TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik

CSR-20-07

Automated Power Line Inspection

Batbayar Battseren · Reda Harradi · Fatih Kilic ·Wolfram Hardt

September 2020

Chemnitzer Informatik-Berichte

Content

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The APOLI (Automated Power Line Inspection) project that is undertaken within the Professorship of Computer Engineering at the Chemnitz University of Technology, involves the usage of multirotor unmanned aerial vehicles (UAVs) for the sake of inspecting high voltage power lines and promptly detecting defects and flaws that generally occur at the level of the power poles or the insulators; such as burn marks, cracks, missing insulator disks, and other damages that may cause the abrupt interruption of the power transmission to private households and vital organizational and economic institutions, if not promptly and appropriately fixed.

It has been a while since UAVs (also known as drones) have been introduced in all types of inspection process due to their significant advantage in cost and time. For instance, according to the Roland Berger GmbH study, the UAV-based asset inspection achieved 90% of savings in the oil rigs, 70% of saving in the storage tanks, and 50% of saving in the wind turbine [1]. In addition to that, an automated UAV increases the inspection efficiency by eliminating the human-based factors, while providing flawless mission and high-quality real-time data. However, the term automated UAV (or autonomous) has a broad range of meanings. T. B. Sheridan [2] has mentioned the human and computer control elements aspect in undersea teleoperation in 1978 in his technical report. There are 10 levels of taxonomies defined by J. M. Beer and others [3] for human-robot interaction and by Hui-Min Huang and others [4] for unmanned systems. Especially, the Drone Industry Insights reports 6 levels of drone autonomy presented in 2019 [5].
AREIOM (Adaptive Research Multicopter Platform) research is focused on developing an unmanned system for various types of aerial inspection applications. There are several main stages, that have to be achieved step by step in order to enable the autonomous vehicle to take any action to perform a part of or an entire mission and to handle the unexpected critical situations. To realize this platform and to plan the research stages, a clear goal of the autonomy level needs to be defined. Correspondingly, these defined autonomy levels can be reflected in the system architecture in order to keep the constant progressive development.

In UAV networks, it is also important to provide reliable real-time video transmission for the mission critical applications; e.g., reconnaissance missions in military, search and rescue missions in which the viewer should be able to see what is happening in almost real-time. In APOLI project, the inspection drone flies very close to the high voltage power line. A minor mistake regarding latency can cause the system damage and failure, therefore the latency plays a critical role for the security concern as well as the video transmission quality for the inspection. Based on this approach, available video encoding methods, streaming protocols, wireless communication technologies should be investigated, and different approaches should be studied to propose a solution.

1.2 Research Areas

We categorize our research into three research focus areas due to their main focus in order to achieve a reliable UAV based autonomous inspection system. These three research focus areas are autonomy levels, network latency considerations and command and control units for UAV systems.

The paper is organized as follows: In chapter 2, the challenging aspects of the unmanned aerial vehicle based autonomous inspection system are explained. In chapter 3, the command and control units of the complete decision system are showcased. Chapter 4 introduces a comprehensive list of factors which affect video streaming quality, and briefly shows their impact in order to propose an efficient solution for improving the real-time video transmission quality by reducing the latency in UAV systems.

2 UAV-based Semi-autonomous Aerial Inspection System Development Aspects

This chapter characterizes the challenging aspects of the unmanned aerial vehicle based autonomous inspection system.

2.1 Research Focus

In this study, we propose a comparable autonomy level concept for UAS (Unmanned Aerial System), an architecture that satisfies this taxonomy. The main objective of this chapter is to address the core challenges of the autonomous system software development with an example.

2.2 Autonomy Level

Based on other's studies mentioned above and our [6] Sense-Think-Act paradigm based previous concept, we have proposed the following UAV autonomy levels for our research (Table 1). There are five distinct autonomy levels according to human and unmanned system roles.

- **Manual** (Level 1): Human is in full control of all operations, UAV is fully manual
- **Partly-automated** (Level 2): Human is responsible for the complete mission, but the UAV performs some vital functions. These are mostly flight relevant functions that are already integrated within the flight controller
- **Highly-automated** (Level 3): UAV can fly itself and execute simple missions, but the human is still in charge of safe operation. For instance, the vehicle uses waypoints to perform flight missions in a given condition and some additional functions are available
- **Semi-autonomous** (Level 4): UAV can perform a complete mission in a certain condition itself and human is responsible for monitoring the operation and acts as a back system in critical condition
- **Fully-autonomous** (Level 5): UAV performs a complete mission in any circumstances without any human assistance, for instance in an unknown state it can make a wise decision to overcome the trouble. Human is out of the loop

Table 2-1 Level of UAV autonomy

Level	Degree of autonomy	Definition	
		UAV	Human
1	Manual	fully manual	controls everything, responsible for complete operation
2	Partly-automated	performs some functions (as an assistant)	responsible for the most part of the operation
3	Highly-automated	performs most of the tasks	responsible for safe operation
4	Semi-autonomous	performs all tasks in certain condition	monitors the operation (as a back system)
5	Fully-autonomous	fully autonomous, responsible for complete operation	out of the loop

2.3 AREIOM Architecture

As mentioned before the AREIOM research's main goal is to develop an autonomous UAV-based system for different kinds of aerial inspection applications. According to the proposed UAV autonomy level, our goal is to reach a state where the UAV is capable of performing a complete inspection mission itself (Table1: Level 4), and additionally to reach a level of making own decision in real-time to run the mission in any circumstances (Table 1: Level 5). To realize this goal, we have introduced a layered architecture along with the software and hardware architectures [7]. This architecture has the following four layers, which are illustrated in Table 2:

- **Flight control layer** – is a software component, which is responsible for controlling the vehicle flight and it is located on the microprocessor where it has a direct connection to the sensors and actuators.
- **Navigation layer** – allows higher (mission and safety) layers to interact with the flight controller at an abstract level, and process the sensor values for further decision-making. It is usually implemented on the companion computer. This layer is the first facing challenge to move forward from the automation to autonomy.
- **Mission layer** – is responsible for controlling and monitoring the inspection mission by making real-time decisions based on the artificial intelligence approach. This is the second challenging part of the autonomous system.
- **Safety layer** – takes care of unexpected situations. It takes control and makes a decision to overcome any problem without any human supervision. The third and the most difficult core ingredient.

