Ad Hoc Networks 83 (2019) 68–77

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Software-defined unmanned aerial vehicles networking for video dissemination services

Zhongliang Zhao[a,*], Pedro Cumino[b], Arnaldo Souza[b], Denis Rosário[b], Torsten Braun[a], Eduardo Cerqueira[b], Mario Gerla[c]

[a] University of Bern, Switzerland
[b] Federal University of Pará, Brazil
[c] University of California, Los Angeles, United States

ARTICLE INFO

Article history:
Received 28 February 2018
Revised 29 August 2018
Accepted 30 August 2018
Available online 31 August 2018

Keywords:
SDN
UAVNet
Video surveillance
Relay placement
Quality of experience

ABSTRACT

Unmanned Aerial Vehicles (UAVs) empower people to reach endangered areas under emergency situations. By collaborating with each other, multiple UAVs forming a UAV network (UAVNet) could work together to perform specific tasks in a more efficient and intelligent way than having a single UAV. UAVNets pose special characteristics of high dynamics, unstable aerial wireless links, and UAV collision probabilities. To address these challenges, we propose a Software-Defined UAV Networking (SD-UAVNet) architecture, which facilitates the management of UAV networks through a centralized SDN UAV controller. In addition, we introduce a use case scenario to evaluate the optimal UAV relay node placement for life video surveillance services with the proposed architecture. In the SD-UAVNet architecture, the controller considers the global UAV relevant context information to optimize the UAVs' movements, selects proper routing paths, and prevents UAVs from collisions to determine the relay nodes deployment and guarantee satisfactory video quality. The experimental results show that the proposed SD-UAVNet architecture can effectively mitigate the challenges of UAVNet and it provides suitable Quality of Experience (QoE) to end-users.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are autonomous aircrafts without the need of a pilot to be on board during the flight. UAV usage started with military applications, but now UAVs are getting more and more popular for public, civil, and even personal applications [1]. In military use cases, UAVs are deployed to execute military operations, such as battlefield inspection, geographic mapping of inaccessible terrain, border control surveillance, etc. On the other hand, in public and civil use cases, UAVs can be used by individuals or companies for aerial photography, video surveillance, etc. Multiple UAVs might collaborate in order to form a UAV network (UAVNet) [2,3] to perform tasks in a more efficient and economic way than having only a single UAV [4]. Compared to a single UAV system, a UAVNet is more robust and can cover a larger area of interest. In this context, video dissemination over an UAVNet enables a large class of multimedia applications such as disaster recovery, environmental monitoring, safety & security, and others [1]. Hence, multimedia data plays an important role to provide rich visual information to help the ground rescue teams to take appropriate decisions [5]. However, video dissemination over UAVNets with Quality of Experience (QoE) support is a hard task due to frequent topology changes caused by UAV mobility [6,7]. For instance, UAVs movements break plenty of communication links, which increase the packet loss during video transmissions.

UAVs can be used for autonomous flights, either by following preprogrammed flight plans or through the help of more complex dynamic automation systems. Therefore, an intelligent inter-UAV coordination protocol must be defined to control the behaviors of multiple UAVs, maximizing the benefits of a UAVNet. This UAV controlling protocol manages both the inter-UAV communication (*i.e.*, routing protocols), as well as the UAV movement trajectory. In this context, Software-Defined Networking (SDN) [8–10] is a promising approach for this multi-UAV controlling task, since it introduces complete network programmability by separating the control plane and data plane. SDN also considers a centralized SDN controller responsible for all control functions such as UAVs placement, collision avoidance, and other services.

SDN has been successfully applied in wired networks, such as data center networks or backbone networks. Specifically, these networks have a centralized architecture, where a control center is

* Corresponding author.
E-mail address: zhao@inf.unibe.ch (Z. Zhao).

https://doi.org/10.1016/j.adhoc.2018.08.023

responsible for network management [11]. The SDN controller is deployed in the control center, and it is aware of the global network conditions to optimize the network operations, by dynamically configuring the data plane routes. For SDN deployments in wired networks with fixed infrastructure topology, programmability means that the control plane adaptively adjusts the data path, and the data plane follows the instructions to forward the packet through different interfaces. On the other hand, SDN applied to UAVNets, the UAV programmability means to control the mobility trajectory of the UAVs in order to avoid UAV collisions or to improve the application performance, to determine the data routing paths, to change the packet transmission parameters (data rate or transmission power) due to performance or energy reasons, and others. UAVNets are mostly deployed to perform specific tasks, such as assisting disaster recovery communications, surveillance monitoring, remote sensing/searching, etc. In these tasks, UAVNet nodes must collaborate with each other and their behaviors (both the data transmission and the UAV movement) should be controlled by a centralized entity, *i.e.*, a controller.

In this article, we propose a Software-Defined UAVNet architecture (SD-UAVNet), which implements the concept of SDN into UAVNets to separate the control and data plane, as well as to provide network programmability by controlling UAVs' operation parameters. With the proposed SD-UAVNet architecture, we focus on the problem of optimal UAV relay node placement for real-time video services, which reduces the impact of node mobility on the QoE of delivered videos. The SD-UAVNet controller considers the global UAV context information to prevent UAV collisions, optimize the UAVs' movements, and establish a routing path that determines the relay node deployment to provide video transmission with QoE support. Specifically, the controller considers multiple UAV context information for routing and UAV placement decisions, namely, UAV positions, UAV movement trajectories, and residual energy.

This article can be seen as a starting effort to explore the benefits of using SDN to control a UAVNet system to perform specific tasks in smart environments. As a first step, we focus on applying the proposed SD-UAVNet architecture to assist the video surveillance services in disaster recovery scenarios. The contribution of the article can be summarized as follows:

- We propose the SD-UAVNet architecture, which includes a SD-UAVNet controller that collects the global context information of UAV nodes, e.g., remaining energy, location information, mobility trajectory, speed, and others. It makes the routing decisions and decides how the UAV nodes should move.
- We focus on a video surveillance application, where the SD-UAVNet controller determines the locations of the UAV relay nodes by considering their current locations and residual energy for collision avoidance during their movements.
- The SD-UAVNet architecture contains a detailed energy model of UAV nodes, which includes the energy consumption for both the data transmission and UAV movements.
- An extensive simulation study validates our system performance, which shows that the SD-UAVNet architecture can effectively mitigate the challenges related to topology changes caused by UAV mobility in order to provide video dissemination with satisfactory QoE.

The article is organized as follows. Section 2 discusses existing work on applying SDN in MANETs or VANETs, and UAVNets for video dissemination services. Section 3 presents the SD-UAVNet architecture and its components. Section 4 describes the simulation settings for the performance evaluation in a video dissemination scenario and also discusses the results. Section 5 concludes the article and foresees future work.

2. Related work

Using small size UAVs in public and civilian applications are getting more popular. UAVs can be deployed in areas of disaster management, network capacity enhancement, etc. Merwaday and Guvenc [12] proposed to use UAVs as aerial base stations (UABSs) to assist public safety communications during natural disasters, as soon as parts of the communication infrastructure become damaged and dysfunctional. They showed that the deployment of UABSs at optimized locations can improve the throughput gains under disaster scenarios. Sharma et al. [13] introduced a neural network-based cost function to assign UAVs to a particular geographical area subject to high traffic demands. Their results showed that leveraging multiple UAVs not only provides long-range connectivity but also better load balancing and traffic offloading. Mozaffari et al. [14] investigated the optimal trajectory and deployment of multiple UAVs used as aerial base stations to collect data from ground Internet of Things (IoT) devices. Even though these works could handle most of the UAVs' deployment problems in some use cases, none of them include a centralized entity to control UAVs as expected in SDN systems. The benefit of using a centralized UAV controlling system is that the controller can make decisions based on the global UAV context information to make sure that all the UAV routing or movement operations are optimized. In military applications, the common architecture is to use one manned ground control station (GCS) to control all UAV devices. The GCS-UAVs communications are 1-hop direct communication, which significantly limits the coverage area of UAV deployments [15].

