Quality of Service Assessment of Live Video Streaming with a Remote-Controlled Drone

Frank Loh, Florian Wamser, Tobias Hoßfeld, Phuoc Tran-Gia
University of Würzburg, Institute of Computer Science, Germany
{frank.loh, wamser, hossfeld, trangia}@informatik.uni-wuerzburg.de

***Abstract*—Today's networks require a deep understanding of applications to optimize networks, efficiently design networks, and meet traffic demands, application heterogeneity, and application requirements. Current application areas include live video streaming and real-time applications, such as those that are named in 5G use cases with automation, disaster recovery, gaming, and Industry 4.0. In this work, we examine an application scenario with live video streaming and parallel real-time requirements in the uplink for disaster recovery. We study the quality of service (QoS) features of a remote-controlled drone. The drone is controlled via a tablet or smartphone while the video from the camera is transmitted from the drone to the user. There are high demands in both the uplink and downlink direction. The contribution of the work is the measurement of the QoS and application parameters for this scenario and the definition of influencing parameters for the application-layer.**

I. Introduction

The continuous development of new communication technologies and devices has led to ever more sophisticated and diverse applications in the networks. Network providers are facing challenges due to the diversity, heterogeneity, application needs, and data traffic of new applications, typically addressed by new network paradigms such as Software-Defined Networking (SDN). An in-depth understanding of the applications is therefore of vital interest to satisfy users and meet the requirements.

The new application areas include, among others, live video streaming applications and real-time applications as required in the field of automation, gaming, or in autonomous driving [1]–[3]. In this paper, we examine a subset of these applications. As an example of live video streaming with real-time constraints, we study the Quality of Service (QoS) features of a drone with live video streaming. Our drone is controlled via a tablet or smartphone while the video from the camera is transmitted from the drone to the user. There are high demands in both the uplink and downlink direction due to the amount of data of the video streaming and the real-time requirements for steering the drone.

978-1-5386-4633-5/18/$31.00 ©2018 IEEE

Such applications with live streaming and real-time control are mentioned in the use cases considered for current 5G mobile communications under the term ”disaster recovery”, ”critical services and infrastructure control”, and ”critical control of remote devices” [4].

The contribution of this work is threefold. On the one hand, we list influencing factors for the drone application and compare them to conventional streaming. The differences are highlighted and discussed. Second, we present measurement results in different directions about application and QoS parameters with a remote-controlled drone. We use an edge cloud-based service that allows to outsource key features to the cloud close to the user such as in [5]. Third, we present conclusions for optimizations and future work.

Therefore, this work is structured as follows. In Section II, related work is summarized. In Section III, general background information about the scenario and the remote-controlled drone is given. The measurement methodology and the testbed is introduced and described in Section IV. A description of the conducted study and the evaluation of the results are given within Section V. In Section VI optimizations for a good device to drone communication are pointed out. Finally, conclusions are drawn in Section VII.

II. Related Work

In this section, related work about unmanned drone controlling is summarized covering first of all new arising challenges with the evolution of drones in the Internet of Things (IoT) context. Afterwards, since this work deals with live video streaming and QoS together with interactive controlling and its latency, existing work with this focus are presented.

In [6], an overview of new challenges and issues going along with the increasing number of drones at the market is presented. Especially the integration into the IoT context is of high relevance. Next to the generated traffic in the form of a live video presented in this work, other traffic generated by the drone operation, like sensor data, software for updates or communication to the ground control station is named. Several arising challenges like machine-to-machine and device-to-device communications are listed, while a deeper look at the performance and trade-offs of the latter one is shown in [7].