Table 2-2 Autonomy level and AREIOM layered architecture

Layers	Autonomy				
	Level 1	Level 2	Level 3	Level 4	Level 5
Flight Control	-	Intermediate	Advanced	Advanced	Advanced
Navigation	-	-	Intermediate	Advanced	Advanced
Mission	-	-	Simple	Intermediate	Advanced
Safety	-	-	-	Simple	Advanced

As shown in Table 2, the AREIOM layer implementation has presented on three different levels. If we give a numeric value to this implementation levels respectively (Simple – 1, Intermediate – 2, Advanced – 3) and add up with the number of layers parallel developing, we can roughly estimate the complexity degree for each autonomy levels (Table 3). As long as the different problems have different solutions, it is impossible to evaluate all implementations with the same scale, but the main outcome from this is to identify the amount of the research work or effort to reach the next level compared to previous levels.

Table 2-3 Approximation of the AREIOM platform development effort

Autonomy	AREIOM Layers				Number of layers	Total score	Increment
	Flight C.	Navigation	Mission	Safety			
Level 1	-	-	-	-	0	0	0
Level 2	-	-	-	-	0	0	0
Level 3	-	2	1	-	2	5	5
Level 4	-	3	2	1	3	9	4
Level 5	-	3	3	3	3	12	3

For our case, the Flight Control Layer is not included due to that we are using an already developed solution in our research. As we have seen in Table 3, the biggest step is between level 2 and level3 in the Increment column, but we can see a linear increment from level 3 to level 5 in the Total score.

2.3.1 Navigation Layer

The true intention of the navigation layer is to make the upper layers independent from the lower layer, namely the hardware layer. In the context of AREIOM platform, the open-source ArduPilot software system is used for controlling the UAV flight as a flight controller, which is presented as a Flight Control Layer in the layered architecture. To realize this concept, the MAVLink Abstraction Layer (MAL) has been developed. MAL is a software component developed in our professorship, which acts as a software-in-the-middle between the decision-making unit and the flight controller.

The second main aspect in this layer is the sensor data fusion. This includes image processing solutions for inspection object detection and localization, automatic damage assessment, and visual navigation. As long as, it is an input of the decision-making layers, it has a major role in the system and it has to be implemented with great care.

2.3.2 Mission Layer

In autonomy Level 3, most of the aerial inspections are only sequential flight actions. Therefore, this type of system can be implemented with a finite-state machine or rule-based systems. The realization is simple but capable of executing an inspection mission in a very restricted environment, and it cannot handle any unknown states. In autonomy Level 4, the system becomes more complex and covers all possible flight scenarios. It detects and avoids the obstacles, but still not capable of fully navigate itself. In autonomy Level 5, it reaches an advanced level, which can execute any type of mission and navigate through the obstacles itself in the given condition. In our research, we selected the expert system to implement AREIOM mission layer, which can imitate the human decision-making ability. The expert system will be the main decision-maker in any circumstances.

2.3.3 Safety Layer

The Safety layer is obliged to overcome the unpredicted and dangerous situations without any human instruction. In order to take the control in critical condition, it has the highest priority in the entire system. In autonomy level 4, it has a predefined routine to handle most of the risky cases. In autonomy level 5, it becomes more complex and it is capable of making an adaptive decision to the situation and learning from its own action based on advanced artificial intelligence approach. In the AREIOM platform, we have chosen the expert system to implement the mission and safety layers. It has two main rule-sets according to its responsibilities which are to run the mission and to guarantee the safe mission.

2.4 Conclusion and Future Scope

The autonomous unmanned aerial system is a new challenge and it is yet not clear how much autonomy is required. We have proposed 5 levels for the unmanned aerial vehicle autonomy and autonomy level reflected architecture. Based on our proposed concepts, it is possible to make a comprehensive plan to reach the target autonomy level. In each autonomy level, each core challenges have addressed in different priority to highlight the main difficulties. However, in our study, the core challenges are not examined in detail. Moreover, in the future study, we will address the critical software components of the autonomous system and their application-specific requirements.

3 UAV Command and Control

This chapter briefly explains the command and control units for the unmanned aerial vehicle based autonomous inspection system. Each unit of this system is introduced and explained, with the focus being set on the decision-making part that pertains the most to guiding and controlling the UAV.

3.1 Research Focus

A whole set of approaches and techniques have been used for the purpose of the above mentioned project in order to design a system that would assist the engineers and technicians solve this tedious task. The proposed system is composed from different hardware units as well as from several software parts that uniformly communicate between each other. As it is shown in Fig. 1, the architecture displays a strong link between the different software components that are running within the companion board and the Control Handler (CTH).

In fact, CTH is the central decision unit for all command and control purposes. Its tasks range from the synchronization and the supervision of the diverse communications between all the software components, to the assessment and the validation of the data packets received from the camera sensors that are used for perceiving the surrounding real world and that are feeding their output to the image processing algorithms, i.e., the Feature Point Detection (FPD) unit for the purpose of controlling the Gimbal where the inspection and navigation cameras are mounted; by the mean of the Camera Gimbal Controller (CGC) unit. The CTH is responsible as well as of providing the knowledge base that is required by the Expert System (EXS) component for firing the suitable actions to be undertaken, based on each flying scenario, before transmitting them to the Flight Controller (FLC) through the MAVLink Abstraction Layer (MAL) for their execution by the UAV [7].

