Caching and D2D Sharing for Content Delivery in Software-Defined UAV Networks

Onur Kalinagac, S. Sinem Kafıloğlu, Fatih Alagöz
Dept. of Computer Engineering
Bogazici University
İstanbul 34342, Turkey
name.surname@boun.edu.tr

Gürkan Gür
Inst. of Applied Information Technology (InIT)
Zurich University of Applied Sciences (ZHAW)
Winterthur 8401, Switzerland
gueu@zhaw.ch

***Abstract*—In cases of catastrophic events such as natural disasters or physical calamities, current network infrastructure can become inoperative. Furthermore, there are transient events leading to excessive demand surges where it is needed to deploy additional network capacity on-demand. In such cases, rapid network deployments become vital to establish communications and enable networked services. Unmanned Aerial Vehicle (UAV) networks are good candidates for this kind of operation. Software-defined networking and content-centric operation are promising technologies to enable agile control, network visibility and efficient content delivery via centralized optimization in these challenged systems. In this work, we consider an edge network which is composed of UAVs and serves in a content-centric mode with in-network caching and device-to-device (D2D) transmissions. We develop a cache placement and selection scheme for energy-efficient operation. We also investigate how such a system performs under different operating conditions.**

I. Introduction

With the help of Unmanned Aerial Vehicles (UAVs), it is possible to deploy a local network infrastructure when there is a need for an easily deployable solution, e.g. in case of demand surges, transient network capacity requirements or post-disaster communications and data collection. For such a temporary network to be configured dynamically with diverse components, one promising approach is to utilize network softwarization where software-defined controller and a data layer allow flexibility and agile applications. This is also instrumental to enable fog or edge computing architectures where elastic resources and services at the edge are expected with complex control plane for distributed computation, storage and communications. Moreover, for many use cases, the content-centric operation emerges due to the content-heavy composition of supported networked services [1]. This is reflected with the recent surge of multimedia traffic in Internet. In that respect, device-to-device (D2D) paradigm is one of the techniques utilized for content-centric operations. It enables higher capacity by exploiting node proximity and better resource efficiency while QoS for multimedia delivery.

The current research works elaborate on different aspects of UAV, caching, software-defined networking (SDN) and D2D paradigm. Sampigethaya et al. [2] describes the opportunities and challenges for SDN in aviation sector. Regarding SDN accompanied with D2D communications, a study Usman et al. [3] uses D2D for public safety applications in 5G networks. There are many caching studies in wireless networks which basically deal with the distributed caching methods such as in small cell networks [4] or integrated resource management with caching in SDN [5]. In the study by Ji et al, fundamental limits of caching in wireless D2D Networks has been analyzed [6]. In [7], OpenCache caching controller in SDN is developed where you can define caching nodes and your caching algorithms on the controller. Another study which uses OpenCache has also promising results [8]. In [9], the heterogeneous networks (HetNets) are studied from content dissemination and D2D mode selection perspectives.

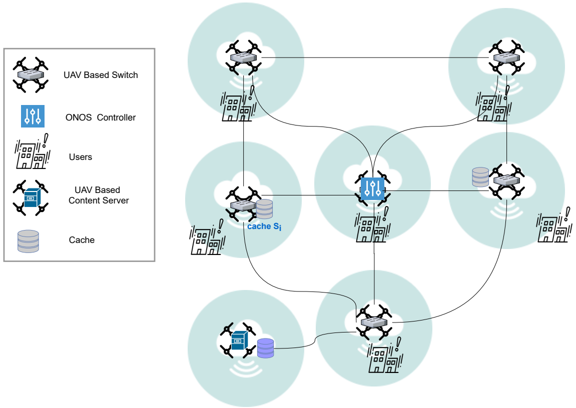


Fig. 1. Software-Defined UAV Network (SD-UAVNet) model.

In these content-centric and software-defined networks, a key challenge is how to improve performance while considering energy constraints by controlling in-network caching. In this work, we focus on this research problem and develop a cache placement and selection scheme for energy-efficient operation in Software-Defined UAV Networks (SD-UAVNets). We also investigate how such a system performs under different operating conditions.