Since this work deals with the transmission of video between a remote controlled drone and the user's device, especially the wireless channel and its utilization between both end devices is essential. In [8], unmanned aerial vehicles (UAVs) are examined with respect to the user-perceived quality of experience (QoE) for wireless devices in a cloud radio access network. User information like visited locations, requested content, job, or device type are cached, evaluated, and improved by a novel algorithm based on a machine learning framework. Based on these results, UAVs are deployed pro-actively in order to minimize the transmit power and increase user's QoE, by predicting each user's content request distribution and mobility pattern with limited information on the states of the users and the available network. Another approach is based on a placement problem of UAVs like discussed in [9]. There, a placement algorithm is presented to maximize the number of covered users.

In [1]–[3], live video streaming and real time streaming are described as use cases to be tackled for standardization. Especially in [2], requirements and architectures for a future softwarized 5G network are outlined. Current technology and research within wireless communication are introduced.

With already developed adaptive HTTP-based video streaming protocols like DASH [10], new challenges like fast-mobility resulting in fluctuating connection quality can hardly be tackled. One way to solve this problem is presented in [11], by dynamically adapting the video rate to meet drone-based application requirements. Based on inertial sensors and GPS coordinate information, the video resolution and bit rate are adapted. Additionally, context-aware compression is used to only transmit relevant video portions. Compared to them, in this work influencing factors for high QoS in a drone based video streaming approach are summarized and quantified. The goal is the creation of a realistic testbed for a broad pilot study in a drone to client communication.

III. Background

In this section, a fundamental background required for this work is provided with an introduction about typical application areas for a drone, a typical drone controlling scenario with all components and the characteristics of such a scenario. Additionally, influencing factors for the service quality of the controlling are introduced.

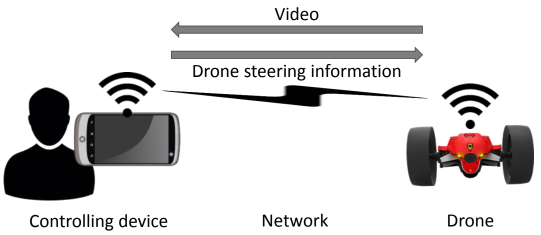


Fig. 1: Scenario with a remote-controlled drone

A. Unmanned Aerial Vehicle Application Areas

The application area of UAVs is manifold. The usage scenarios differ from private usage to commercial application. Private use cases are, for example, intruder detection, gaming, or simple photography and streaming. Especially for commercial usage the market is growing. This involves express delivery, disaster management by gathering information, weather or thermal sensors, event streaming, geographic mapping of inaccessible terrain, or deployment of drones as mobile access points. With the mass of use cases for a remote controlled drone, the importance of guaranteeing a specific QoS for such a scenario increases.

B. Drone Controlling Scenario

A simple drone controlling scenario consists of a user with a controlling device, the network where the packets between control unit and drone are sent, and the drone, see Figure 1.

Controlling Device: The controlling device is used to steer the drone. It sends steering information from the user to the drone. Additionally, the video of the drone is received and displayed. At best, the drone pilot wants to receive a video stream with *no delay* in *high quality*. Additionally, no steering commands must be *lost* or *delayed*.

Network: The network in such a scenario is a wireless connection like WiFi or bluetooth used for packet transmission. To guarantee the fast and errorfree transmission of enough packets for a smooth stream, interferences with other signals has to be avoided. Additionally, the signal strength has to be adjusted to the current scenario.

Drone: Up-to-date drones can be classified in two categories, flying and driving drones either steered by an accurate control unit only suitable for this particular

TABLE I: Factors influencing a device to drone communication

Environment	Network Properties	Device Properties
movement	delay	fps
distance	packet loss	image quality
obstacles	bandwidth	codec
	signal strength	resolution
	interference	bitrate
		errorfree steering

drone or any smart device like a smartphone, tablet, or laptop. In this work, it is concentrated on the latter ones. Such a drone, presented to the right of Figure 1 has a camera for video recording. Additionally, by exposing a wireless access point or a mobile base station, it can send the created video to the drone pilot. Based on that video, it is possible to steer the drone with the drone application at the user's end device. Compared to other video streaming, several different and additional factors arise characterizing a good drone to client streaming, based on the test environment, the network, and the device properties. In the following, these factors are summarized, while an overview of all are presented in Table I.