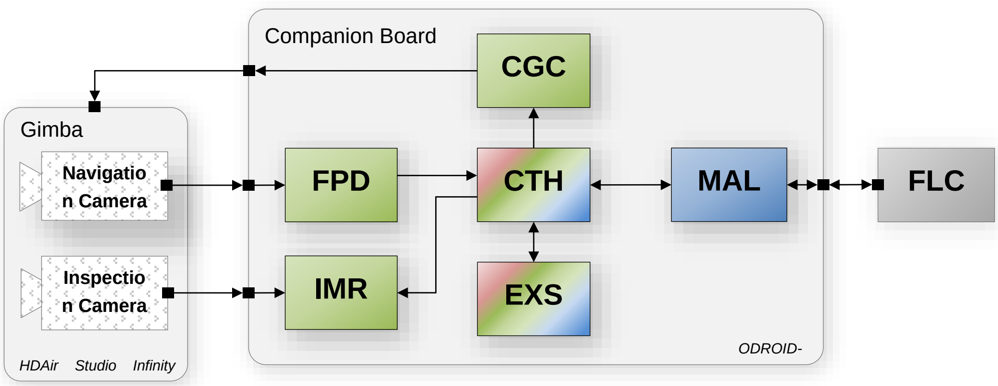

Figure 3-1 The mapping between software components and hosting hardware

As a matter of fact, the Control Handler manages the situational behavior of the UAV based on a decision-making mechanism that underlies within the Expert System.

3.2 Expert System-based UAV Controlling

The Expert System is an artificial intelligence-based inferring mechanism that relies on a C Language Integrated Production System (CLIPS) shell and that applies the RETE forward-chaining pattern matching algorithm [8], which consists of a rule-based reasoning engine that precisely distributes the production rule sets in a directed acyclic graph; i.e., the RETE network as shown in the illustration example in Fig. 2. The RETE network builds a set of independent nodes where each of which is representing a pattern that is occurring within the left-hand side of a rule and is used to assert a set of patterns that are represented by their related facts, in order to infer the possible matching of the rules that need to be fired in each match-resolve-act cycle.

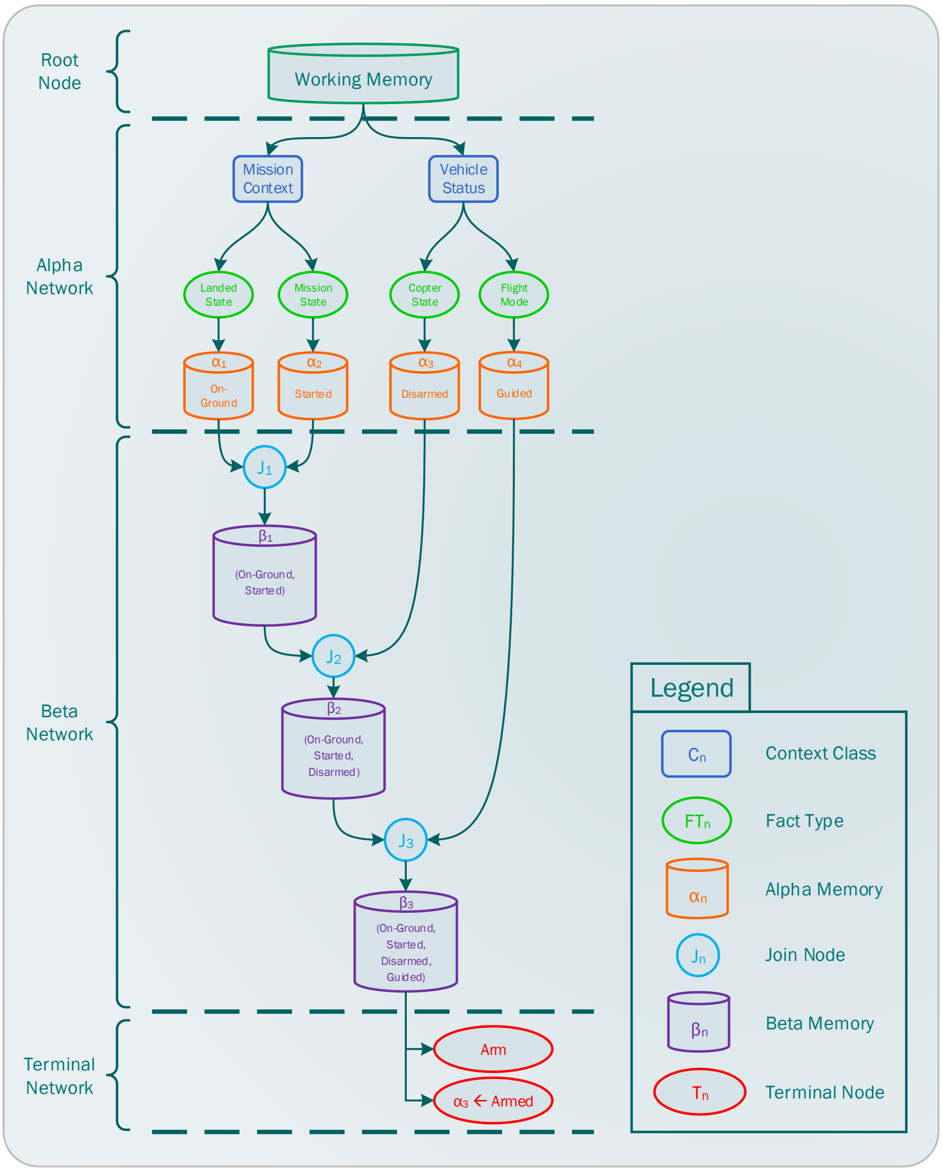

Figure 3-2 An inferring schematic for arming an UAV using the RETE network

Furthermore, the Expert System exhibits a bidirectional communication channel with the Control Handler, in such a way that all the messages that come from the other different software components are handled by the CTH. This ensures the decentralization of the overall system and makes the CTH repeatedly prepare and update the list of facts that are present in the Working Memory and that are needed for the functioning of each component.