Recently, there were some attempts to borrow the idea of splitting control and data planes into MANETs or VANETs. Yu et al. [16] implemented a practical SDN-MANET system to support management flexibility in a wireless ad-hoc network. Similarly, Ku et al. [17] proposed a SDN-based VANET architecture and showcased the benefits of SDN-VANET services by performing some preliminary simulation studies. Yang et al. [18] claimed that they exploited the SDN-based scheme to integrate drone-cells to 5G networks to address the high traffic demands. However, they do not give any details about how the controlling mechanism works, where is the controller, how the controller is selected, etc. Compared to their work, we focus on design details of a SD-UAVNet architecture with solid considerations of UAV specifics, such as energy constraints, collision free requirements, dynamic mobility, etc.

Energy conservation is extremely important for UAVNets, since UAVs are all battery-powered and their aerial movements are very energy-consuming. There are mainly two sources for energy consumptions at UAVs: i) communication equipment consumes energy for data transmission among UAVs and ii) energy consumption of power engines for UAV movements. Di Franco and Buttazzo [19] developed a UAV energy consumption model derived from real measurements, which considers different mission-relevant parameters, such as UAV movement speeds. Tseng et al. [20] conducted empirical studies to measure the energy consumption of UAV hovering, vertical climbing, horizontal movements, and proposed energy models under various scenarios.

Unlike MANETs or VANETs, UAVNets consist of nodes that are unmanned-piloted, which means safety is of high priority in a mission. Therefore, UAV movements should be controlled by a centralized controller to avoid collisions and optimize UAV node operations. However, none of the existing works tackle this problem. Mozaffari et al. [21] proposed an efficient deployment method to determine the optimal 3D UAV locations by considering the target area, coverage requirements, and the number of UAVs. However, the work focused on coverage using only static UAVs. Existing UAV node placement solutions are all distributed, and do not have collision avoidance guarantees. Rosário et al. [22]

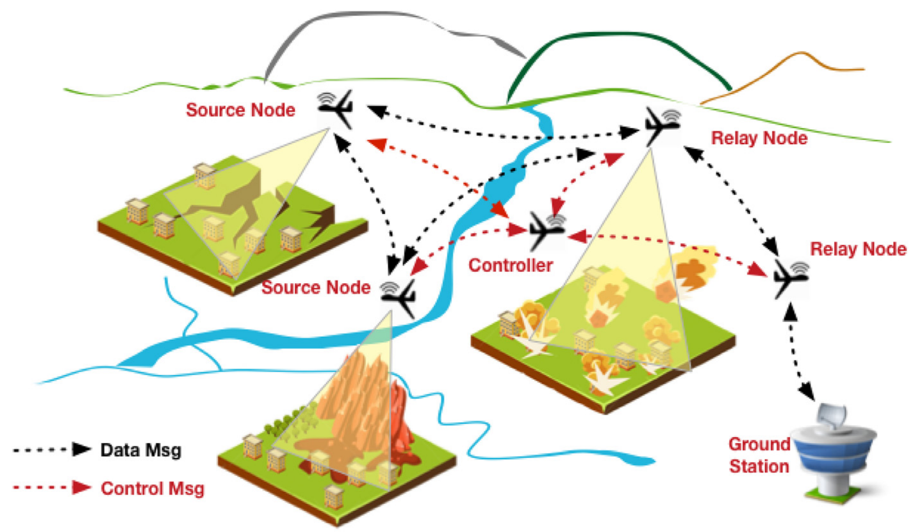


Fig. 1. SD-UAVNet deployed for video surveillance in a disaster recovery scenario.

proposed a distributed relay placement mechanism to support live video transmissions with satisfactory QoE to the users. However, the relay placement mechanisms could be enhanced with a centralized controller. In our design, a centralized controller manages not only the mission-specific decisions such as relay node selection, but also defines UAVs' movement parameters, such as mobility models, speeds, etc.

Routing is another challenging issue in UAVNets, where it is essential to consider an efficient relay placement service to find the ideal relay location by considering the global UAV geographical location to provide video dissemination with QoE support. Although there are many protocols to deliver videos with QoE requirements, they are not effective due to both communication flaws and void areas caused by UAV mobility. Magán-Carrión et al. [23] introduced a relay placement mechanism, where the overall problem is divided into steps. In the first step, a set of relay nodes are homogeneously distributed along the edges to connect partitions in the network, according to a distance-based spanning tree. Afterwards, the mechanism estimates the optimal locations for relay nodes by taking into account Leave-one-out (LOO) and particle swarm optimization (PSO) algorithms. Given the above limitations of the existing work, we propose SD-UAVNet, a SDN-enabled UAVNet simulation framework that fully addresses the unique characteristics of UAV networks.

3. SD-UAVNet system for video dissemination services

This section describes the SD-UAVNet architecture, including the SD-UAVNet controller function, the communication protocol between controller and UAV relay nodes, the routing and relay placement algorithms. With the proposed SD-UAVNet architecture, we focus on a disaster recovery scenario using a UAVNet to disseminate real-time videos. The SD-UAVNet controller is responsible for managing the topology of UAV networking and controls the locations and mobility models of all UAV nodes for ensuring a QoE-satisfied video surveillance mission.

3.1. System model

Fig. 1 depicts a SD-UAVNet deployed to endangered areas under the disaster recovery scenario in order to provide real-time video surveillance, as soon as the normal communication infrastructures are temporarily unavailable. In this scenario, live video content retrieved from the event area enable humans in the ground control center to take action in order to explore a hazardous area, where rescuers are unable to reach the disaster area easily and quickly.

We consider n UAVs (nodes) deployed in the monitored area, and each UAV has an individual identity ($i \in [1, n]$). Those UAVs are represented in a dynamic graph $G(V, E)$, where the vertices $V = \{v_1, \cdots, v_n\}$ mean a finite set of UAVs, and edges $E = \{e_1, \cdots, e_n\}$ build a finite set of asymmetric wireless links between 1-hop UAV (v_i) neighbors. We denote $N(v_i) \subset V$ as a subset of all 1-hop neighbors within the radio range ($R_{\max}$) of a given UAV v_i. Each UAV v_i has a queue (Q) with a maximum queue capacity ($Q_{\max}$). The queue policy schedules the packet transmission by using the First In First Out (FIFO) algorithm and drops packets using the Drop Tail algorithm. In such scenario, each UAV v_i is equipped with a camera, an image encoder, a radio transceiver, and limited energy supply. For convenience of notation, we denote $SN \subset V$ (Source Node) as the UAV v_i responsible for capturing video flows and transmitting them to the Destination Node (DN) $\subset V$, *i.e.*, ground station, in a multi-hop fashion. We assume a UAVNet scenario composed of one static DN equipped with a radio transceiver, an image decoder, and unlimited energy, which is responsible for receiving the video for further processing and dissemination. The location of the DN is assumed to be known a priori by each node v_i.

The Controller Node (CN) $\subset V$ is responsible for managing not only the mission-specific decisions such as relay node selection. It also defines all the UAV movement parameters such as mobility models, speeds, and others. The controller CN could be a ground station or a regular UAV node. For instance, a fixed wing UAV has more energy and larger radio range, which can periodically collect UAV information and send configuration commands. In addition, for large scale environments, it might be necessary to deploy multiple local Controller Nodes that are responsible only to forward the control message to/from the central Controller Node. This enables to transmit control messages with lower transmission power, which leads to reduced interference and energy consumption, as well as better control message reliability.

