The Tactile Internet for the flight control of UAV flocks

Fabio D'Urso[1], Christian Grasso[2], Corrado Santoro[1], Federico Fausto Santoro[1], Giovanni Schembra[2]
[1]Department of Mathematics and Computer Science
[2]Department of Electrical, Electronics and Informatics Engineering (DIEEI)
University of Catania
Viale Andrea Doria, 6 - 95125 Catania, Italy

***Abstract*—This paper presents a software architecture, based on the Tactile Internet 5G network slice, to control a flock of UAVs performing a monitoring mission. In contrast to classical approaches for UAV flock control, which employ ad-hoc software running directly on board of of UAVs, we propose a solution in which flock and mission control tasks run *at the edge* of a 5G network; the architecture takes advantage of the *Tactile Internet* to implement a ultra-low latency communication link needed to send driving commands to UAVs. As a result, the deployment of computations in the edge provides many advantages, in terms of scalability and fault-tolerance, and avoids processing latencies, due to communication links, that represent an important drawback in traditional solutions.**

***Index Terms*—5G, Tactile Internet, MEC, UAV Flock Control, Virtual Function Placement and Chaining.**

I. Introduction

Unmanned Aerial Vehicles (UAV), also known as "drones", are nowadays devices widely available, and that can be bought off-the-shelf. Small UAVs can be easily obtained from sellers, and their employment ranges from toys to professional applications. A wide range of monitoring applications is now made possible with reasonable costs thanks to the availability of such small professional UAVs that can be equipped with cameras or other kinds of sensing devices.

However, one of the main drawbacks of such UAVs is their autonomy that, with the technology currently used for high capacity batteries, allows a flight time of only 15-20 minutes. This limitation impedes long-term missions and, as consequence, restricts the size of the areas to monitor. To avoid such a problem, an approach proposed by several researchers is the adoption of a *flock of UAVs* [1]–[3] that, by flying in a proper formation, can—in principle—cover an area that is n times (with n the number of drones) larger than the area covered by a single drone.

State-of-the-art solutions exploit *decentralized* approaches: each UAV is equipped with a proper hardware-software infrastructure able not only to ensure flight stabilization and control, but also to interact with other UAVs (by means of a communication technology) in order to plan all together the best path for the mission to perform. The result is an *emerging behavior* that drives the flock towards the coverage of the area to be monitored. Using a distributed/decentralized architecture is the key to ensure also load balancing and fault tolerance. Moreover, thanks to the autonomy, should a UAV fail the other UAVs can detect the event and adopt proper countermeasures.

As it is well known, any decentralized architecture that is asked to support an emerging behavior needs a proper *communication infrastructure*, otherwise entities cannot self-organize. Wireless (ad-hoc) networks can, in this case, serve for the purpose, but they suffer of two main drawbacks:

- *Limited Range.* Wireless devices used in the UAVs (as in any wireless sensor network) have a limited range, so a complete coverage of the whole flock could not always be possible (above all when the flock is large). In this case, routing protocols must be adopted thus increasing the communication latency.
- *Medium Access.* A well-known key problem of wireless ad-hoc networks is the medium access control. Even if specific protocols are used, the probability of packet collisions on the same shared transmission medium is always present, thus constituting another factor that affects bandwidth and latency.

A solution to the above issues is provided by the incoming fifth generation (5G) of cellular systems [4]–[9]. One of the main peculiarities of 5G technology is the introduction of Multi-Access Edge Computing (MEC) [10], [11], an ETSI proposal that leverages on network softwarization paradigms like Software Defined Networks (SDN) [12] and Network Functions Virtualization (NFV) [13]. Leveraging on the above paradigms, a Telco operator is able to offer application developers and content providers cloud-computing capabilities at the edge of the network to achieve ultra-low latency and high bandwidth, as well as real-time access to radio network information.

Another key element introduced with 5G and complementary to MEC is the concept of *network slicing* [14], [15], defined by the Next Generation Mobile Network Alliance (NGMN) as an independent virtualized end-to-end network allowing operators to run different deployments based on different architectures in parallel. Specifically, the term network slice refers to an instance of such a logical network using network and application function chains for delivering services to a given group of devices. An exemplary application for network slicing is the Tactile Internet, with the purpose of achieving interactions between a human or machine and

978-1-5386-4633-5/18/$31.00

physical objects in timescales of 1 ms. Extremely high reliability and security are additional requirements, nonetheless the mission-critical characteristics associated with its applications.