The list of facts is then used by the Expert System's inference engine in order to activate; and store within the Conflict Set, all the terminal production rules that have been matched and whose left-hand side constraints have been fulfilled. This ensures that the right actions are taken by the UAV depending on the observed environment that is surrounding it and that is being detected by the sensors which are mounted on board. Depending on the rules which are defined within the knowledge base, the appropriate actions that are pertinent to every single flight and inspection scenario are undertaken by the UAV accordingly.

3.2.1 RETE Network – Alpha Network

The RETE network divides the graph into a set of alpha nodes and beta nodes. On the one hand, the alpha side of the graph; also called the left side, produces a discrimination chain that helps select distinct Working Memory Elements (WMEs) depending on simple conditional tests that match each WME's or a combination of WMEs' context classes and types against constant values. Each WME is passed to the following node in the graph, once it has been successfully matched against the conditions of the previous node, until it terminates at an alpha memory, where the set of WMEs that have matched every condition in each node in a given branch is stored. It is to be noted that the alpha memories are not conceived to materialize a set of WMEs that has failed to match at least one condition in a given branch.

3.2.2 RETE Network – Beta Network

The beta side of the graph, also referred to as the right side, consists on the other hand of beta nodes; i.e., join nodes, that strictly have two inputs and one output. In fact, this side of the graph is responsible of joining different WMEs as the network is being traversed and storing them in beta memories that characterize partial matches of the left-hand side conditions of a given set of production rules. Moreover, the inference engine finds all the possible matches for the WMEs that are asserted into the Working Memory during every match-resolve-act cycle, until a leaf node; i.e., a terminal node is reached when a fact or a combination of facts satisfies the patterns for a given rule to be fired.

3.2.3 RETE Network – Conflict Resolution

Once all the matches are determined, the corresponding productions are activated in the engine's agenda. The matching engine then classifies the instances of these activated rules in the agenda in a conflict set and defines in which order they should be fired following a given pre-defined control regime. Several conflict resolving strategies could then be applied for selecting the right rule to be chosen [9], based on different factors such as the saliency of the rule, which depends on its priority or its importance with respect to the flying scenario where it has been fired, the order of the rule, the recency of the rule, i.e., the time at which the facts that exist on the rule's condition part have been asserted to the Working Memory, the complexity of the activated rule, or some other pre-defined criteria that are relevant to the flying scenario to which the activated rule is pertaining.

3.2.4 RETE Network – Production Execution

After having made sure that all conflicts have been resolved and depending on the specified controlling regime, the engine fires the appropriate rules' instances which leads to executing a list of actions that are accompanying the fired rules. These actions can include, but are not limited to, running instructions to assert or retract WMEs from the engine's Working Memory, which would make the engine enter a new match-resolve-act cycle where the new updates to the WMEs would be taken into account in the newly updated Working Memory. It is also good to note that the inference engine follows the refraction principle, where every rule is fired at most only once in every match-resolve-act cycle, in order to avoid multiple instances of the same rule to be fired within any given flight scenario's inference cycle. Lastly, the Expert System's inference engine reports the fired rules to the Control Handler in every cycle and recommends then the suitable commands and packets that need to be transmitted by the MAL to the UAV's flight controller, in order to accomplish the needed UAV's maneuvers.

3.3 Conclusion and Future Scopes

This chapter mainly tackled the decision-making module that is responsible of guiding the system in assessing which actions are more appropriate to be undertaken by the UAV at any given point of time. The plan of evolving this system in using a more hybrid approach using other artificial intelligence techniques, such as machine or deep learning is part of the future scopes that are set for this project. These could assist the system in increasing the efficiency of the rules inference part in a way that would make the copter take the right decisions in a faster, more precise and thus safer manner.

4 Latency Considerations for Real-time Video Transmission in UAV Networks

This chapter briefly explains the effects of the system design on latency for real-time video transmission in the unmanned aerial vehicle based autonomous inspection systems.

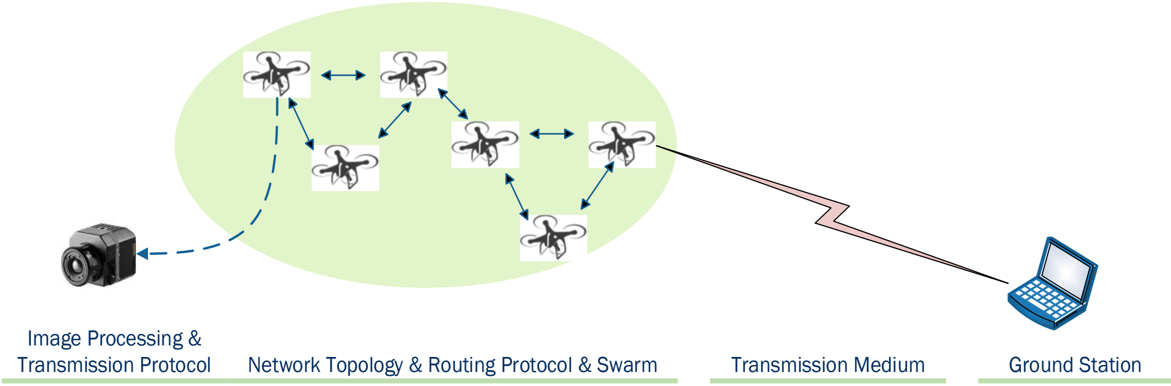

Figure 4-1 Overview of a UAV network for the real-time video transmission

Figure 4-1 illustrates the factors which are affecting the real-time video transmission latency in UAV network, and these are classified into four main groups as it follows: image processing, video transmission protocols, transmission medium (wireless link) and ground station (client side).