Each UAV v_i is able to estimate its remaining energy (RE_i), where it spends energy to transmit a packet (P_{tx}), to receive a packet (P_{rx}), and to move with a given speed v (P_v). We consider the energy consumption model for UAV movements provided by Tseng et al. [20]. It should be highlighted that UAV movements require much more energy than for packet transmissions. Each UAV

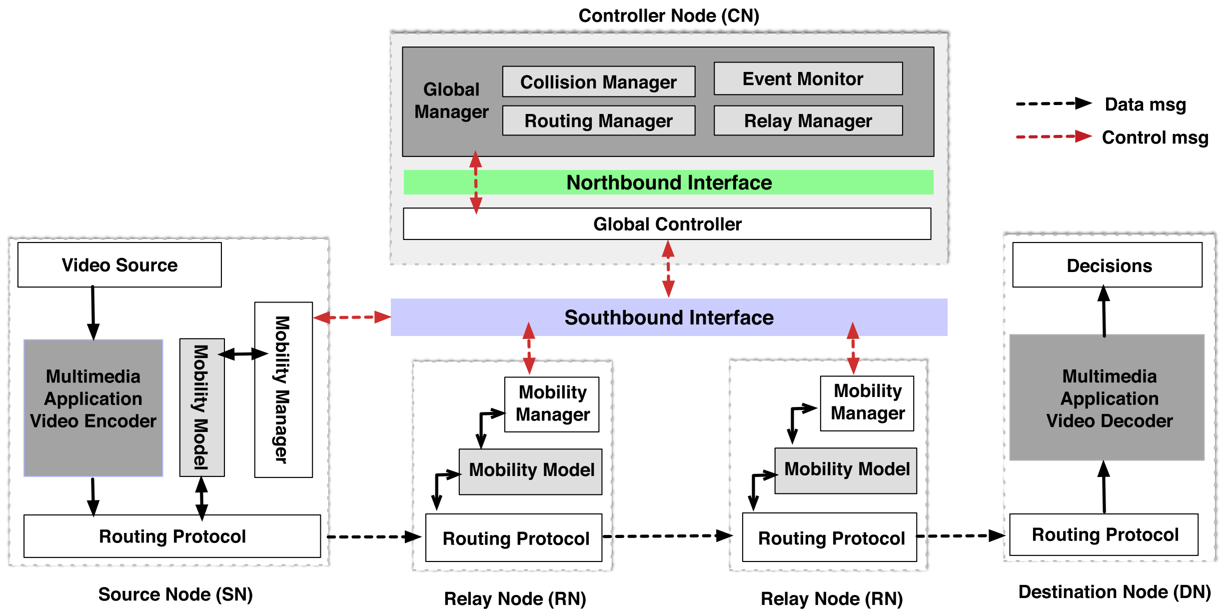


Fig. 2. SD-UAVNet node component architecture.

v_i is aware of its own location $L(X_i, Y_i)$ by means of a positioning system, *e.g.*, GPS or Galileo. Each UAV (v_i) flies with a given speed s_i ranging between a minimum (*e.g.*, $s_{\min}$) and a maximum (*e.g.*, $s_{\max}$) speed limit towards a given trajectory ($traj_i$). We consider that UAVs fly following the Paparazzi mobility model [24], because it enables UAVs to adapt to any type of mission, and also groups most possible UAV movements by changing the probability of each movement type as needed [25]. Particularly, this mobility model considers five possible trajectories ($traj_i$): Stay-At (*i.e.*, UAV flies in a circle), Way-point (*i.e.*, UAV flies following a straight line to a destination position), Eight (*i.e.*, UAV trajectory has the 8 form around two fixed positions), Scan (*i.e.*, UAV performs a scan in an area defined by two points along the round trip trajectories); and Oval (UAV trajectory has an oval form).

3.2. SD-UAVNet architecture

Fig. 2 introduces the proposed SD-UAVNet architecture, which implements the concepts of SDN in a UAVNet. The proposed architecture is composed of four types of nodes, namely, multiple *Source Nodes*, one *Destination Node*, one *Controller Node*, and multiple *Relay Nodes*. Specifically, the *Controller Node* performs all control functions, while *Source Nodes* transmit video flows to the *Destination Node* via multiple *Relay Nodes* in a multi-hop fashion. In the following, we give detailed explanations of each component inside of every node, with more details on the design of the *Controller Node* design.

- The *Controller Node* is the key component of the SD-UAVNet architecture. It is responsible for maintaining the network topology updated to make sure that UAVs are connected, to establish the routing path, and also to place UAVs at the optimal locations to forward the packets.
- A *Source Node* captures live video stream of the interested area, encodes it, and transmits the packets towards the destination *DN*.
- A *Relay Node* is responsible for forwarding the packets to reach the destination *DN*. Their locations and movement patterns are decided by the *SD-UAVNet Controller Node*.
- A *Destination Node* is the ground station, which receives the video streams, supervises the target areas, and makes corresponding decisions.

The *Controller Node (CN)* is responsible for UAVNet management, via the *Global Manager* and *Global Controller*. The *CN* must be aware of each UAV v_i contextual information, such as remaining energy RE_i, location information $L(X_i, Y_i)$, mobility trajectory $traj_i$, speed s_i, etc. These data are periodically collected by the *Global Controller* from each UAV via the southbound SDN interface. Specifically, the *Global Controller* plays the role of an interface between the *CN* and the UAVNet, synchronizing data exchange in both directions. It is in charge of requesting the UAV contextual information, and also sending instructions to UAVs. It also decides and imposes the frequency of this process, *i.e.*, it controls how often contextual information should be collected from UAVs, and thus how often decision making will be made.

On the other hand, the *Global Manager* module includes 4 components: *Collision Manager, Event Monitor, Routing Manager*, and *Relay Manager*. It performs all the decision making based on the inputs provided by the *Global Controller* module, such as controlling the UAV movements, deciding the optimal locations of relay nodes, configuring the transmission power of relay nodes, establishing the routing data path, and others. For instance, based on specific algorithms and monitored node movement patterns, it can estimate collision possibility and, thus, decides which UAV should be the relay node, and how each relay UAV v_i should move by considering the global topology information. In addition, based on UAV contextual information and the event location, the *Global Manager* module is able to establish the routing path, and also to place UAVs at the optimal locations to forward the packets, reducing the impact of node mobility on the QoE of delivered video. In the following, we describe the global context information acquisition, the relay optimal placement mechanism, and the routing decision procedures that are performed by the *Global Controller*, which are depicted in Fig. 3. The key functions are described below.

3.2.1. Global context acquisition

The *Controller Node* has to learn the status of each UAV v_i through the *Global Controller* module and each UAV v_i periodically sends its contextual information to the controller node,

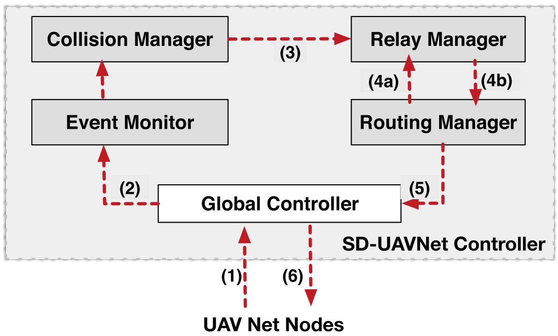


Fig. 3. Message flow between SD-UAVNet controller and UAVNet.