C. Characteristics of a Drone to Client Connection

Below, the characteristics of a drone to client connection are introduced and the main criteria for a good quality are presented.

Environment Properties: In terms of the scenario, the factors affecting the drone are the distance, the wireless channel, the surrounding obstacles, and the movement of the drone itself. The surrounding obstacles, unlike line-of-sight connection, affect the wireless channel and ultimately the streaming and control command transmission. The speed of the drone induces the Doppler shift for the wireless transmissions and impacts the sending of the data packets.

Network Properties: Increasing the distance between drone and end device has a direct impact on the network parameters. This results in a dropping bandwidth, higher packet loss, or delay. Especially with obstacles in the direct way of the communication, the connection might be disturbed by interferences primarily in an indoor scenario. All these properties detected in the network can be directly mapped to the received QoS of the remote drone.

Device Properties: The environment and network properties directly impact the device performance. In order to control the drone, no matter in which streaming quality, the frames per second (fps) and the errorfree steering are the important factors. Since the drone is controlled by the received video stream, low fps result in video pausing or juddering. The main factor for a good QoE in steering itself is to guarantee an errorfree steering data transmission. If packets are delayed or lost during drone controlling, information are received too late or not at all. The result is a degregated drone driving.

Thus, the main goal in a device to drone environment is to guarantee high fps and a smooth transmission of steering data. For a changing test environment, other properties at the device can be changed by adaption or real time encoding to decrease for example the required bandwidth. The main factors are image quality, codec, video resolution, and video bitrate like summarized in Table I.

IV. Methodology and Testbed

In this section, the methodology and the testbed used in the work to quantify influencing factors on a remote device to drone communication is presented.

A. Drone Streaming Testbed

The testbed created in this work is based on the softwarized edge cloud environment consisting of five components: a tablet used to control the drone, a wireless access-point, a remote controllable drone, a hardware server with a running virtual machine, and a measurement instance to capture packets. By hosting a web server instance at the edge cloud, the drone can expose its stream there. Thus, the user can connect directly to that server without the requirement of an additional application. Furthermore, the stream can be monitored in real time and bottlenecks can be detected. In the following, an overview of all its components is given.

User End Device: The user's end device in our testbed is a Samsung Galaxy Tab S2 with Android version 7.0.

Access-Point: To connect the mobile end device and the drone to the server, an access-point is required. In our case, a Fritz!Box 3390 is used.

Drone: The controlled drone in the measurement scenario is a Parrot Jumping Race Drone Max. The drone is exposing a video stream in the Motion JPEG (MJPEG) format, sent to the access-point. The protocol used for the MJPEG stream is the User Datagram Protocol (UDP).

TABLE II: Signal strength comparison

distance	1 m	25 m	100 m
avg. signal strength	-30.01 dBm	-51.54 dBm	-63.37 dBm
min. signal strength	-24.00 dBm	-46.00 dBm	-61.00 dBm
max. signal strength	-35.00 dBm	-65.00 dBm	-66.00 dBm

Server: At the central hardware server, a web server is installed, where the MJPEG frames sent by the drone are received. This web server can be accessed by the user's end device via web browser. The VM the server is running at is a headless qemu VM managed by virsh. As basis for the implementation the fog computing Extension Open-Source software called OpenVolcano [12] is used.

Measurement Instance: For traffic monitoring, a tool written in C++ for deep packet inspection (DPI) is installed at the drone, the access-point, and the server. From all packets among others, the timestamp and the packet size are captured. Additionally, it is decided whether a packet is a frame packet or not by means of a regular expression. In that way, the stream is classified into steering packets and streaming packets in real time during drone controlling.

Scenario Description: Like introduced in Table I, environment properties influencing the quality of a remote controlled drone scenario are movement, distance, and obstacles within the line-of-sight. Thus, in the following different scenarios are defined.