Table 4-1 Latency factors in UAV network

Functional domain	**Latency factors**
Image processing	Video capture (Frame rate) [11], latency during image enhancement (image rotation, resolution) [10], compression (encoding algorithms, bitrate) [10,11], buffer [10].
Video transmission protocol	*Push-based:* RTSP, RTP, UDP [24]. *Pull-based:* HTTP (HLS, MPEG-DASH, Adobe HTTP Dynamic Streaming, Microsoft Smooth Streaming) [10].
Transmission medium	*Communication technology:* Wi-Fi, WiMAX, 5G etc. [27]. *Path loss:* Interference, multipath fading, distance, transmission power, antenna technology etc. [19]. *Network Topology and Routing Protocols:* FANETs etc. [15,21,22]. *Swarm Algorithms:* Bee Ad-hoc, Ant etc. [21,22].
Ground station	Processor and graphic card (decompression and play-out buffer) [10].

4.1 Research Focus

Although the research is focused on video transmission protocols and especially communication technology for the video transmission, it is significant to consider the system as a whole and see all performance criteria. Therefore, each group affecting system latency in the system is briefly explained.

4.2 Image Processing

4.2.1 Frame Rate

The high frame rates have the lower capture times. An example of this would be a 30-fps camera takes 33ms to capture each frame of video while 60-fps reduces it to 16.5ms [11]. The delays caused in buffers will be reduced with higher frame rates [10].

4.2.2 Resolution

The higher resolution the more data to be encoded, which causes more latency [10].

4.2.3 Image Enhancements

Enhancements like rotating, scaling may also add latency [10].

4.2.4 Encoding

There are several encoding techniques available such as H.261, MPEG-1, MPEG-2, MPEG-4. are used to reduce the data rate needed for transmitting video frames [11,12]. The H.264 compression standard is a very common technique for recording and compressing video in drones [11]. According to [12], it is stated that “The video compression technics play crucial role for wireless communication applications where the bandwidth is a limiting factor. Considering H.264 video codec has the lowest bitrate compared to H.261, MPEG-1, MPEG-2, MPEG-4.”. There are several factors affecting bitrate such as compression level, resolution, frame rate, light conditions etc. [10].

4.2.5 Buffer

There are sometimes needs for short-term buffers between processing stages. These also put additional latency in the camera side [10].

4.3 Video Transmission Protocols

In [24,26] video transmission protocols have been divided into two main categories as push-based and pull-based protocols, as it was given in Table 1. In order to promote high efficiency, traditionally UDP-based (RTP/RTSP) or TCP-based (RTMP) approach are utilized for the video streaming solutions [13]. In case of huge amount of network traffic, HTTP video streaming over TCP is widely used [13]. The Real Time Streaming Protocol (RTSP) [29] is a non-connection-oriented application layer protocol that uses a session associated with an identifier. RTSP typically applies the UDP protocol to share video and audio data, and TCP for control, if necessary [24,25]. WebRTC is an API that allows browser applications to make calls, video chats and to use P2P files without any plugin [24,25]. RTP usually runs on top of UDP so that RTP has a lower transmission overhead compared to HTTP over TCP [26]. According to a complete analysis of the most used video streaming protocols, specially RTSP and WebRTC protocols are considered in [24], the use of the WebRTC protocol offers a better performance than the others.

4.4 Transmission Medium

4.4.1 Wireless Communication Technology

In [27], the comprehensive information about wireless communication standards had been given by comparing them in different criteria such as communication range, latency, throughput and robustness.

Table 4-2 Comparison of wireless communication technologies

Technology	Range	Latency	Throughput	Robustness
Wi-Fi	50-250 m	50ms	IEEE 802.11.b: 6,5 Mbps IEEE 802.11.g: 25 Mbps IEEE 802.11.n: 200 Mbps	Medium
WiMAX	5 km	25 - 40ms	1 Gbps	Low
LTE-U	2 - 5 km	9ms	1 Gbps	Low
Zigbee	500 m	20ms	250 Kbps	High
Bluetooth	10m	100ms	1-3 Mbps	Medium
Ingenu	> 2 km	Insensitive	Up to 600 Kbps	Medium
LoRa	> 2 km	low latency Class-C	100 bps	Medium
SigFox	> 2 km	Insensitive	50 Kbps	High
5G	50 km (expected)	< 1ms	50 Gbps (expected)	Very high (expected)

Considering each different criterion, the appropriate communication technology can be chosen depending on UAVs swarm application area. In cases where the high throughput is not necessary such as given in [18], when only the swarm configurations (i.e., height, distance between drones, speed, direction) are distributed to UAVs through the gateways, it was stated that the communication range of LoRa is eight times better than WiFi and provided %100 reliability. In terms of reliability and polling delay, LTE network had the highest performance compared to LoRa and WiFi. Therefore, it was suggested to use LTE network by UAV swarms when cellular network coverage is available [18].