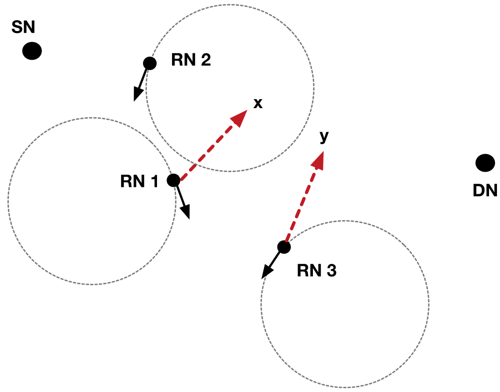


Fig. 4. Controller-assisted UAVs collision avoidance.

including remaining energy RE_i, location information $L(X_i, Y_i)$, mobility trajectory $traj_i$, speed s_i, and others. In this way, the *Controller Node* has full knowledge of UAVs contextual information. Based on specific mobility prediction algorithms, the *Global Controller* can change the frequency of requesting UAV contextual information, reducing bandwidth consumption by signaling overhead.

3.2.2. Collision avoidance

The *Controller Node* needs to consider all the available UAVs' contextual information to make sure that the relay node movement will not introduce any possible collisions. For instance, suppose that three relay nodes, *i.e.*, RN_1, RN_2, and RN_3, are following their movement trajectory of circular mobility, as shown in Fig. 4. Let us assume that both RN_1 and RN_3 want to move to their ideal locations (*location x* and *location y*) to forward the packet for *SN*. In this case, there is high probability of having collisions between RN_1 and RN_2 when the RN_1 flies from its current location to *location x*. However, with the global topology, the *Collision Manager* could avoid UAV collisions by choosing RN_3 as the relay node.

3.2.3. Routing data path with relay location optimization

The *Routing Manager* computes the routing data path $P_{SN,\,DN}$ between source and destination nodes via multiple relay nodes. It takes into account UAV contextual information and the event location. As soon as the *Event Monitor* detects an event at a given location, the *Routing Manager* must find the UAV closest to the event area to capture and transmit videos from the event area. This UAV must have enough energy to capture and transmit the video, as well as to fly. The *Routing Manager* must compute a function $F_i \in [0,1]$ to all UAVs v_i based on Eq. (1). It must select the UAV v_i with the lowest value to become the *SN*, since this is the node closest to the event with enough energy to transmit the video and to fly. Arriving at the event location, the *SN* must fly over this point following the Stay-At, Eight, or Oval movement trajectories with the minimum speed $S_{\min}$, which enable the *SN* to fly around a fixed position to capture highly relevant videos from the detected event.

$$SN/RN = \min(v_i) = v_i = \begin{cases} 1 & \text{if } t_{\text{move}} > t_{\max} \\ 1 & \text{if } E_{\text{estimated}} > RE_t \\ w_1 \times t_{move} + w_2 \times RE_i & \text{otherwise} \end{cases} \quad (1)$$

Time to move t_{move} computed based on Eq. (2) represents the time needed to fly from the current location $L(X_i, Y_i)$ to the event location $L(X_e, Y_e)$ at the maximum speed $S_{\max}$. Even spending more energy, SN flies with maximum speed $S_{\max}$ to quickly reach the event location, since the main goal is to disseminate video in cases of an event and not save energy. The *Routing Manager* must select UAVs with t_{move} lower than a defined threshold $t_{\max}$.

$$t_{move} = \frac{dist(L(X_i, Y_i), L(X_e, Y_e))}{S_{max}} \quad (2)$$

Estimated energy $E_{estimated}$ computed based on Eq. (3) denotes the energy required to capture and transmit the video, as well as to fly. $E_{estimated}$ must be higher than the RE_i to ensure that the *SN* will have enough energy to transmit the video, while flying over the event area.

$$E_{estimated} = E_{video} + E_{fly} \quad (3)$$

Afterwards, the *Routing Manager* must compute again the function F_i to all UAVs v_i based on Eq. (1) to select the relay nodes. It must select the UAV v_i with the lowest value to become the RN, since this UAV v_i is the closest to the ideal relay node location, represented as $L_{ideal}(X_i, Y_i)$.

Placing relay nodes at the ideal location reduces the impact of node mobility on the QoE of the delivered video. In this sense, the *Relay manager* computes the optimal locations of relay nodes $L_{ideal}(X_i', Y_i')$ based on Eq. (4) [22]. We consider the following metrics: *SN* location $L(X_{SN}, Y_{SN})$, *DN* location $L(X_{DN}, Y_{DN})$, radio range ($R_{\max}$), Stay-At movement range (*MR*), and the Euclidean distance ($dist_{SN,\,DN}$) between nodes *SN* and *DN*. The ideal location must seek to reduce the number of hops, while mitigating route failures and void areas caused by UAV movements.

$$L_{ideal}(X_i', Y_i') = \begin{cases} X_i' = X_{SN} - \frac{(R_{max} - 2MR) \times (X_{SN} - X_{DN})}{Dist_{SN,DN}} \\ Y_i' = Y_{SN} - \frac{(R_{max} - 2MR) \times (Y_{SN} - Y_{DN})}{Dist_{SN,DN}} \end{cases} \quad (4)$$

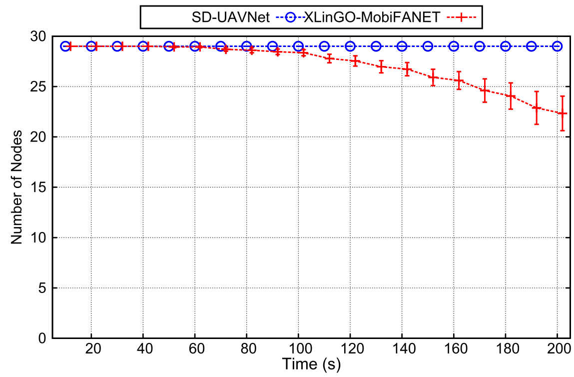
This might keep the connectivity between the nodes of a given path $P(SN, DN)$. The selected relays *RN* must fly to $L_{ideal}(X_i, Y_i)$ with the maximum speed $S_{\max}$ to quickly reach such location, since the main goal is to place each *RN* at $L_{ideal}(X_i, Y_i)$ to forward the video packets with QoE support. Arriving at $L_{ideal}(X_i, Y_i)$, the *RN* must fly over this point following the Stay-At movement at a given speed $S_{\min}$ to keep the connectivity among UAVs that belong to $P_{SN,\,DN}$. The *Global Controller* sends these information to the UAVs that belong to the path $P(SN, DN)$. Finally, the *Event Monitor* analyzes the remaining energy of all UAVs that belong to a given path $P(SN, DN)$. In this sense, it might ask the *Routing Manager* to replace a given relay node that will run out of energy during the video transmission, avoiding route failures caused by the energy depletion of a given relay node.

4. Evaluation

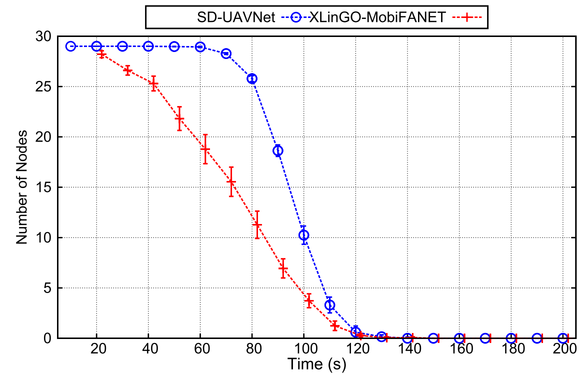
This section presents the methodology and metrics applied to evaluate the SD-UAVNet architecture for video surveillance applications with a relay placement mechanism in a disaster recovery scenario. We evaluated the impact of different UAV speeds on the maintenance of the number of route failures, as well as QoE.