For the *movement* parameter, the movement speed of the drone is varied between no (0 km/h), slow (1-2 km/h), and fast (14 km/h) movement. The fast movement is the maximum speed of the used drone.

For the *distance* parameter, the distance between steering device and drone is varied between small (1 m), medium (25 m), and large (100 m) distance. The 100 m distance is used in order to have a scenario large enough making drone controlling impossible.

For the *obstacle* parameter, measurements are done with a direct connection between both end devices and a massive concrete wall in between.

B. Signal Strength Comparison

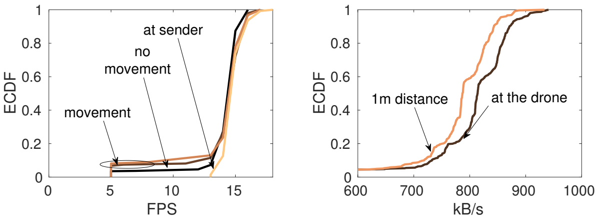
To quantify the connection between sender and receiver, the signal strength for the different distances is measured and analyzed. The result is shown in Table II, where the x-axis shows the received signal strength indicator (RSSI) in dBm, the y-axis the empirical cumulative density function (ECDF). The table shows the different RSSI values for 1 m, 25 m, and 100 m distance between a standing drone and the receiver. For 1 m distance, the result is a mean RSSI of -30.01 dBm, the maximal value is -24 dBm, the minimal is -35 dBm. The variance of the signal strength is 3.61, the standard deviation is 1.90. In the 25 m distance scenario, the mean received RSSI value is -51.54 dBm, the maximal value is -46 dBm, and the minimal one is -65 dBm. Additionally, in only 0.4 % of the measured time, the RSSI value dropped below -60 dBm. In this scenario, the variance is with 12.78 and the standard deviation with 3.58 larger than for the small distance scenario. Compared to that, having 100 m distance between sender and receiver the RSSI drops to an average value of -63,37 dBm, the maximum is -61 dBm, the minimum is -66 dBm. In more than 11 % of the measured duration the RSSI value dropped at -65 dBm or lower, for 27 % of the measured duration it is -64 dBm or lower. The variance is with 0.51 and the standard deviation with 0.71 the smallest of all presented measured distances.

V. Evaluation

According to Section III, a good device-drone communication consists of two factors, the streaming quality and the control information transmission. First, an fps comparison as the main influencing factor on the streaming quality is done. Additionally, for each scenario different movement patterns are used to evaluate the measured impact on the streaming quality. For that reason, for each scenario three different measurements are performed: one with a fixed drone, one with a constantly slow, and one with a fast moving drone. Each measurement took 15 min, creating 900 measurement points with more than 12.000 received frames for the low distance scenario and 3.000 to 9.500 for the high distance scenario.

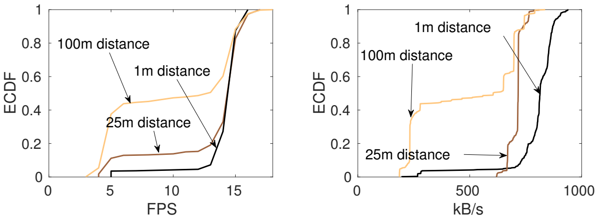
A. Small Distance to the Drone

The distance between drone and receiver in the first scenario is 1 m. The captured traffic is analyzed based on throughput and fps, shown in Figure 2 at different measurement points. Figure 2a shows the detected fps at the drone and at the receiver for different movement patterns, Figure 2b the video throughput at the drone and at the receiver with a not moving drone. In both figures, the y-axis presents the ECDF. The fps measured at the drone is 13 as minimum, 18 as maximum, and 15.17 as average, presented in yellow. The figure shows that there is no significant difference in received fps for this scenario, independent of the drone behavior. Although the drone is moving, more than 14 fps are received. This is demonstrated in Figure 2b, presenting a comparable video throughput. At the drone, an average