In this paper, we propose a software architecture based on a 5G communication infrastructure to control a flock of UAVs in their monitoring mission. The architecture is composed of three main modules: *flight control*, which is the flight stack needed to control stability, speeds and waypoints, the *flock control*, which is in charge of deciding the UAV movements to maintain a given shape of the flock, and the *mission control*, which is in charge of controlling the mission by sending proper commands to the UAVs. The above modules usually run on board the UAVs because of the strict interaction needs with the controlled UAVs. Instead, in this paper, by taking advantage of ultra-low latencies guaranteed by the 5G Tactile Internet, these modules are designed to run on the *edge of the network*, i.e. in proper servers placed either in the edge cloud deployed in the Central Office (CO) at the access point of the core network, or even in the base stations, according to the latency requirements between them and the UAVs they are controlling.

These modules also feature *replication* in base stations of adjacent cells in order to support the continuity of operations even when one or more UAVs perform the handover. Finally, some additional modules not presenting so hard requirements, can be run on remote core clouds, and this is necessary if they require very huge amounts of storage and/or computation.

The paper is structured as follows. Section II describes the reference system model. Section III introduces the proposed architecture, also presenting performance requirements for each of the interfaces between all the framework components. Finally, Section IV ends up the paper with our conclusions and some indication of future work.

II. Reference System Model

We consider, as a reference scenario, a certain area of terrain—more or less wide—to be overflown for a specific reason, e.g. aerophotogrammetry, video or IR inspection, seeding or fertilizer spreading for agriculture, etc. Here a set of UAVs is employed, with the objective of subdividing the whole area into portions, each overflown and monitored by one UAV. The way in which the area is subdivided and the parts assigned to UAVs vary on the basis of the employed approach.

Some solutions presented in the literature [16]–[18] adopt a centralized model in which a (centralized) entity performs an off-line (batch) processing in order to determine area partitions, thus assigning each partition to a different UAV that will execute the overflight in autonomy. The central entity has also the task of checking that all UAVs are working and, in case of failure, can assign the area of the faulty UAV to another (or more) working UAVs.

Other proposals [1], [19]–[21] adopt a completely decentralized approach by avoiding any central entity, and distributing the mission algorithm to the UAVs themselves. In this case UAVs cooperate by exchanging messages and self-organize in order to *(i)* form and maintain a "flock of drones"; and *(ii)* plan the optimal path to cover the area to monitor.

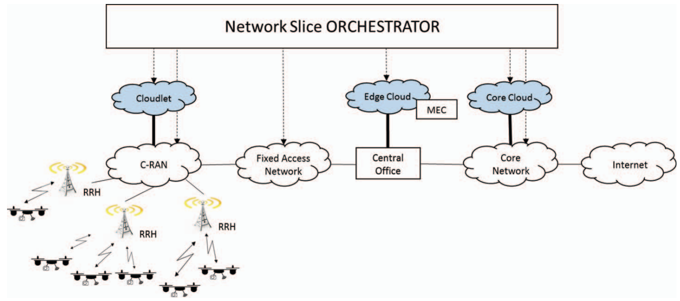

Fig. 1. Hierarchical Cloud Architecture of a 5G system.

By continuously exchanging messages, each UAV can also detect if another UAV fails (i.e. it becomes "silent") and, altogether, re-plan the path in order to re-scan areas lost by the failed entity. This kind of solutions implies that UAVs can continuously interact to each other, therefore they need a proper communication infrastructure able to guarantee certain quality-of-service parameters.

In this paper we consider the latter approach, i.e. a distributed solution to control a flock of drones belonging to an aerial monitoring platform, but leveraging on a 5G Tactile Internet slice as the communication and computation commodity. This slice provides support for ultra-low latency communications among all the elements constituting the monitoring platform considered in this paper. The physical structure of the 5G network is sketched in Figure 1. It is hierarchical from the radio access network (RAN) to the remote Internet. The RAN is realized as a Cloud-RAN (C-RAN), that is, constituted by an additional layer of small cells into the existing macro-cell network, deployed as remote radio heads (RRHs), and connected to a cloud where baseband units (BBU) are pooled in a single geographical point for a one-to-one logical mapping.