II. System Model

Our network model consists of N_U UAVs (U_i) with M storage caches ($M < N_U$) denoted as S_i as shown in Figure 1. The connectivity degree for inter-UAV connections is D. Each UAV U_i has H hosts connected to it. Each UAV has cache capacity C_{UAV} while each host's capacity is C_H. The total content request rate generated by the network hosts

TABLE I
SYSTEM PARAMETERS.

Symbol	Definition
U_i	The i^{th} UAV
N_U	The number of UAVs
S_i	The i^{th} cache
M	The number of caches
D	The connectivity degree for inter-UAV connections
H	The number of hosts connected to each UAV
T	The number of pairwise content requests between hosts
C_{UAV}	The cache capacity of a UAV
C_H	The cache capacity of a host
λ	The content request rate
N_C	The number of contents
s_j	The size of content j
Communication Channel Model Parameters	
η_{LoS}	The additional LoS attenuation factors due to the LoS connections
η_{NLoS}	The additional NLoS attenuation factors due to the NLoS connections
f	The system carrier frequency
m, n	A2G channel parameters
$\theta_i(t)$	The elevation angle
$P_{LoS,i}(t)$	The probability of LoS connection
$P_{NLoS,i}(t)$	The probability of NLoS connection
$P_{i,H}$	The average received power of host from UAV i
$L_{i,H}(t)$	The link capacity between UAV i and host at time t
$L_{i,j}(t)$	The link capacity between UAV i and UAV j at time t
ρ	U2U path loss exponent
BW	Channel bandwidth
Energy Consumption Model Parameters	
$E_i(t)$	The residual energy level of U_i at time t
E_{in}	The initial energy level of a UAV
P_{fly}	The fly power consumption of a UAV
P_{tx}	The transmission power consumption of a UAV
P_r	The read power consumption of a UAV
t_{fly}	The flying time passed from the system boot of a UAV
t^i_{tx}	The transmission duration for content services of U_i
b^i_w	The total number of bytes written into the U_i cache
b^i_r	The total number of bytes read from the U_i cache
Mobility Model Parameters	
$\mathbb{F}$	The movement stepsize for UAVs. ($\in \{-1, 0, 1\}$ m)
P	Time period for UAV movement

is λ and there are N_C many contents in the system each having size s_j. The notations of the system, communication channel, energy consumption and mobility models are given in Table I. For the cache replacement management, we apply a global-popularity based algorithm in all UAV and host caches where the least popular content is evicted for caching new arrivals. When a content is not reachable within the network boundaries, it is served from the content server which acts as the ultimate content provider.

Two different channel models for air-to-ground and air-to-air communications from [10] are used in our network model.

1) Air-to-Ground (A2G) Communication Channel Model: In time slot t, the LoS and NLoS pathloss from the i^{th} UAV U_i to the BS is given by $PL_{LoS,i}(t) = PL_{FS,i}(t) + \eta_{LoS}$ and $PL_{NLoS,i}(t) = PL_{FS,i}(t) + \eta_{NLoS}$ where $PL_{FS,i}(t) = 20log(4\pi d_{i,U2H}(t)f/c)$ and probability of LoS and NLoS connections are $P_{LoS,i}(t) = \frac{1}{1+m\times exp(-n(\theta_i(t)-m))}$ and $P_{NLoS,i}(t) = 1 - P_{LoS,i}(t)$, respectively.

Then, the average received power of host from U_i is $P_{i,H} = P_{tx} - \bar{P}L_i(t)/10$ where the average pathloss is

$$\bar{P}L_i(t) = P_{LoS,i}(t)PL_{LoS,i}(t) + P_{NLoS,i}(t)PL_{NLoS,i}(t), \tag{1}$$

Based on these quantities, the link capacity between a host and UAV is

$$L_{i,H}(t) = BW \times log_2(1 + SNR_{i,H}) \tag{2}$$

using $SNR_{i,H} = 10^{\frac{P_{i,H}-\sigma^2}{10}}$ and noise power variance σ^2.