4.4.2 Path Loss

There are several factors that limit the range of wireless video link. When the distance increases, the path loss itself will decrease the signal quality, and if there are obstacles between drone and ground station will give additional attenuation. However, in a rural area there are some less obvious challenges to the radio-link that require advanced solutions. The interference and multipath fading due to reflections are the two main issues to be considered [19,29]. Interference occurs by other sources of radio transmission in the environment which resulting in a noisy video image and limited range of the link, and if it is powerful, it can be eliminated by Typical High-Power

Blockers which “[…] can be radars, broadcast towers or military radios. Technical measures for handling interference can be to ensure good front-end channel filtering of the video receiver and use a directive ground antenna to minimize interference from other directions.” [19]. Multipath fading due to reflection causes sudden link dropouts and symbol delay spread [19,20]. Using MIMO (Multiple-Input Multiple-Output) antenna design, the multipath fading problem can be solved with the help of constructive combination of reflected signals [19].

4.4.3 Network Topology and Routing Protocols for Multiple-UAVs

Flying Ad-Hoc Networks (FANETs) are basically form of an ad-hoc network for UAVs. It differs from Mobile Ad-hoc Networks (MANET) and Vehicular Ad-hoc Networks (VANET) where its requirements vary largely from traditional networking model [15]. FANETs ensures UAVs to connect and detach to the network frequently and this makes it best solution in most cases [17,23]. It is stated in [17] that, decentralized communication architecture is more suitable to ensure quick and robust communication between the UAVs. While FANET needs high computational power, it has various advantages over MANET and VANET in case of the following examples;

- Mobility: High compactness [14,15,23]
- Topology change: Fast [14,15,16,23],
- Radio propagation model: Line of sight in most cases [14,15,16,23]

There are several routing protocols for FANETs such as secure-based, energy-based, topology-based, swarm-based and so on [23]. The some of these are;

- Position-based: ARPAM, GLSR, MPGR, AeroRP, CRUV, P-OLSR etc. [21],
- Topology-based: OLSR, HWMP, AODV etc. [21],
- Cluster-based: URP, ULSN, ERSUAV etc. [22],
- Swarm-based: APAR, BeeAdhoc and AntHocNet etc [20,21].

Routing protocols for FANETs have pros and cons regarding to their application area. Therefore, the different routing protocols can be used for different application areas. For example;

- UAV Routing Protocol (URP) can be used for the UAV-based Internet of things (IoT) technology for agriculture to take care of crop health. [22]
- UAV-based Linear Sensor Routing Protocol (ULSN) can be used to collect data and transmit from sensors to the sink node. [22]

4.4.4 Swarm Algorithms

In multiagent systems, self-organization and complex behavior can occur, even if each agent has a simple behavior strategy. The concept of the collective behavior of a decentralized self-organized system consisting of multiple agents is described as “swarm intelligence". and “[…] it is considered an optimization method in artificial intelligence theory.” [21]. In FANETs, the UAVs should have capability to share information with other UAVs to realize multiagent task distribution. The swarm intelligence has proved their efficiency in solving the routing problems in such self-organized networks as MANET, VANET [20].

The swarm intelligence is one among sub-fields in artificial intelligence domain. Therefore, it includes several bio-inspired algorithms such as ant and bee algorithms [20].

- The Ant Colony Optimization algorithms for MANET: ARA, AntHocNet, Ant-DSR, HOPNET [20].
- The algorithms modelling bees’ behavior in wild life for MANET: BeeAdHoc, BeeIP, BeeAIS, Bee-MANET [20].

At this moment, there is no general solution for routing problems in FANETs. However, the solutions based on bio-inspired algorithms such as AntHocNet and BeeAdHoc prove to be efficient in comparison with the traditional routing algorithms such as AODV, DSDV, and DSR [20].

4.5 Client Side

Processor and graphic card

The CPU and graphic card play an important role in the client-side latency during video decoding process [10,11].

4.6 Conclusion and Future Scope

In this chapter, a detailed overview of system design considerations for real time video streaming systems was given. We briefly discussed and presented the factors which cause latency in encoding methods, transmission protocols, communication technologies. Based on the investigated results in previous sections, the following steps should be considered and implemented.

1. The appropriate encoding method such as H.264 and video transmission protocol (RTSP over UDP) should be implemented and tested.
2. The key Wi-Fi performance metrics like RSSI, latency and throughput in indoor and outdoor environment should be analyzed after the implementation.
3. If it is necessary, renewing the current Wi-Fi AP or different implementation such as Dual Channel Wi-Fi technology or the different communication system such as 5G or SDR (Software Defined Radio) solution should be considered.
4. In case of integrating swarm communication technologies into our project, swarm intelligence algorithms must be studied in detail. Additionally, the appropriate communication technology should be chosen for the swarm application, and necessary equipment should be obtained.

Furthermore, the most importantly the real-life experiments must be performed to test the efficiency of the video encoding methods, transmission protocol and communication technology presented in this chapter.

5 Bibliography

[1] Uwe Welchenhaln, “Cutting costs in infrastructure maintenance with UAVs,” January 29, 2019. Accessed on: December 19, 2019. [Online]. Available: https://www.rolandberger.com/de/Publications/Drones-The-future-of-asset-inspection.html

[2] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," The MIT Press, Cambridge, 1978.

[3] J. M. Beer, A. D. Fisk and W. A. Rogers, "Toward a Framework for Levels of Robot Autonomy in Human-Robot Interaction," Journal of Human-Robot Interaction, Vols. 3, No. 2, pp. 74-99, 2014.