4.1. Simulation description and metrics

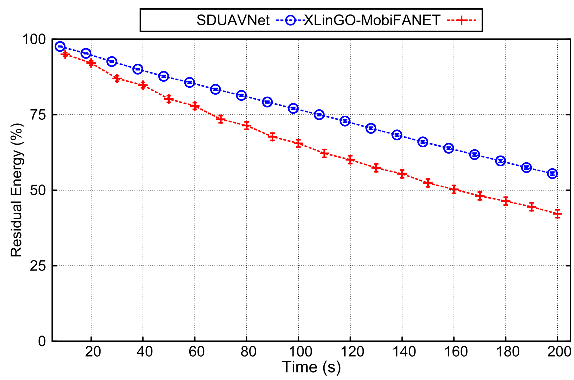
We implemented the SD-UAVNet on the Mobile Multi-Media Wireless Sensor Network (M3WSN) OMNeT++ framework [26]. We



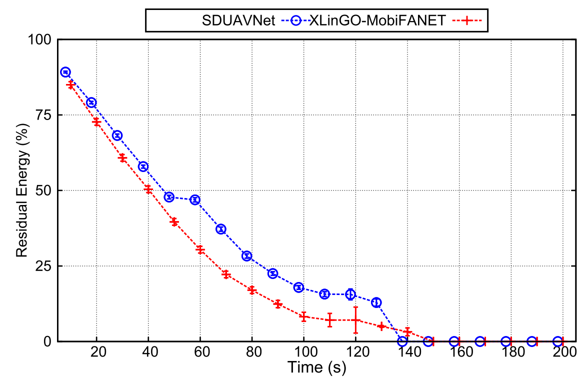
(a) Number of UAVs alive for UAV flying with speeding raging from 10 to 15 m/s



(b) Number of UAVs alive for UAV flying with speeding raging from 15 to 20 m/s



(c) Remaining Energy for UAV flying with speeding raging from 10 to 15 m/s



(d) Remaining Energy for UAV flying with speeding raging from 15 to 20 m/s

Fig. 5. Remaining energy and number of UAVs alive.

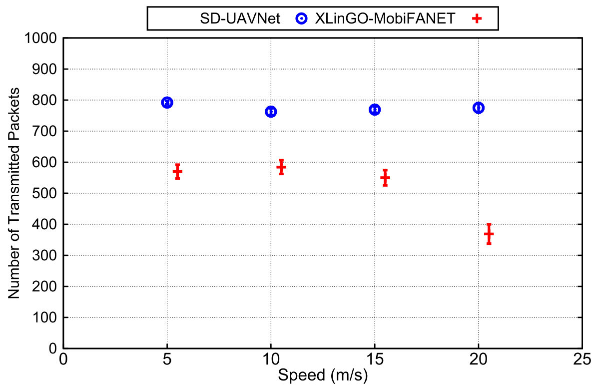
conducted 33 simulation runs with different randomly generated seeds, and the results show the values with a confidence interval of 95%. The simulations last for 200 seconds (s) and run with the lognormal shadowing path loss model. We set the simulation parameters to allow wireless channel temporal variations, link asymmetry, and irregular radio ranges, as expected in a real UAVNet scenario.

We consider a UAVNet scenario composed of 35 UAVs, where 29 UAVs are flying following the PPRZM [24] over the entire flat terrain of 200 x 200m [5] to explore, and disseminate live video streaming from the environment. Such UAVs are flying with different speed limit intervals: i) 1 to 5 m/s; ii) 5 to 10 m/s; iii) 10 to 15 m/s; iv) 15 to 20 m/s. As expected in UAVNet multimedia applications such as safety & security, environmental monitoring, and natural disaster recovery, we have one fixed Destination Node (*DN*) located at (0, 100) [22]. We consider one UAV located at (100, 100), working as controller node *CN* to periodically collect UAV information and send configuration commands. In addition, we have 4 UAVs responsible only to forward control message for the central controller (*CN*). Further, all UAV nodes are equipped with IEEE 802.11 radio and the transmission power is set to 12dBm, resulting in a nominal $R_{\max}$ of 55 m. Based on the simulation area and $R_{\max}$, videos are received at the *DN* via 1 to 4 relay nodes RN_i depending on the routing protocol. Each UAV is equipped with a battery with an initial energy of18/720 Joules, which is the typical energy of two AA batteries.

In the simulation, UAVs rely on the CSMA/CA MAC protocol, without using RTS/CTS messages and retransmissions. In case of buffer overflow, UAVs consider a drop tail mechanism to drop packets. At the application layer, UAVs take into account a QoE-aware redundancy mechanism to add redundant packets only to priority frames [5]. We have conducted simulations with two setups: SD-UAVNet, and XLinGO-MobiFANET. Particularly, XLinGO-MobiFANET considers XLinGO [5] as routing protocol coupled with the MobiFANET [22] relay placement mechanism. XLinGO-MobiFANET takes all decisions in a distributed fashion. On the other hand, SD-UAVNet establishes the route data path with relay location optimization.

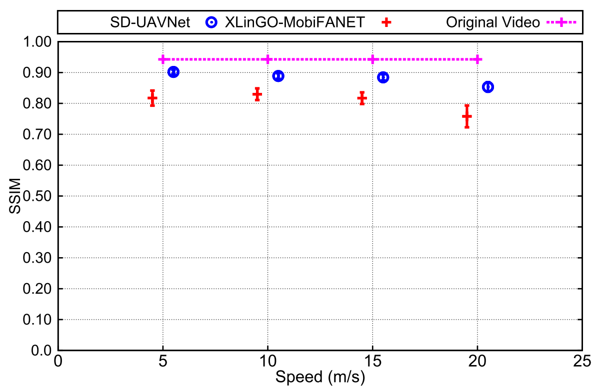
We considered video sequences with different video features downloaded from the YUV video trace library and YouTube [27], *i.e.*, Container, UAV_1, and UAV_2. Specifically, the Container video has similar characteristics as a UAV hoovering in a given area to capture the video, which means that there is a small moving region of interest on a static background. On the other hand, UAV_1 and UAV_2 videos are captured from a UAV flying in a city and in a rural environment, but UAV_2 has a higher motion level than UAV_1 caused by UAV instability during the flight. We encoded those videos with a H.264 codec at 300 kbps, 30 frames per second, Group of Pictures(GoP) size of 20 frames, and common intermediate format (352 x 288 pixels). The decoder applies a Frame-Copy method for error concealment to replace each lost frame with the last received one, reducing frame loss and maintaining the video quality.

In terms of video quality evaluation, QoS schemes alone are not enough to assess the quality level of multimedia applications, because they fail in capturing subjective aspects of video

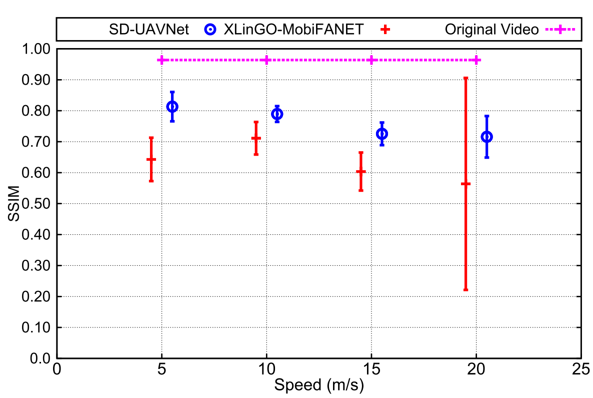


(a) Number of Transmitted Packets by Each Relay (b) Number of Received Packets by the Destination Node

Fig. 6. Impact of node mobility on the number of transmitted packet by each relay node and the number of packets received at the destination for each video transmission.