The fixed network comprises the access and the core domains. The *Fixed Access Network* interfaces the radio link with the *Core Network*. While the core domain is commonly implemented with optical transport, the access domain uses a heterogeneous set of transport technologies. However, adopting optical transport also in the access domain is a key enabler for the Tactile Internet as it offers high capacity and small propagation delays, mandatory features to achieve the requirements imposed to the Tactile Internet.

Distribution of physical resources, as Figure 1 shows, allows different deployment levels of computing resources, all open to third-party service providers to run their applications:

- The *Core Cloud*, which is a centrally located data center that hosts a large collection of processing, storage, networking, and other fundamental computing resources. On this cloud, the provider of the considered aerial video-monitoring service is allowed to deploy and run software, e.g., operating systems and applications, to realize its service. Typically, only a few core clouds are installed in a nationwide telco Operator network.
- An *Edge Cloud*, which is implemented inside an access

branch of the fixed network, closer to the end user, typically deployed inside a central office (CO). This location constitutes a good trade-off between proximity with end-users and devices, and computation and storage capacities.

- A *Cloudlet*, or *Nano-Cloud*, which is a mobility-enhanced small-scale cloud data center, usually co-located with the macro cell sites. The main purpose of the cloudlet is hosting the deployment of the Tactile Support Engine where running resource-intensive and mobile applications to provide ultra-low latency services and Artificial Intelligence (AI) to mobile devices like the UAVs we are considering in this paper.

Physical and virtual resources belonging to the underlying network infrastructure are managed and orchestrated by the *Orchestrator* entity, which is in charge of the lifecycle of all the network and application functions, their placement and their mapping on the available physical resources.

The optimal allocation of processing functions composing the aerial video-monitoring service application on the different levels of computing resources described so far will be described in the following according to the interaction requirements with the UAVs.

III. Control of UAV Flocks and Performance Requirements

As said so far, the best strategy to control UAV flock flight is by applying a distributed approach. However, difficulties in propagating information from each UAV to all the others belonging to the same flock due to the short-range communications, strongly limit the number of UAVs composing a flock, their maximum allowed speed and the UAV density in the flock.

The idea at the base of this paper, that can be realized only thanks to the usage of a 5G Tactile Internet slice, is to create a digital twin of each UAV with a Virtual Drone image running on the ground as a chain Virtual Application Functions (VAF), and timely distributed among the three different cloud levels shown in Figure 1.

As depicted in Fig.2, the proposed aerial video-monitoring service application can be realized with a number of Virtual Drones, each representing a virtual image of a physical drone participating to the monitoring mission, and a set of additional elements aimed at coordinating them and integrating their work with additional facilities. Each component of the above elements is implemented as a VAF running inside an execution container like, for example, a virtual machine, a Linux Container or a programmable embedded system. VAFs chained to realize a Virtual Drone, as well as the additional VAFs *Global Video Processor*, *Historical Video DB* and *Mission Manager*, will be described in details in Section III-A, while interfaces will be defined in Section III-B.

A. Virtual Application Functions

UAV Flight Controller. The *UAV Flight Controller* is one of the most critical parts of the system. It provides the low-level functionalities to perform control, and stabilization of a UAV and implements the overall classical flight stack by exploiting the inertials and positioning sensors that are installed on the physical UAV. Indeed, the state-of-the-art technique employed to control a UAV is to determine its *attitude* and *position*, by computing Euler angles—*roll*, *pitch* and *yaw* and their derivatives—from the data by an Inertial Measurement Unit (IMU[1]), and the global position by means of GPS and a barometer. These data are analyzed and processed by the stabilization software that implements the control loops and, in turn, outputs data for driving propellers in order to let the UAV keep the desired target attitude and position.