(a) Fps comparison (b) Video throughput

Fig. 2: Small distance scenario



(a) Fps comparison (b) Video throughput

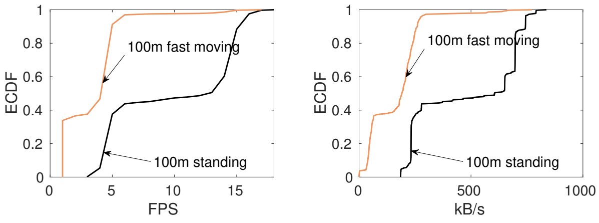
Fig. 3: Different distances

of 796.10 kB/s is measured with about 20 % of the measurement larger than 800 kB/s, shown in black. For 1 m distance to the drone the average throughput consumed by the video is 770.49 kB/s. As a result, it is shown that the fps loss is very low with 1 m distance between drone and receiver, although the drone is moving.

B. Medium Distance to the Drone

For medium distance scenarios, measurements are made with a distance of 25 m and 100 m from the drone to the user's end device. The measured fps and throughput of video data for different distances with a standing drone is presented in Figure 3. Figure 3a shows the fps comparison, Figure 3b the throughput comparison. The black line presents the result for 1 m distance between drone and receiver, the brown one for 25 m and the yellow one for 100 m distance. The axes are kept like above.

The figure shows that the distance has an influence on the received fps. With 25 m distance, in close to 20 % of the cases fps drop below 12. A large drop in fps is observed for 100 m distance. There, in around 50 % of the cases the fps drop below 12. This is a result of decreasing bandwidth shown in Figure 3b. In contrast to the 1 m distance scenario in black, the measurements with 25 m distance show an average throughput decreased to 709.05 kB/s presented in brown and 482.04 kB/s for 100 m distance, depicted in yellow. Especially for 100 m distance, the large loss in more than 40 % of the measured time is detected with less than 300 kB/s throughput.



(a) Fps comparison (b) Video throughput

Fig. 4: Different movement

TABLE III: Average fps for all Scenarios

movement	1 m distance	25 m distance	100 m distance
none	14.45	13.78	10.38
slow	14.00	13.59	5.08
fast	14.24	13.98	3.77

This is a result of lower received signal strength, introduced in Section IV. There, Table II shows the differences in the RSSI value for all scenarios. The result for an RSSI value between -25 dBm and -60 dBm is a very low frame loss. In these scenarios, the frame loss percentage is less than 10 %, with 4.75 % in the 1 m distance scenario and 9.16 % for 25 m. Thus, it is detected that up to a RSSI value of -60 dBm, the video transmission is possible with little frame loss. With 100 m distance, the average RSSI value is -63.37 dBm that results in a frame loss percentage of 31.58 % between drone and receiver. Thus, for this scenario the QoS of the video stream is bad.

Next to increasing the distance between drone and receiver, the drone movement is an influencing factor on the overall QoS. The results for 100 m distance with a standing and moving drone respectively is presented in Figure 4. The subfigure order and axes is kept like above. It is shown that additionally to the distance, the movement pattern of the drone in larger distance scenarios has an impact on the received fps like shown in Figure 4a. In more than 95 % of the measurement, less than 5 fps are received if the drone is moving. Compared to that, when the drone is standing it is in only about 40 %. This is again a direct result of the received throughput for the video shown in Figure 4b. To summarize, an overview of the average fps for all scenarios is given in Table III.

C. Obstacle in the Direct Connection

The last scenario introduced in Section IV is having an obstacle in the direct way between users end device and drone. Measurements show that the fps drop as soon as the drone is behind the obstacle to zero. In most measurements the total stream even crashed completely. Measurement with decreased quality streaming showed that it has only low influence on fps received at the sender. Thus, it can be stated that a direct connection between drone and receiving device is essential for an errorfree client drone communication.