(a) Container (b) UAV_1

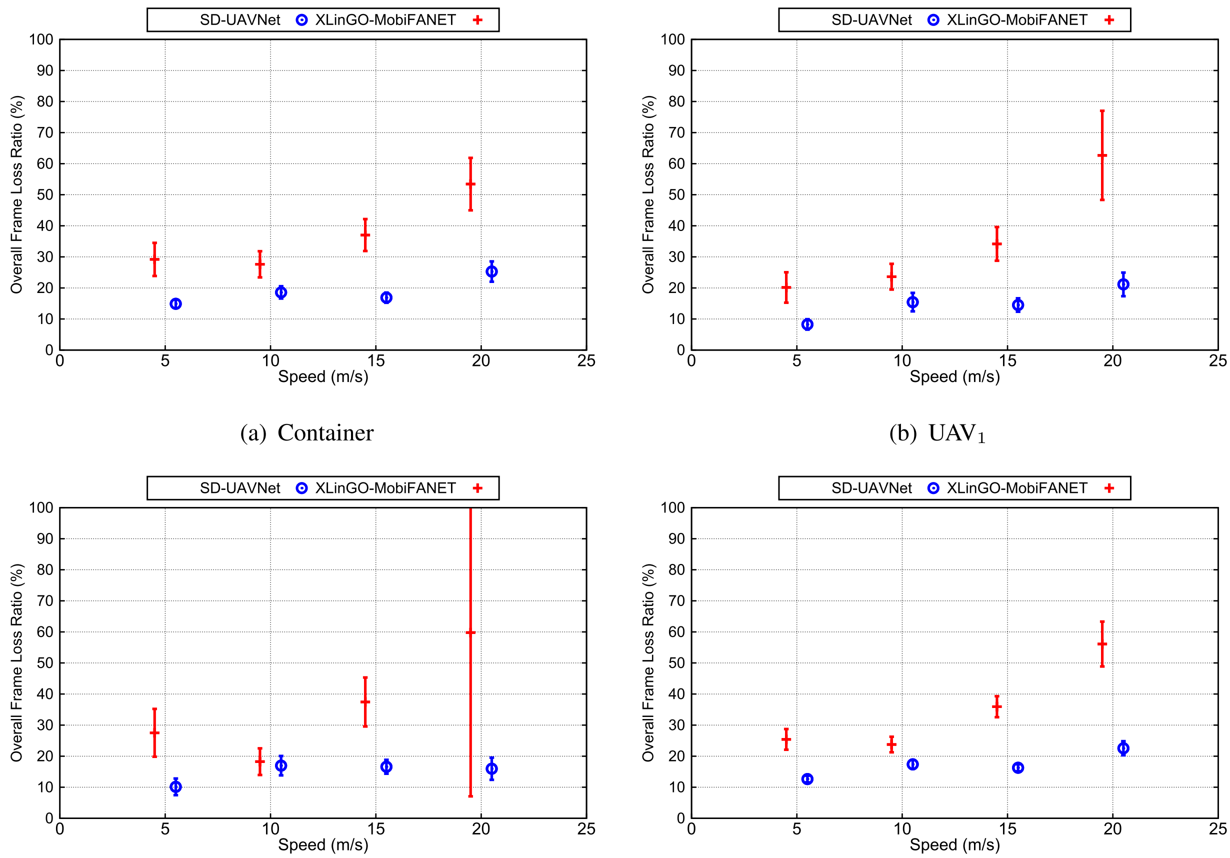


(c) UAV_2 (d) All Videos

Fig. 7. SSIM of UAVs flying at different speeds and different video transmissions.

content related to human experience [28,29]. QoE metrics overcome those limitations, and thus we rely on two well-known objective QoE metrics, namely Structural Similarity (SSIM) and Video Quality Metric (VQM). Specifically, SSIM $\in$ [0,1] is based on a frame-by-frame assessment of three video components, i.e., luminance, contrast, and structural similarity. Higher SSIM values mean better video quality. On the other hand, VQM $\in$ [0,4] measures the “perception damage” of video experienced based on features of the human visual system, namely blurring, noise, color distortion and distortion blocks. For VQM, a value closer to 0 means a video with a better quality. We used the MSU Video Quality Measurement Tool (VQMT) to measure the SSIM and VQM values for each transmitted video.

We consider the number of transmitted packets per hop and the number of received packets at the *DN*, in order to evaluate the effects of UAV mobility on the packet transmission. Finally, we consider the number of nodes alive and the average remaining energy of alive nodes to measure the effects of energy consumption of UAV flying and packet transmissions.



(a) Container (b) UAV_1 (c) UAV_2 (d) All Videos

Fig. 8. Frame loss for UAVs flying at different speed and different video transmissions.

4.2. Simulation results

Fig. 5 shows the number of UAVs alive and the remaining energy for UAVs flying with speed ranging from 10 to 15 m/s, and from 15 to 20 m/s for both SD-UAVNet and XLinGO-MobiFANET. By analyzing the results of Fig. 5(a), we can conclude that the first UAV finishes its energy source at simulation time 70 seconds for UAVs using XLinGO-MobiFANET with speeds ranging from 10 to 15 m/s. This is because UAVs spent energy to fly and also to transmit packets. In this sense, in XLinGO-MobiFANET, all decisions are made in a distributed fashion without considering energy for decision making and to make a fair distribution of energy resources. In addition, when UAVs are running the XLinGO-MobiFANET protocol with movement speeds ranging from 10 to 15 m/s, 22 UAVs are alive when the experiment is finished. On the other hand, all 30 UAVs finished the experiment alive when running the proposed SD-UAVNet architecture, as shown in Fig. 5(a). Additionally, UAVs finished the simulation with 30% more remaining energy with SD-UAVNet compared to XLinGO-MobiFANET, as shown in Fig. 5(b). Finally, as soon as the speed limits increase, the results become even worse, since UAVs spent more energy to fly at high speed limits, as depicted in Figs. 5(c) and 5(d). With SD-UAVNet, all relay nodes follow the movement instructions given by the *Controller Node*. Therefore, they have more efficient and optimized movement trajectories, which leads to longer flying time.

Fig. 6 shows the impact of UAV speeds on the number of transmitted packets by each hop and the number of received packets at the *DN* for each video transmitted by SD-UAVNet and XLinGO-MobiFANET. By analyzing the results, we can see that SD-UAVNet established a reliable data path between source and destination, regardless of the speed limits. For instance, relay nodes transmitted around 800 packets, and the destination received around 700 packets with SD-UAVNet. On the other hand, XLinGO-MobiFANET has 30% less transmitted packets by the relay nodes, and 20% less received packets at the destination compared to SD-UAVNet. This is because SD-UAVNet always selects source and relay nodes closer to the ideal location, *i.e.*, the ones that take less time to arrive, reducing the effects of UAV mobility on the packet transmission. In addition, SD-UAVNet considers energy to create the data path, and also the controller assigns another relay as soon as a given relay node is running out of its energy. These issues are not taken into account in XLinGO-MobiFANET, which leads to increased packet loss due to the effects of UAV mobility. For instance, the performance of XLinGO-MobiFANET for UAVs flying at speed ranging from 15 to 20 m/s is even worse, since UAVs will run out of energy during video transmission, and XLinGO-MobiFANET does not consider relay replacement.