Since this VAF has to interact with the physical entity, it must comply with the requirements related to the dynamics of UAV control loops that, in general, must be in the order of $500\ Hz$ (i.e. $2\ ms$ of maximum duration) and requires strict real-time computations otherwise the system would result uncontrollable. This is the reason why flight stacks are usually implemented by means of embedded/microcontrollered systems that, however, even if they can provide the needed real-time requirements, usually present a not so high computation power that, on the contrary, would be really needed for data processing and sensor fusion (e.g. Kalman filters) implementation.

Flock Controller. The *Flock Controller* is in charge of implementing the Flocking formation and maintenance algorithm. We consider, in particular, the flocking algorithm presented in [1], [19], that we briefly describe here[2]. The objective is to establish a flock shape that can be optimal to perform area scan; the used approach exploits a decentralized algorithm that, on the basis of the mutual positions of the various UAVs, determines the target horizontal and angular speeds to be applied to UAVs, in order to ensure that the desired flock is maintained. This process is performed by each Flock Controller using an autonomous loop that:

1) Obtains the positions of the other UAVs by contacting the relevant Flock Controllers;
2) Executes the control algorithm according to [1], [19];
3) Sends the computed target speeds to the UAV Flight Controller in order to apply them to the physical UAV.

The dynamics of execution of the control loop of Flock Controller is in the order of $\frac{1}{50}s$ while, as for the amount of data exchanged, the Flock Controller obtains GPS position (latitude, longitude, heading and altitude) from the UAV Flight Controller and the positions of the other UAVs from peer Flock Controllers; this last set of data is dependent on the number of UAVs employed in the mission.

One of the main characteristics of the flocking algorithm is the selection of one UAV which assumes the role of the *leader*, a role particularly important for the area coverage task, which is described in the next subsection. The leader is selected by all

[1]An IMU is in general made of 3-axial gyros, accelerometers and magnetometers

[2]The reader interested in understanding the details of the algorithm can refer to the cited bibliography.