D. Adapting Video Quality

By reacting on low bandwidth conditions between sender and receiver, one way to increase fps is adaptive streaming, for example changing the quality streaming. Figure 5 shows the differences in received fps with high and low quality streaming. The two brown lines present the results from the previous figure as baseline with no adaption. The black line depicts the amount of received fps for a standing drone with low quality streaming, while the yellow line for a fast moving drone. It can be stated that the received fps increase a lot with reducing the quality streaming for a standing drone. For the whole measurement duration, the fps are above 12. The drawback of this solution is losing video information in each frame due to the reduced quality. In the moving condition, a smaller improvement is observed. There, in only about 10 % of the measurement the fps increased.

This difference is also presented in the received throughput shown in Figure 6. Compared to the normal stream with 100 m distance and a high quality video in black, the orange line shows the throughput for the stream with the same distance but low quality. With an average video data throughput of 317.56 kB/s, the value is lower than the normal stream but the variance is a lot smaller. Thus, with worse quality a more stable stream can be established, also indicated in the fps measurements.

E. Steering Data Transmission

Next to the video stream, in Figure 7 the data stream containing the steering packets is analyzed. There, no visual difference between the data captured at the drone and the receiver with 1 m distance is detected, thus not taken into consideration in this plot. For that reason little or no loss in steering packets for small distances is assumed. There is only slightly less throughput for 25 m distance compared to 1 m at about 2 kB/s. Compared to that, for 100 m distance without changing the quality the throughput of steering data is dropping. Only between 0.5 kB/s and 1.2 kB/s in 80 % of the measurement duration is observed. This is a sign for a massive packet loss of steering packets making the drone impossible to control. If the video stream is adapted to a lower quality (lq), enough bandwidth is reserved for the steering command stream improving the throughput of it to about 2 kB/s again, presented in yellow.

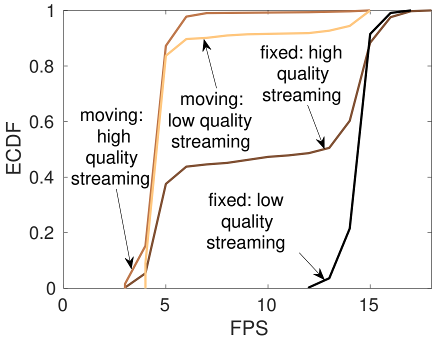


Fig. 5: Impact of quality streaming on fps

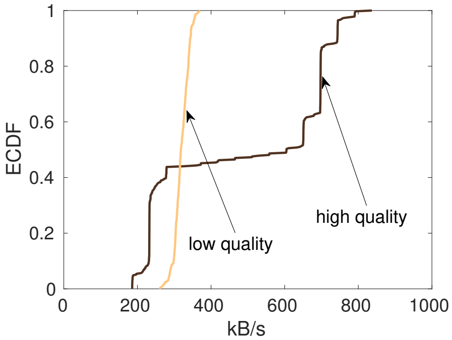


Fig. 6: Throughput of video data depending on quality

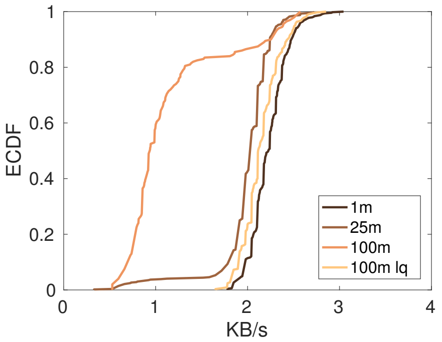


Fig. 7: Throughput of steering data

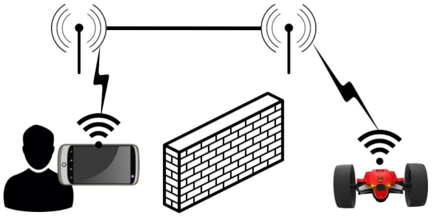


Fig. 8: Obstacle in the line-of-sight

VI. Optimization Possibilities

In this section, internal processes are described to optimize the client-drone stream for changing conditions. Specifically frequent factors for quality degradations are stated and improvement possibilities of specific parameters are introduced.