2) Air-to-Air (A2A) Communication Channel Model: For U2U communication, free-space channel model is utilized. When U_i transmits signals to U_j, the received power at U_j is expressed as

$$P_{i,j} = P_{tx} + G_{tx} + G_{rx} - PL_{FS,ij}(t) \tag{3}$$

where $PL_{FS,ij}(t) = 10log(4\pi f/cd_{i,j}(t))^\rho$

This leads to SNR value $SNR_{i,j} = 10^{\frac{P_{i,j}-\sigma^2}{10}}$. Accordingly, the link capacity from U_i to U_j is

$$L_{i,j}(t) = BW \times log_2(1 + SNR_{i,j}). \tag{4}$$

3) Energy Consumption Model: The energy consumption in our UAVNet is composed of various components due to flying, transmission, read and cache write operation. The residual energy level of the i^{th} UAV U_i at time t is calculated as shown below:

$$E_i(t) = E_{in} - t_{fly} \cdot P_{fly} - t^i_{tx} \cdot P_{tx} - b^i_w \cdot P_w - b^i_r \cdot P_r \tag{5}$$

Although we do not control the trajectory or flight of the UAVs, we include the related energy consumption for the sake of completeness.

4) Mobility Model: The UAVs are assumed to be anchored to fixed locations. However, they have small-scale mobility where these aerial systems are assumed to hover around these points in a limited space. Specifically, they are assumed to be moving in three directions (x,y,z) with a movement step uniform-randomly selected from set $\mathbb{F} = \{-1, 0, 1\}$ m every P seconds.

III. PROBLEM FORMULATION

The problem that we address is the maximization of the network utilization in a software-defined UAV network. The decision variable for activating cache of the i^{th} UAV U_i is u_i. If U_i is decided to have an active cache u_i is assigned to one. Otherwise, it is equal to zero. The decision variable for storing content j on UAV i is defined as c_{ij}. If the content j is stored on UAV i c_{ij} is set to one. Else, it is assigned to zero. The list of parameters that we utilize in the problem formulation are given in Table II. Besides, we utilize indicator function 1_w that is assigned to one if w is true, zero otherwise. Now, let us present the optimization problem which will be solved by the controller application to decide on the cache placement at each period as follows:

TABLE II
OPTIMIZATION MODEL PARAMETERS.

Symbol	Definition
U_i	The i^{th} UAV
N_U	The number of UAVs
u = $[u_i]$	The decision variable for cache placement of UAVs
S_i	The i^{th} cache
N_C	The total number of contents
C_{UAV}	The cache capacity of a UAV
c = $[c_{ij}]$	The decision variable for content storages
$E_i(t)$	The residual energy level of U_i at time t
E_{in}	The initial energy level of a UAV
$\mathbb{N}_{\max}$	The maximum number of allowed UAVs for cache placement
$\mathbb{B}$	The total number of bits sent successfully
$\boldsymbol{R}$	The set of routes
λ	The content request rate

TABLE III
CPS^{2}DU PARAMETER/VARIABLE LIST.

Symbol	Definition
$E_i(t)$	The residual energy level of U_i at time t
$\mathcal{L}_i(t)$	The connectivity of the i^{th} UAV at time t
$L_i^S(t)$	The total link capacity of the i^{th} UAV at time t
$L_i^N(t)$	The total link capacity of the neighbours of the i^{th} UAV at time t
$L_i^H(t)$	The total number of hops to connect to content server at time t
w_s	The weight of link capacity
w_n	The weight of neighbour capacities
w_h	The weight of hop counts
$\mathbb{C}_i(t)$	The cache placement metric of the i^{th} UAV at time t
α	The energy coefficient
β	The connectivity coefficient
z	The punishment for content server
K_i^s	The starting cost of the i^{th} UAV cache
K_i^f	The cache filling cost of the i^{th} UAV
$\mathbb{N}_{\max}$	The maximum number of allowed UAVs for cache placement

$$\max_{u} \quad \mathbb{B}(u, \lambda, \boldsymbol{R}) \tag{6}$$

subject to:

$$c_{ij} \leq u_i \qquad \forall i, \forall j \tag{7}$$

$$\sum_{j=1}^{N_C} s_j \cdot c_{ij} \leq C_{UAV} \qquad \forall i \tag{8}$$

$$u_i \leq 1_{(0 < E_i(t))} \qquad \forall i \tag{9}$$

$$0 \leq \sum_{i=1}^{N_U} u_i \leq \mathbb{N}_{\max} \tag{10}$$

$$0 \leq E_i(t) < E_{in} \qquad \forall i \tag{11}$$

$$u_i \in \{0, 1\} \qquad \forall i \tag{12}$$

$$c_{ij} \in \{0, 1\} \qquad \forall i, \forall j \tag{13}$$

Our objective function $\mathbb{B}(u, \lambda, \boldsymbol{R})$ in (6) gives the number of bits transmitted successfully in a period that we want to maximize. Const. (7) allows a content to be cached only if that UAV is selected as a storage unit while (8) constraints the total size of items to be cached with the UAV cache capacity. With const. (9), we ensure that only UAVs having residual energy can be selected as cache placement units. Const. (10) reflects the fact that total number of UAVs selected for cache placement cannot exceed the maximum allowed by the system. Const. (11) ensures that the residual energy level of a UAV does not exceed the initial energy state. Finally, const. (12) and (13) define the variables. The optimal solution of this problem in a timely manner is computationally hard for practical systems such as energy-constrained UAVs. Therefore, we propose a heuristic approach to maximize utility of the network in one energy-lifetime (i.e. assuming no recharging or UAV replacement occurs). We devise our polynomial complexity heuristic in the next section.

IV. CACHE PLACEMENT AND SELECTION HEURISTIC FOR CONTENT-CENTRIC SD-UAVNET (CPS^{2}DU)

For the management of cache placement and selection in our SD-UAVNet architecture, we propose a low-complexity heuristic. First, we define required functions for the derivation of our figure of merit $\mathbb{C}_i(t)$ which provides the key metric for caching related decisions. A is the set of the neighbor UAVs of the U_i and B is the set of the neighbor of the UAVs$\in A$ at time t. We define the total link capacity of the U_i at time t $L_i^S(t)$ in (14). Similarly, the total neighbor link capacity of U_i at time t $L_i^N(t)$ is in (15). $L_i^H(t)$ is the shortest path hop count of U_i to the server.

$$L_i^S(t) = \sum_{j \in A} L_{i,j}(t) \tag{14}$$

$$L_i^N(t) = \sum_{i \in A} \sum_{j \in B} L_{i,j}(t) \tag{15}$$

The weighted sum of these functions in (16) is defined as the connectivity measure in our system. Note that the symbols used in the equations are listed in Table III.

$$\mathcal{L}_i(t) = w_s \cdot L_i^S(t) + w_n \cdot L_i^N(t) + w_h \cdot L_i^H(t) \tag{16}$$

Using $\mathcal{L}_i(t)$ and according to the energy capacity of U_i $E_i(t)$ and cache replacement cost related parameters (K_i^s, K_i^f), we calculate the key metric $\mathbb{C}_i(t)$:

$$\mathbb{C}_i(t) = \frac{(\alpha - z) \times E_i(t) + \beta \times \mathcal{L}_i(t)}{\chi + K_i^s + K_i^f} \tag{17}$$

In our caching algorithm periodically at time step t, we sort the cache placement metrics $\mathbb{C}_i(t)$ of all active UAVs in the system. We select first $\mathbb{N}_{\max}$ UAVs as active caching units. If there are less than $\mathbb{N}_{\max}$ many UAVs with residual energy, all of them are chosen to be active caching units. This cache placement process is devised at each period. In this way, we want to keep UAVs alive as long as possible while serving content requests properly. The algorithm pseudocode for cache placement is provided in Alg. 1.

For each request, the different potential content sources are determined by the content server. Among these candidates, the minimum-hop shortest paths are calculated as potential routes. Then for each of them, the overall cost metric $K_{route,i}$ is estimated by aggregating proportional battery losses of the UAVs due to transmission and possible read and cache write operations on the given path. Finally, the minimum cost route is selected which in turn determines the final source selection for that request as shown in Alg. 2.

Algorithm 1: Cache Placement in CPS^{2}DU

```
Actively caching switches array $CS$;
Cache placement period $T_{cp}$;
Number of actively caching switches $k$;
Number of working UAVs $n$;
$t = 0$;
$n = N$;
$k = M$;
$CS$ = [Randomly selected $k$ switches];
while $n > 0$ do
    Wait $T_{cp}$;
    $t = t + T_{cp}$;
    $n = getNumberOfWorkingUAVs()$;
    $k = min(n, M)$;
    for $i = 1$ to $N$ do
        if $UAV_i$ works at time $t$ then
            Calculate $\mathbb{C}_i(t)$;
        end
    end
    $CS$ = [$k$ switches with the largest $\mathbb{C}_i(t)$ scores];
end
```