Fig. 7 shows the SSIM for each video delivered via SD-UAVNet, XLinGO-MobiFANET, and the original video in a scenario of UAVs flying at different speed limits. The original video in the plot represents an error-less video transmission, which is used as a benchmark for video quality, since there are no SSIM values higher than it. This is because video coding/decoding operation itself introduces video quality impairments [30]. In addition, this maximum value is different for each video, due to the motion and complexity level of each video. Thus, this maximum SSIM value helps to see exactly the quality loss due to packet loss in data transmission.

By analyzing results of Figs. 7(a), 7(b), and 7(c), we observe that SD-UAVNet delivered Container, UAV_1, and UAV_2 video sequences with a better quality than XLinGO-MobiFANET, regardless of the moving speed and video type. In addition, videos delivered by SD-UAVNet have reduced SSIM values compared to the original videos, which are the video quality benchmark. Specifically, SD-UAVNet delivered the Container and UAV_1 videos with a SSIM value of 5% lower than the original video, and UAV_2 video with a SSIM value of 20% lower than the original video. This is because the frame loss on the Container video has lower impact on the quality level compared to UAV_1 and UAV_2, since the Container video has a low motion level. In addition, the videos delivered by XLinGO-MobiFANET have a higher variation, since some relay nodes ran out of their energy during the video transmission, which leads to significant packet losses. For instance, XLinGO-MobiFANET delivered only 5 videos for UAVs flying at speed ranging between 15 and 20 m/s, due to the energy issue.

Fig. 7(d) shows the SSIM values for all videos transmitted via SD-UAVNet and XLinGO-MobiFANET. We observe that SD-UAVNet delivered videos with high and constant SSIM values compared to XLinGO-MobiFANET. Specifically, videos transmitted in SD-UAVNet increased the SSIM values by 15% compared to XLinGO-MobiFANET. This is because the controller node established reliable routes for video transmission in SD-UAVNet, as explained by the results of Fig. 6. On the other hand, videos delivered via XLinGO-MobiFANET have a poor performance due to the effects of UAV mobility and energy consumption.

Fig. 8 shows the frame loss rate for each video delivered via SD-UAVNet and XLinGO-MobiFANET, which helps to explain the SSIM results depicted in Fig. 7. This is because video dissemination requires low frame loss to provide QoE support [29]. For instance, SD-UAVNet reduced the overall frame loss by 50% compared to XLinGO-MobiFANET. Particularly, a video is composed of I-, P- and B- frames, which have different priorities, and the loss of priority frames causes severe video distortions [30]. More specifically, the loss of an I-frame causes error propagation in other frames within a Group of Picture (GoP), due to the decoder uses the I-frame as a reference for reconstruction of other frames within the same GoP [5]. In this way, the QoE only recovers when the decoder receives an unimpaired I-frame. On the other hand, for the loss of a P-frame, the impairments extend to the remaining frames within a GoP, where the loss of P-frames at the beginning of a GoP causes higher video distortion than loss at the end of a GoP. The loss of a B-frame only affects the video quality of that particular frame. Based on the simulation results, we concluded that videos delivered via SD-UAVNet reduced losses of I-frames by 60% compared to XLinGO-MobiFANET. In this way, it transmitted priority frames with high deliver probability, which increases the video transmission quality.

5. Conclusions

This article proposed the Software-Defined UAV networking architecture, called SD-UAVNet. In our framework, the controller collects the UAV networking topology information, and optimizes UAV nodes' locations by considering the global UAV contextual information of UAV energy level, overall UAV movement distance, UAV collision avoidance, etc. With the proposed framework, we focus on the problem of using SD-UAVNet to provide live video surveillance under the disaster recovery scenario. The experimental results show that the proposed SD-UAVNet architecture can effectively mitigate the challenges of UAVNet and it provides satisfactory QoE to control center end-users.

In the future, we plan to extend our framework to include more specific tasks, such as UAV collaborative searching with multiple source inputs, remote object tracking, etc. All the relevant SD-UAVNet controller operations will also be defined and implemented. To address the system scalability, a hierarchical controller architecture will also be implemented to include multiple controllers to support large-scale application scenarios. Our vision is to provide a framework that support autonomous UAV behavior adaptation based on the observed scenarios.

Acknowledgment

This work is supported by CNPq (431474/2016-8) and EU COST Action CA 15127.

References

[1] Y. Zeng, R. Zhang, T.J. Lim, Wireless communications with unmanned aerial vehicles: opportunities and challenges, IEEE Commun. Mag. 54 (5) (2016) 36–42.
[2] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub, M. Anwander, UAVNet: A mobile wireless mesh network using unmanned aerial vehicles, in: Globecom Workshops (GC Wkshps), 2012 IEEE, IEEE, 2012, pp. 1603–1608.
[3] L. Pimentel, D. Rosário, E. Cerqueira, T. Braun, A. Abelem, M. Gerla, Context-aware adaptation mechanism for video dissemination over flying ad-hoc networks, in: Proceedings of the IFIP Wireless Days (WD), IEEE, 2014, pp. 1–3.
[4] L. Gupta, R. Jain, G. Vaszkun, Survey of important issues in uav communication networks, IEEE Comm. Sur. Tutor. 18 (2) (2016) 1123–1152, doi:10.1109/COMST.2015.2495297.
[5] D. Rosário, Z. Zhao, A. Santos, T. Braun, E. Cerqueira, A beaconless opportunistic routing based on a cross-layer approach for efficient video dissemination in mobile multimedia IoT applications, Comput. Commun. 45 (1) (2014) 21–31.
[6] D. Rosário, M. Seruffo, E. Cerqueira, C. Both, T. Braun, M. Gerla, Trends in human-centric multimedia networking scenarios, in: Proceedings of the 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), IEEE, 2016, pp. 1–5.
[7] W. Lobato, D. Rosario, M. Gerla, L.A. Villas, Platoon-based driving protocol based on game theory for multimedia transmission over VANET, in: proceedings of the IEEE Global Communications Conference (GLOBECOM), 2017, pp. 1–6.
[8] J.A. Wickboldt, W.P. De Jesus, P.H. Isolani, C.B. Both, J. Rochol, L.Z. Granville, Software-Defined networking: management requirements and challenges, IEEE Commun. Mag. 53 (1) (2015) 278–285.
[9] L. Tartarini, M.A. Marotta, E. Cerqueira, J. Rochol, C.B. Both, M. Gerla, P. Bellavista, Software-defined handover decision engine for heterogeneous cloud radio access networks, Comput. Commun. 11521–34, year = 2018, issn = 0140–3664, doi = https://doi.org/10.1016/j.comcom.2017.10.018,.
[10] Z. Zhao, E. Schiller, E. Kalogeiton, T. Braun, B. Stiller, M.T. Garip, J. Joy, M. Gerla, N. Akhtar, I. Matta, Autonomic communications in software-driven networks, IEEE J. Sel. Areas Commun. 35 (11) (2017) 2431–2445.
[11] H. Kim, N. Feamster, Improving network management with software defined networking, IEEE Commun. Mag. 51 (2) (2013) 114–119.
[12] A. Merwaday, I. Guvenc, UAV assisted heterogeneous networks for public safety communications, in: 2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2015, pp. 329–334.
[13] V. Sharma, M. Bennis, R. Kumar, UAV-assisted heterogeneous networks for capacity enhancement, IEEE Commun. Lett. 20 (6) (2016) 1207–1210.
[14] M. Mozaffari, W. Saad, M. Bennis, M. Debbah, Mobile internet of things: Can UAVS provide an energy-efficient mobile architecture? in: 2016 IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1–6.
[15] Current and future UAV military users and applications, Air Space Eur. 1 (5) (1999) 51–58.
[16] C.Y. Hans, G. Quer, R.R. Rao, Wireless SDN mobile ad hoc network: From theory to practice, in: Communications (ICC), 2017 IEEE International Conference on, IEEE, 2017, pp. 1–7.
[17] I. Ku, Y. Lu, M. Gerla, R.L. Gomes, F. Ongaro, E. Cerqueira, Towards software-defined VANet: Architecture and services, in: 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), 2014, pp. 103–110, doi:10.1109/MedHocNet.2014.6849111.
[18] P. Yang, X. Cao, C. Yin, Z. Xiao, X. Xi, D. Wu, Proactive drone-cell deployment: overload relief for a cellular network under flash crowd traffic, IEEE Trans. Intell. Transp. Syst. 18 (10) (2017) 2877–2892, doi:10.1109/TITS.2017.2700432.
[19] C. Di Franco, G. Buttazzo, Energy-aware coverage path planning of UAVS, in: Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference on, IEEE, 2015, pp. 111–117.
[20] C. Tseng, C. Chau, K.M. Elbassioni, M. Khonji, Flight tour planning with recharging optimization for battery-operated autonomous drones, CoRR abs/1703.10049 (2017).
[21] M. Mozaffari, W. Saad, M. Bennis, M. Debbah, Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage, IEEE Commun. Lett. 20 (8) (2016) 1647–1650.
[22] D. Rosário, J. Arnaldo Filho, D. Rosário, A. Santosy, M. Gerla, A Relay Placement Mechanism Based on UAV Mobility for Satisfactory Video Transmissions, in: Proceedings of the 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), IEEE, 2017, pp. 1–8.