[4] Hui-Min Huangi, Kerry Pavekii, Brian Novakiii, James Albusi and Elena Messinai "A Framework For Autonomy Levels For Unmanned Systems (ALFUS)," Proceedings of the AUVSI’s Unmanned Systems North America 2005, Baltimore, 2005

[5] Millie Radovic, “DRONEII: Tech Talk – Unraveling 5 Levels of Drone Autonomy,” March 07, 2019. Accessed on: December 19, 2019. [Online]. Available: https://www.droneii.com/drone-autonomy

[6] B. Battseren, U. Tudevdagva and W. Hardt, "A Finite State Machine Based Adaptive Mission Control of Mini Aerial Vehicle," Embedded Selforganising Systems, vol. 5, pp. 6-10, 2018

[7] R. Harradi, A. Heller, B. Battseren and W. Hardt. “AREIOM: Adaptive Research Multicopter Platform”, in *Proc. 14th IFOST Conf.*, Tomsk, Russia, Oct. 2019, pp. 219–223.

[8] C. L. Forgy, ”RETE: A fast algorithm for the many pattern/many object pattern match problem”, *Artif. Intell.*, vol. 19, no. 1, pp. 17–37, Sep. 1982. doi: 10.1016/0004-3702(82)90020-0.

[9] J. Halbig, A. Windisch, P. Kingsbury, N. Oswald and W. Hardt. “Development of a real-time decision-making system for certifiable autonomous unmanned systems,” in *2012 Proc. AMS Conf.*, Stuttgart, Germany, Sep. 2012, pp. 119–128. doi: 10.1007/978-3-642-32217-4.

[10] “Latency in live network video surveillance,” Axis Communications, Stockholm, Sweden,2015.[Online].Available: https://www.axis.com/files/whitepaper/wp_latency_live_netvid_63380_external_en_1504_lo.pdf

[11] B. Dennis, D. Prajakta, “Low-latency design considerations for video-enabled drones,” Texas Instruments, Dallas, Texas, USA, 2016. [Online]. Available: http://www.ti.com/lit/wp/spry301/spry301.pdf

[12] A. Ksentini, Abdelhak. Gueroui, and M. Naimi, “Improving H.264 video transmission 802.11e EDCA,” in *Proceedings 14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005.*, San Diego, CA, USA, 2005, pp. 381–386, doi: 10.1109/ICCCN.2005.1523891.

[13] J. Yoon and S. Banerjee, “Hardware-Assisted, Low-Cost Video Transcoding Solution in Wireless Networks,” *IEEE Trans. Mob. Comput.*, vol. 19, no. 3, pp. 581–597, Mar. 2020, doi: 10.1109/TMC.2019.2898834.

[14] C. Aranzazu Suescun and M. Cardei, “Unmanned Aerial Vehicle Networking Protocols,” in *Proceedings of the 14th LACCEI International Multi-Conference for Engineering, Education, and Technology: “Engineering Innovations for Global Sustainability,”* 2016, doi: 10.18687/LACCEI2016.1.S.078.

[15] Md. H. Tareque, Md. S. Hossain, and M. Atiquzzaman, “On the Routing in Flying Ad hoc Networks,” presented at the 2015 Federated Conference on Computer Science and Information Systems, Oct. 2015, pp. 1–9, doi: 10.15439/2015F002.

[16] L. Gupta, R. Jain, and G. Vaszkun, “Survey of Important Issues in UAV Communication Networks,” *IEEE Commun. Surv. Tutor.*, vol. 18, no. 2, pp. 1123–1152, 2016, doi: 10.1109/COMST.2015.2495297.

[17] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, “Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols,” in *2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, Karachi, Nov. 2017, pp. 1–9, doi: 10.1109/INTELLECT.2017.8277614.

[18] Z. Yuan, J. Jin, L. Sun, K.-W. Chin, and G.-M. Muntean, “Ultra-Reliable IoT Communications with UAVs: A Swarm Use Case,” *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 90–96, Dec. 2018, doi: 10.1109/MCOM.2018.1800161.

[19] “INTERRUPT INSIDE – Drones & wireless video,” Data Respons ASA, Høvik, Norway, 2016. [Online]. Available: https://www.datarespons.com/wp-content/uploads/2014/11/Interrupt_1_2016_web-1.pdf

[20] A. V. Leonov, “Applying Bio-Inspired Algorithms to Routing Problem Solution in FANET,” *Bull. South Ural State Univ. Ser Comput. Technol. Autom. Control Radioelectron.*, vol. 17, no. 2, pp. 5–23, 2017, doi: 10.14529/ctcr170201.

[21] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. B. Yagoubi, “A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs),” *Veh. Commun.*, vol. 10, pp. 29–56, Oct. 2017, doi: 10.1016/j.vehcom.2017.10.003.

[22] M. Y. Arafat and S. Moh, “A Survey on Cluster-Based Routing Protocols for Unmanned Aerial Vehicle Networks,” *IEEE Access*, vol. 7, pp. 498–516, 2019, doi: 10.1109/ACCESS.2018.2885539.

[23] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, Md. H. Tareque, and Md. S. Hossain, “Routing in Flying Ad Hoc Networks: Survey, Constraints, and Future Challenge

Perspectives," *IEEE Access*, vol. 7, pp. 81057–81105, 2019, doi: 10.1109/ACCESS.2019.2923840.

[24] I. Santos-González, A. Rivero-García, J. Molina-Gil, and P. Caballero-Gil, "Implementation and Analysis of Real-Time Streaming Protocols," *Sensors*, vol. 17, no. 4, p. 846, Apr. 2017, doi: 10.3390/s17040846.

[25] "Streaming Protocols: Everything You Need to Know," [Online]. Available: https://www.wowza.com/blog/streaming-protocols

[26] A. Begen, T. Akgul and M. Baugher, "Watching Video over the Web: Part 1: Streaming Protocols," in IEEE Internet Computing, vol. 15, no. 2, pp. 54-63, March-April 2011, doi: 10.1109/MIC.2010.155.