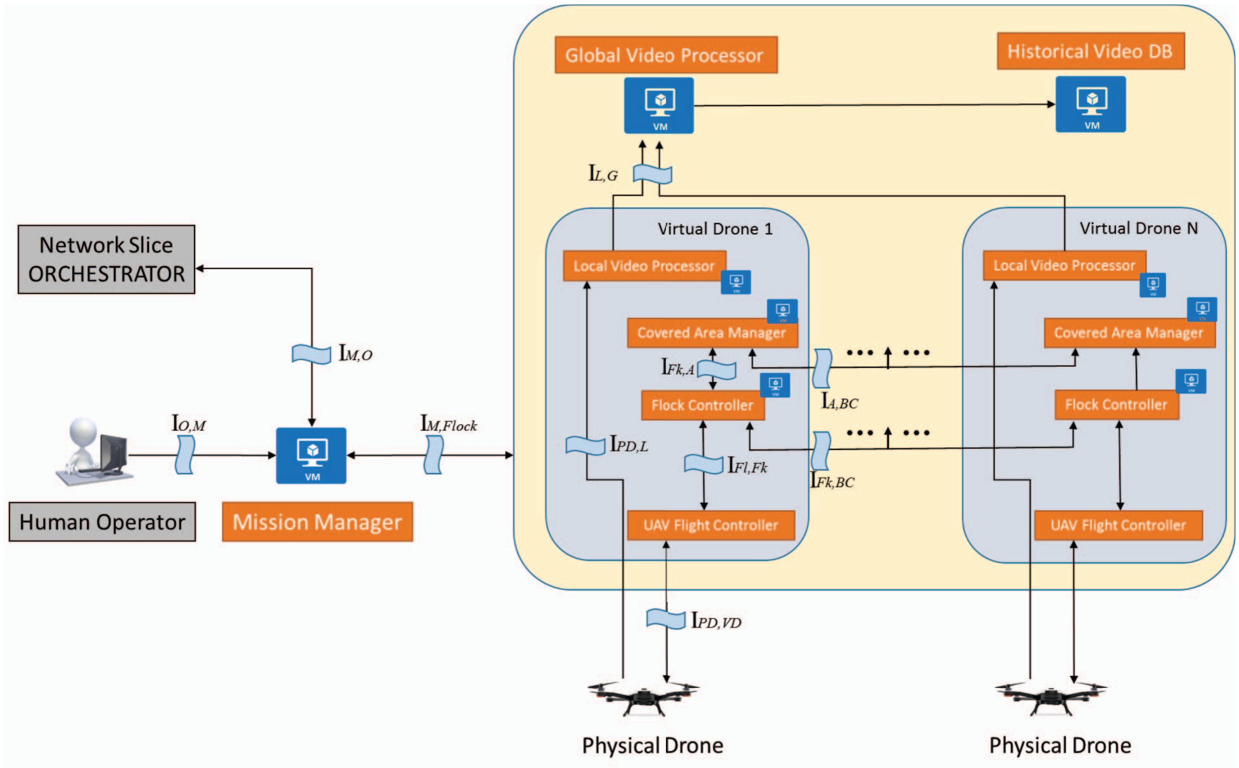

Fig. 2. Virtual Infrastructure of the Proposed Architecture.

UAVs using a simple mechanism: since it is assumed that each UAV has a *unique ID*, the leader is selected as the UAV with the *lowest ID* according to a given metric. This choice ensures that, without requiring forms of negotiation or election, all UAVs will select, in autonomy, the same leader.

Covered Area Manager. The main job of the *Covered Area Manager* VAF is the implementation of the area coverage algorithms. Also in this case (since the solution is distributed), each Virtual Drone has its own Covered Area Manager, and its objective depends on the UAV role, i.e. whether it is the leader or a non-leader. In the latter (non-leader) case, this VAF takes only into account the area parts which are gradually covered by the UAV, storing the relevant information in a local *Area Part Database (APD)*. In the former (leader) case, the functionality is a little bit more complex and includes the following steps:

1) the leader polls the APD of all the other UAVs and obtains all the parts already covered by the flock;
2) the obtained data is merged in order to have a global and unique view of the area parts to be still covered;
3) on this basis, the leader plans some possible paths that allow the needed covering and selects the best one (shortest);
4) the leader starts to fly on the chosen path and all the other UAVs, thanks to the flocking algorithm, follow the leader and perform the monitoring functions.

This algorithm is executed periodically, with a period of about $\frac{1}{8}s$; this is required since, if one or more UAVs fail, the area parts covered by them (but not yet transmitted and stored) must be rescanned: thanks to the said algorithm, the path is continuously updated and adapted to changing conditions.

Local Video Processor. We suppose that each twin physical UAV is equipped with a camera sensor that acquires images or video of the monitored area. Such a video is streamed to the *Local Video Processor* that has the task of performing a pre-processing according to the functions specified by the high-level user like, for example, compression, feature extraction, encryption, etc.

Global Video Processor. The *Global Video Processor* receives all the video flows pre-processed by each Local Video Processor and containing the images relating to each monitored sub-area, and compose them to create a global video flow of the monitored area as a whole. The resulting video flow is locally stored by the Global Video Processor for a short time, and then is transferred to the *Historical Video DB* to be stored definitively.

Historical Video DB. The *Historical Video DB* VAF is a simple storage element; it receives from the Global Video Processor, data that not only contains the images themselves but also additional information including metadata, tags and information extracted from the Local Video Processor. All of these data are thus indexed and properly stored in order to be available for future search and analysis.

A further VAF that plays a key role in the whole monitoring system is the ***Mission Manager***, whose job is to control and manage the flight of the UAV flock. For example, it imposes the shape and the speed of the flock, according to the instructions received by the human service operator that is referred to as *Mission Initiator* in Fig. 2.

As specified in [1], [19], the UAV Flight Controllers of all the UAVs need to exchange information. Thanks to the fact that they run inside the same slice, we can create a backend virtual network connecting all the Flock Controllers that, therefore, can communicate with each other by broadcasting their information to be shared with the other UAVs. The same is achieved for information exchange among the Covered Area Managers of all the virtual UAVs.

Now, considering the flocking formation and maintenance algorithm and the data to be exchanged through each interface shown in Fig. 2, we have derived the delay requirements for each interface shown in Table I, together with the kind and the size of exchanged data.

B. Interface Definition

The interfaces defined between VAFs are listed in Table I together with communications requirements that have to be negotiated with the *Network Slice Orchestrator*.