Influencing Network Properties: To tackle the problem with decreasing QoS of the drone controlling, at network side it is possible to avoid interference or increase the signal strength. The interference avoidance, especially indoor with the reflection of the own signal by walls or other obstacles can be tackled by the usage of beam-forming, directly from the access point or user's end device to the drone. Nevertheless, the most common improvement is adapting the signal strength at the current scenario. Thus, it is possible to monitor the current connection and adapt the signal. By this adaptation, it is possible to increase the end to end bandwidth and decrease the packet loss.

In a test environment with an obstacle between drone and end device blocking the radio signal, only increasing the signal strength is no solution. In such a scenario, the usage of the edge cloud is a practical improvement. By monitoring the end to end connection quality, it is possible to dynamically discover antennas with better wireless connection to the drone. In that way, the traffic can be redirected in real time to another base station connecting to the drone.

Influencing Device Properties: Next to the possibility of improving the stream at the network side, device properties, listed in Table I can be adjusted depended on the current connection quality. It is possible to use adaptive streaming, changing the video resolution, bitrate, or codec, keeping the fps high and save up bandwidth for an errorfree steering connection. In case of a very bad connection, it is also possible to adapt the frame rate, though measurements in this work show that a too massive frame rate degradation has a severe impact on the streaming.

VII. Conclusion

As an example of live video streaming with real-time constraints, a remote-controlled drone is characterized and investigated in this work. The drone is controlled via a tablet while the video is transmitted from the camera to the user. Both QoS and application parameters are measured in a test setup. A drone with live video streaming is a difficult scenario for networks due to the amount of live video data sent in the downlink direction and the real-time requirements in the uplink. The measurements carried out in the context of this work show that the packet loss and thus a decreasing fps rate and a loss in steering commands is the most serious factor influencing a client drone communication. Especially for large distances the loss increased due to a degradation of the signal strength between drone and receiver. For 25 m distance with a RSSI larger than -60 dBm in 99 % of the measured time, only a frame loss of less than 10 % occured. Having a distance of 100 m to the drone, the RSSI is smaller than -63 dBm in average, resulting in a frame loss of more than 30 %. Based on the measurement results, optimization possibilities are presented and left as work in progress. For a not moving drone, real-time encoding is already detected as approach to increase the fps at larger distances with a low signal strength but with the disadvantage that the image quality decreases.

References

[1] Ericsson, "5G systems - enabling the transformation of industry and society," Ericsson white paper, Jan. 2017.

[2] A. Osseiran, J. F. Monserrat, and P. Marsch, *5G mobile and wireless communications technology*. Cambridge University Press, 2016.

[3] N. Alliance, "Ngmn 5g white paper," *White paper, Feb*, 2015.

[4] Ericsson, "5g use cases," Ericsson white paper, 2017. [Online]. Available: https://www.ericsson.com/en/5g/use-cases

[5] F. Wamser, F. Loh, M. Seufert, P. Tran-Gia, R. Bruschi, and P. Lago, "Dynamic cloud service placement for live video streaming with a remote-controlled drone," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*. IEEE, 2017, pp. 893–894.

[6] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[7] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[8] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, 2017.

[9] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, “3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal coverage,” *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[10] T. Stockhammer, “Dynamic adaptive streaming over http–: standards and design principles,” in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.

[11] X. Wang, A. Chowdhery, and M. Chiang, “Skyeyes: adaptive video streaming from uavs,” in *Proceedings of the 3rd Workshop on Hot Topics in Wireless*. ACM, 2016, pp. 2–6.

[12] R. Bruschi, P. Lago, G. Lamanna, C. Lombardo, and S. Mangialardi, “Openvolcano: An open-source software platform for fog computing,” in *Teletraffic Congress (ITC 28), 2016 28th International*, vol. 2. IEEE, 2016, pp. 22–27.