP Internet Protocol. 10

MQTT Message Queuing Telemetry Transport. 5, 18, 19

MVVM Model-View-ViewModel. 19

NAT Network Address Translation. 32

OS Operating System. 5, 6, 9

RSS Received Signal Strength. 4, 6, 12, 16, 21, 23–25

SNAT Source Network Address Translation (NAT). 10

UAV Unmanned Aerial Vehicle. 4, 26, 27, 29, 30, 33

UE User Equipment. 4, 6–8, 10, 12, 15–19, 21–27, 29, 33

References

[1] *AMD-V*. URL: https://www.amd.com/en/technologies/virtualization.

[2] Jeff Geerling. *Ansible for DevOps: Server and configuration management for humans*. 2015. ISBN: 098639341X.

[3] Robert Hanmer. *Pattern-Oriented Software Architecture For Dummies*. 1st. For Dummies, 2013. ISBN: 1119963990.

[4] *Intel Virtualization Technology*. URL: https://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html.

[5] Shannon Bradshaw Kristina Chodorow Eoin Brazil. *MongoDB: The Definitive Guide*. 3rd. O'Reilly Media, Inc. ISBN: 9781491954454.

[6] *nginx Online Documentation*. URL: https://nginx.org/en/.

[7] *NodeJS Online Documentation*. URL: https://nodejs.org/en/.

[8] *RabbitMQ Online Documentation*. URL: https://www.rabbitmq.com/documentation.html.

[9] Pejman Roshan and Jonathan Leary. *802.11 Wireless LAN Fundamentals*. 2010. ISBN: 1587142244.

List of Figures

List of Tables