$I_{PD,VD}$ is a bi-directional interface between the Physical Drone and the Virtual Drone. In the uplink direction, the Physical Drone sends all the data related to inertial and position sensors, i.e. gyroscope, accelerometer, magnetometer, barometer and GPS (for the first tree sensors, a triple of values is acquired, one for each geometric axis X, Y and Z). In the downlink direction, the Virtual Drone sends the power of the motors computed by the stabilization and navigation algorithm.

The $I_{Fl,Fk}$ interface connects the UAV Flight Controller and the Flock Controller. Here exchanged data are the ones that are relevant to the flocking algorithm: the uplink conveys position data while the downlink carries the set points for translational and angular speeds.

The $I_{Fl,A}$ interface connects the Flock Controller to the Covered Area Manager; here the uplink is used to carry the information about the area portions which are gradually covered, and the downlink, which is used only by the leader, transmits the path planned to perform area coverage.

Interfaces $I_{Fk,Bc}$ and $I_{A,Bc}$ are two broadcast communication interfaces used to share data among respectively Flock Controllers and Covered Area Managers. They must support the same bandwidth and latency of $I_{Fk,Fl}$ and $I_{Fk,A}$.

The Interface $I_{PD,L}$ is used by the Physical Drone to transmit the captured images to the Local Video Processor, the requirements of such an interface are therefore similar to (but not the same of) those of traditional video channels, even if real-time capabilities are not requested since the video must not be displayed. The same characteristics are required by the $I_{L,G}$ interface, which connects each Local Video Processor to the relevant Global Video Processor.

The $I_{M,Flock}$ interface is used by the Mission Manager to control the flock, by sending to the Virtual Drones mission-specific parameters, such as area bounds, flight altitude, max flight speeds, etc.

The $I_{O,M}$ interface allows the human operator to configure the mission by sending the configuration parameters to the Mission Manager. On the other side, the Mission Manager periodically provides the Human operator with the updates about the mission progress.

Finally, the $I_{M,O}$ interface allows the Mission Manager to negotiate the network slice parameters, in terms of computation, storage and latencies, with the Network Slice Orchestrator, not only at the slice setup, but also at run time if some modification is required during the service lifetime.

Information contained in Table I will be used by the Network Slice Orchestrator to decide the placement of the VAFs inside the clouds available in the network, and the links to realize the virtual graph of the overall virtual application service. For example, each UAV Flight Controller must be deployed in the Cloudlet closest to the current access point of the corresponding Physical Drone, while the Global Video Processor and the Historical Video DB must be placed in clouds where high computation resources and high storage resources, respectively. are available. Given that no strict latency requirements are specified for these last elements, the most suitable placement candidates for them are the core clouds.

IV. Conclusions

This paper proposes a distributed platform for an aerial video-monitoring service realized by leveraging on the Tactile Internet slice of a 5G communication system. The UAVs providing this service are organized in flock, and their control is performed by a chain of virtual application functions (VAF) running on the ground, in clouds at the edge of the network. Very hard control loops, as the one needed to control the engines of each UAV, which are usually realized by control functions installed on board of the UAVs, can now be realized outside the UAVs, i.e. on the ground, only thanks to the ultra-low latency and high reliability peculiarities guaranteed by the Tactile Internet. The resulting deployment of computations in the edge provides many advantages, in terms of scalability and fault-tolerance, and avoids processing latencies, due to communication links, that represent an important drawback in traditional solutions. Dimensioning the maximum number of UAVs that can belong to the same flock, their minimum distance and the maximum speed for a given UAV are considered as a future work. Another future work will consist in the definition of autonomous placement, resource allocation and orchestration policies that will be able to decide where placing VAFs, and the amount of networking, storage and computing resources providing to each of them, even in presence of mobility, that is, immediately after handover events, which can be very frequent during a flock flight.

References

[1] M. De Benedetti, F. D'Urso, G. Fortino, F. Messina, G. Pappalardo, and C. Santoro, "A fault-tolerant self-organizing flocking approach for uav aerial survey," *J. Netw. Comput. Appl.*, vol. 96, no. C, pp. 14–30, Oct. 2017.