Algorithm 2: Source Selection in CPS^{2}DU

```
Host a requires file m;
if File m is cached in other hosts or switches then
    Calculate $K_{route,i}$ for each source;
    $X$ = The source with the lowest $K_{route,i}$;
    return $X$
else
    return The content server
end
```

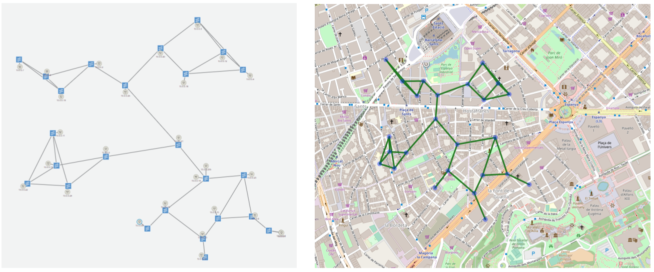
V. System Implementation

To run our tests, we use a machine with Intel Core i7-5600U (2.60GHz × 2), 20GB RAM and 256GB SSD storage to host two virtual machines (VM1 and VM2). Each VM has 4GB memory and 20GB HD space. We use ONOS Raven 2.1 on Ubuntu 16.04 as SDN controller on VM1. Mininet 2.2.2 is used for creating network topologies on Ubuntu 14.04 LTS with Open vSwitch 2.0.2 on VM2. For traffic generation from hosts on VM2 in our experiments, iPerf 2.0.5 is adopted [11]. For developing the network control and content delivery applications, Python 2.7.6 is used and they are deployed on VM2.

There are three main application modules for implementing and running simulations for our content delivery system in SD-UAVNet environment. The main application named `systemDriver` which configures the system on start for tests and cleans it on exit. After building the topology, the file request listener applications are called on the related hosts. Its secondary task is creating content request commands and sending them to a random hosts terminals to be executed so that we have a controlled test run.The `contentServer` application stores the request statistics and which hosts have which files types information. It also controls the cache management in the system. It also provides caching related descriptive statistics and live logs for performance monitoring and analysis. The `HostCacher` application handles the redirected content requests and utilizes iPerf to generate the predefined traffic from the selected source to the file-requesting host.

TABLE IV
KEY SIMULATION PARAMETERS.

Parameter	Value
Zipf parameter (α)	1.1
Mean content request rate ($1/\lambda$)	1.5 sec^{-1}
N_U	20
Content size (randomly distributed)	8-12MB
N_C	1000
C_H	50MB
C_{UAV}	100MB
E_{in}	12Ah
E_{in} of content server connected UAV	18Ah
P_{fly}	11W
P_{tx}	0.1W
P_r	0.03W/25MBps
P_w	0.01W/25MBps
$\mathbb{N}_{\max}$	0.5×N_U
σ^2	−84dBm
f	2.4GHz
ρ (air-to-air pathloss coefficient)	2
η_{LoS}, η_{NLoS}	1dBm, 20dBm



(a) Network topology.

(b) Nodes as located in Barcelona topology.

Fig. 2. A sample network topology in the simulations and the visualization of topology elements in Barcelona network.

For our experiments, we have used randomly sampled network segments from Barcelona topology as shown in Figure 2 [12]. Each down-sampled topology was selected for having a reasonable size and geographical distribution as UAVNet. It includes UAVs acting as switches with one entailing the controller as well as a content repository. There are 20 Mininet OpenFlow switches (UAV based switches) and 20 Mininet hosts (users) with one Mininet host as content server (UAV Based Content Server). The specific content requests are determined according to Zipf distribution. Then these content requests are generated according to Poisson distribution with parameter λ.

VI. Performance Evaluation

For performance investigation, simulation results from our experiments are given in Table V. In the table, system control parameters are highlighted with the same color for distinguishing between the changed parameters in each group. Additionally, the configuration IDs are defined on the first

TABLE V

Simulation Results for Different System Configurations.