[23] R. Magán-Carrión, R.A. Rodríguez-Gómez, J. Camacho, P. García-Teodoro, Optimal relay placement in multi-hop wireless networks, Ad Hoc Netw. 46 (2016) 23–36.
[24] O. Bouachir, A. Abrassart, F. Garcia, N. Larrieu, A Mobility Model for UAV Ad Hoc Network, in: Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2014, pp. 383–388.
[25] S. Hayat, E. Yanmaz, R. Muzaffar, Survey on unmanned aerial vehicle networks for civil applications: A Communications viewpoint, IEEE Communications Surveys Tutorials 18 (4) (2016) 2624–2661.
[26] D. Rosário, Z. Zhao, C. Silva, E. Cerqueira, T. Braun, An OMNeT++ Framework to Evaluate Video Transmission in Mobile Wireless Multimedia Sensor Networks, in: International Workshop on OMNeT++, ICST, 2013, pp. 277–284.
[27] Videos used for Simulations, 2015, Plus.google.com/117765468529449487870/videos.
[28] E. Liotou, D. Tsolkas, N. Passas, L. Merakos, Quality of experience management in mobile cellular networks: key issues and design challenges, IEEE Commun. Mag. 53 (7) (2015) 145–153.
[29] O.B. Maia, H.C. Yehia, L. de Errico, A concise review of the quality of experience assessment for video streaming, Comput. Commun. 57 (2015) 1–12.
[30] J. Greengrass, J. Evans, A.C. Begen, Not all packets are equal, part 2: the impact of network packet loss on video quality, IEEE Internet Comput. 13 (2) (2009) 74–82.

Zhongliang Zhao received his Ph.D. degree from the University of Bern in 2014. Since 2014, he holds an appointment of a Senior Researcher with the University of Bern. He was appointed as a work package leader on several work packages of the EU FP7 MCN project and a Co-Primary Investigator (PI) of the Sino-Swiss Science and Technology Cooperation (SSSTC) project M3WSN. Currently, he is the Technical Coordinator of the SNSF SwissSenseSynergy project and the co-PI of the Orange-funded industry project Context Awareness Engine.

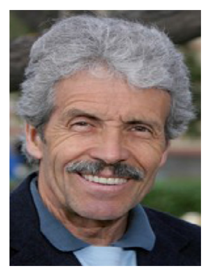
Pedro Cumino graduated in Computer Engineering at the Federal University of Pará, Brazil. Currently, he is doing a Master degree in Computer Science at the Federal University of Pará, Brazil. His current research interests include the following: Smart Cities, Internet of Things, FANET, Quality of Experience, and Software Defined Network.

Arnaldo Souza graduated in Information Systems at the Federal University of Pará, Brazil. Currently, he is doing a Master degree in Computer Science at the Federal University of Pará, Brazil. His current research interests include the following: FANET, Quality of Experience, and Software Defined Network.

Denis Rosário received his PhD degree in Electrical Engineering at the Federal University of Pará, Brazil with joint supervision undertaken by the Institute of Computer Science and Applied Mathematics of University of Bern, Switzerland in 2014. Currently, he is a Professor at Federal University of Pará. His current research interests include the following topics: Multimedia adaptation, Wireless Networks, FANET, VANET, Mobility, Quality of Experience, and Software Defined Network.

Torsten Braun got his Ph.D. degree from University of Karlsruhe (Germany) in 1993. From 1994 to 1995, he was a guest scientist at INRIA Sophia-Antipolis (France). From 1995 to 1997, he worked at the IBM European Networking Centre Heidelberg (Germany) as a project leader and senior consultant. Since 1998, he is a full professor of Computer Science at University of Bern. Currently, he holds an appointment of a vice president of the SWITCH (Swiss Research and Education Network Provider) Foundation since 2011. He was a Director of the Institute of Computer Science and Applied Mathematics at University of Bern between 2007 and 2011. He is serving as Deputy Dean of the Faculty of Science, University of Bern since 2017. He received best paper awards from LCN 2001, WWIC 2007, EE-LSDS 2013, WMNC 2014, and the ARMS-CC-2014 Workshop as well as the GI-KuVS Communications Software Award in 2009. In the scope of EU funded projects, he was leading WPs of FP6-EUQOS and FP7-MCN. Moreover, he coordinated national projects such as SNSF SwissSenseSynergy and SNSF CONTACT.

Eduardo Cerqueira received his Ph.D. in Informatics Engineering from the University of Coimbra, Portugal (2008). He is an associate professor at the Faculty of Computer Engineering of the Federal University of Pará (UFPA) in Brazil, as well as invited researcher at the Network Research Lab at UCLA-USA and Centre for Informatics and Systems of the University of Coimbra-Portugal. His publications include 5 edited books, 5 book chapters, 4 patents and over than 180 papers in national/international refereed journals/conferences. He has been serving as a Guest Editor for 6 special issues of various peer-reviewed scholarly journals. His research involves Multimedia, Future Internet, Quality of Experience, Mobility and Ubiquitous Computing.



Mario Gerla is a Professor in the Computer Science at UCLA. He holds an Engineering degree from Politecnico di Milano, Italy and the Ph.D. degree from UCLA. He became IEEE Fellow in 2002 is a recipient of the 2018 IEEE INFOCOM Achievement Award. At UCLA, he was part of the team that developed the early ARPANET protocols under the guidance of Prof. Leonard Kleinrock. At Network Analysis Corporation, New York, from 1973 to 1976, he helped transfer ARPANET technology to Government and Commercial Networks. He joined the UCLA Faculty in 1976. At UCLA he has designed and implemented network protocols including ad hoc wireless clustering, multicast (ODMRP and CodeCast) and Internet transport (TCP Westwood). His publications include over than 720 papers in national/international refereed journals/conferences. His current research projects cover: design and performance evaluation of protocols and control schemes for wireless networks; cloud computing; software defined networks; routing, congestion control and bandwidth allocation in wide area network.