Interface	Packet	Size	Latency
$I_{PD \leftarrow VD}$	sensor data (accel, gyro, magnetom.), lat,lng, altitude, status	60 byte	1 ms
$I_{PD \rightarrow VD}$	motor power	32 byte	1 ms
$I_{Fl \rightarrow Fk}$	lat,lng,roll,pitch,yaw,status	29 byte	$\frac{1}{50}ms$
$I_{Fk \rightarrow Fl}$	V_x,V_y, V_z, ω_z	16 bytes	$\frac{1}{50}ms$
$I_{Fk \rightarrow A}$	Image bounds	16 bytes	$\frac{1}{8}ms$
$I_{A \rightarrow Fk}$	Path bounds	16 bytes	$\frac{1}{8}ms$
$I_{Fk,Bc}$	lat,lng,hgt,yaw,status	17 byte	1/8 s
$I_{Fk,L}$	compressed image from camera	1 MB - 10 MB	best effort
$I_{L,G}$	compressed image from camera	1 MB - 10 MB	best effort

TABLE I
INTERFACE LIST AND DESCRIPTION

[2] Z. Zheng, A. K. Sangaiah, and T. Wang, "Adaptive communication protocols in flying ad hoc network," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 136–142, Jan 2018.

[3] G. Schembra C. Rametta, "designing a softwarized network deployed on a fleet of drones for rural zone monitoring," *Future Internet*, vol. 9, no. 1, March 2017.

[4] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van, "The tactile internet: vision, recent progress, and open challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 138–145, May 2016.

[5] A. A. Ateya, A. Muthanna, and A. Koucheryavy, "5g framework based on multi-level edge computing with d2d enabled communication," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, Feb 2018, pp. 507–512.

[6] S. Iellamo, J. J. Lehtomaki, and Z. Khan, "Placement of 5g drone base stations by data field clustering," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, June 2017, pp. 1–5.

[7] L. Chiaraviglio, W. Liu, J. A. Gutierrez, and N. Blefari-Melazzi, "Optimal pricing strategy for 5g in rural areas with unmanned aerial vehicles and large cells," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Nov 2017, pp. 1–7.

[8] M. Dohler, T. Mahmoodi, M. A. Lema, M. Condoluci, F. Sardis, K. Antonakoglou, and H. Aghvami, "Internet of skills, where robotics meets ai, 5g and the tactile internet," in *2017 European Conference on Networks and Communications (EuCNC)*, June 2017, pp. 1–5.

[9] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, "Realizing the tactile internet: Haptic communications over next generation 5g cellular networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 82–89, April 2017.

[10] France:ETSI Sophia Antipolis, "Mobile edge computing: A key technology towards 5g," Sep 2015.

[11] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1657–1681, thirdquarter 2017.

[12] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[13] J. d. J. Gil Herrera and J. F. Botero Vega, "Network functions virtualization: A survey," *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 983–997, Feb 2016.

[14] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5g: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, May 2017.

[15] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, May 2017.

[16] Jean-Claude Latombe, "Exact cell decomposition," in *Robot Motion Planning*, pp. 200–247. Springer, 1991.

[17] Jean-Claude Latombe, "Approximate cell decomposition," in *Robot Motion Planning*, pp. 248–294. Springer, 1991.

[18] Ioannis Rekleitis, Ai Peng New, Edward Samuel Rankin, and Howie Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 109–142, 2008.

[19] Massimiliano De Benedetti, Fabio DUrso, Fabrizio Messina, Giuseppe Pappalardo, and Corrado Santoro, "3d simulation of unmanned aerial vehicles," in *XVIII Workshop "Dagli Oggetti agli Agenti"*. CEUR-WS, 2017.

[20] Pasquale Pace, Gianluca Aloi, Giuseppe Caliciuri, and Giancarlo Fortino, "A mission-oriented coordination framework for teams of mobile aerial and terrestrial smart objects," *Mobile Networks and Applications*, vol. 21, no. 4, pp. 708–725, 2016.

[21] M. D. Benedetti, F. D'Urso, F. Messina, G. Pappalardo, and C. Santoro, "Uav-based aerial monitoring: A performance evaluation of a self-organising flocking algorithm," in *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, Nov 2015, pp. 248–255.