[27] N. Neji and T. Mostfa, "Communication technology for Unmanned Aerial Vehicles: a qualitative assessment and application to Precision Agriculture," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, Jun. 2019, pp. 848–855, doi: 10.1109/ICUAS.2019.8797879.

[28] Y. Zeng, Q. Wu, and R. Zhang, "Accessing From The Sky: A Tutorial on UAV Communications for 5G and Beyond," *ArXiv190305289 Cs Eess Math*, Mar. 2019, Accessed: May 07, 2020. [Online]. Available: http://arxiv.org/abs/1903.05289.

Chemnitzer Informatik-Berichte

In der Reihe der Chemnitzer Informatik-Berichte sind folgende Berichte erschienen:

CSR-14-01 International Summerschool Computer Science 2014, Proceedings of Summerschool 7.7.-13.7.2014, Juni 2014, Chemnitz

CSR-15-01 Arne Berger, Maximilian Eibl, Stephan Heinich, Robert Herms, Stefan Kahl, Jens Kürsten, Albrecht Kurze, Robert Manthey, Markus Rickert, Marc Ritter, ValidAX - Validierung der Frameworks AMOPA und XTRIEVAL, Januar 2015, Chemnitz

CSR-15-02 Maximilian Speicher, What is Usability? A Characterization based on ISO 9241-11 and ISO/IEC 25010, Januar 2015, Chemnitz

CSR-16-01 Maxim Bakaev, Martin Gaedke, Sebastian Heil, Kansei Engineering Experimental Research with University Websites, April 2016, Chemnitz

CSR-18-01 Jan-Philipp Heinrich, Carsten Neise, Andreas Müller, Ähnlichkeitsmessung von ausgewählten Datentypen in Datenbanksystemen zur Berechnung des Grades der Anonymisierung, Februar 2018, Chemnitz

CSR-18-02 Liang Zhang, Guido Brunnett, Efficient Dynamic Alignment of Motions, Februar 2018, Chemnitz

CSR-18-03 Guido Brunnett, Maximilian Eibl, Fred Hamker, Peter Ohler, Peter Protzel, StayCentered - Methodenbasis eines Assistenzsystems für Centerlotsen (MACeLot) Schlussbericht, November 2018, Chemnitz

CSR-19-01 Johannes Dörfelt, Wolfram Hardt, Christian Rosjat, Intelligente Gebäudeklimatisierung auf Basis eines Sensornetzwerks und künstlicher Intelligenz, Februar 2019, Chemnitz

CSR-19-02 Martin Springwald, Wolfram Hardt, Entwicklung einer RAD-Plattform im Kontext verteilter Systeme, März 2019, Chemnitz

Chemnitzer Informatik-Berichte

CSR-19-04 Johannes Götze, René Schmidt, Wolfram Hardt, Hardwarebeschleunigung von Matrixberechnungen auf Basis von GPU Verarbeitung, März 2019, Chemnitz

CSR-19-05 Vincent Kühn, Reda Harradi, Wolfram Hardt, Expert System for Adaptive Flight Missions, Juni 2019, Chemnitz

CSR-19-06 Samer Salamah, Guido Brunnett, Christian Mitschke, Tobias Heß, Synthesizing gait motions from spline-based progression functions of controlled shape, Juni 2019, Chemnitz

CSR-19-07 Martin Eisoldt, Carsten Neise, Andreas Müller, Analyse verschiedener Distanzmetriken zur Messung des Anonymisierungsgrades θ, Juni 2019, Chemnitz

CSR-19-08 André Langer, Valentin Siegert, Martin Gaedke, Informationsverwertung basierend auf qualitätsoptimierten semistrukturierten Datenbeständen im Wachstumskern "LEDS", Juli 2019, Chemnitz

CSR-20-01 Danny Kowerko, Chemnitzer Linux-Tage 2019 - LocalizeIT Workshop, Januar 2020, Chemnitz

CSR-20-02 Robert Manthey, Tom Kretzschmar, Falk Schmidsberger, Hussein Hussein, René Erler, Tobias Schlosser, Frederik Beuth, Marcel Heinz, Thomas Kronfeld, Maximilian Eibl, Marc Ritter, Danny Kowerko, Schlussbericht zum InnoProfile-Transfer Begleitprojekt localizeIT, Januar 2020, Chemnitz

CSR-20-03 Jörn Roth, Reda Harradi, Wolfram Hardt, Indoor Lokalisierung auf Basis von Ultra Wideband Modulen zur Emulation von GPS Positionen, Februar 2020, Chemnitz

CSR-20-04 Christian Graf, Reda Harradi, René Schmidt, Wolfram Hardt, Automatisierte Kameraausrichtung für Micro Air Vehicle basierte Inspektion, März 2020, Chemnitz

CSR-20-05 Julius Lochbaum, René Bergelt, Time Pech, Wolfram Hardt, Erzeugung von Testdaten für automatisiertes Fahren auf Basis eines Open Source Fahrsimulators, März 2020, Chemnitz

CSR-20-06 Narankhuu Natsagdorj, Uranchimeg Tudevdagva, Jiantao Zhou, Logical Structure of Structure Oriented Evaluation for E-Learning, April 2020, Chemnitz

CSR-20-07 Batbayar Battseren, Reda Harradi, Fatih Kilic, Wolfram Hardt, Automated Power Line Inspection, September 2020, Chemnitz

Chemnitzer Informatik-Berichte

ISSN 0947-5125

Herausgeber: Fakultät für Informatik, TU Chemnitz
Straße der Nationen 62, D-09111 Chemnitz