	$\mathbb{C}_i(t)$				$\mathcal{L}_i(t)$					**Composition of Content Source Selection**			
ID	α	β	$\mathbb{N}_{\text{max}}$	$1/\lambda$	w_s	w_n	w_h	Th$_{avg}$(Mbps)	Data$_{tx}$(MB)	Content Server	Local Hit	Switch	Other host
1	0.5	0.5	10	3	0.1	0.6	0.3	25.73	3020	40%	21%	28%	11%
2	0.5	0.5	10	3	0.7	0.1	0.2	25.12	3133	41%	20%	24%	15%
3	0.5	0.5	10	3	0.1	0.2	0.7	25.46	3234	41%	18%	29%	12%
4	0.5	0.5	10	1.5	0.1	0.2	0.7	25.65	5040	37%	23%	27%	13%
5	0.5	0.5	10	1	0.1	0.2	0.7	25.94	6171	35%	25%	30%	9%
6	0.8	0.2	10	1	0.1	0.2	0.7	24.72	6271	36%	27%	25%	12%
7	0.2	0.8	10	1	0.1	0.2	0.7	26.42	6194	37%	25%	27%	11%
8	0.5	0.5	4	1	0.1	0.2	0.7	21.81	5746	40%	24%	15%	21%
9	0.5	0.5	16	1	0.1	0.2	0.7	24.75	6187	35%	26%	35%	5%
10	0.5	0.5	10	1	0.1	0.2	0.7	23.49	5962	37%	24%	36%	4%

column. For instance, the blue shaded cells mean that these parameters are varied while the other ones such as α and β are kept the same in the corresponding configurations (C1-C3). Th_{avg} gives the total system throughput whereas Data$_{tx}$ defines the total transmitted data until all UAV energies are depleted.

First, we elaborate on how the content request rate parameter λ affects the system performance. With increasing λ (increasing load), the content server selection rate (the percentage of requests which are served from the content server) decreases. Next, we inspect the maximum number of UAVs that can be used for cache placement. The major improvement in the performance is observed for varying $\mathbb{N}_{\text{max}}$. With increasing $\mathbb{N}_{\text{max}}$, the number of activated UAVs for the cache placement gets larger. In that case, selection rates for the content server and hosts decrease while the switch selection rate (i.e. UAVs) increases. Hence, if a content is not served locally, it will be served from some active UAV switch with greater probability as the number of such UAVs increases. The host selection rate decreases since the availability of content in a UAV close to the requester is more likely, making the cases where a host provides the content less frequent. In Configuration 10 (C10) given on the last row in Table V, a baseline source selection technique is applied, which only selects the source according to the minimum hop count to requesting host. In that case, there is a slight performance decrease in terms of average throughput Th_{avg} and Data$_{tx}$ compared to configurations with ids C5, C6, and C7 where $\mathbb{N}_{\text{max}}$ and mean content request rates are the same. As shown in Fig. 3, UAV battery levels monotonically drop over time in both C4 and C5 as expected (no recharging assumed in our model)[1]. In C5, higher request rate λ compared to the C4 results in the situation where central switches deplete their energy reserves rapidly.

VII. Conclusion

In this work, we develop an energy-aware cache placement and selection scheme for a SD-UAVNet which operates in a content-centric mode with in-network caching. We also investigate how such a network performs under different system configurations.

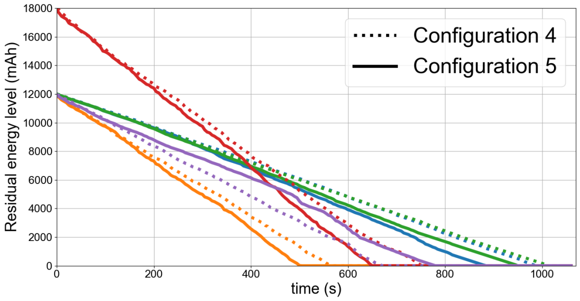


Fig. 3. UAV battery level changes in 4th&5th configurations (C4 and C5) for randomly selected 4 UAVs and content server (red lines) in each configuration.

References

[1] G. Gür, "Energy-aware cache management at the wireless network edge for information-centric operation," *Journal of Network and Computer Applications*, vol. 57, pp. 33 – 42, 2015.

[2] K. Sampigethaya, "Software-defined networking in aviation: Opportunities and challenges," in *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, pp. 1–21, April 2015.

[3] M. Usman, A. A. Gebremariam, U. Raza, and F. Granelli, "A software-defined device-to-device communication architecture for public safety applications in 5G networks," *IEEE Access*, vol. 3, pp. 1649–1654, 2015.

[4] E. Bastug, *Distributed caching methods in small cell networks*. Theses, Université Paris-Saclay, Dec. 2015.

[5] R. Huo, F. R. Yu, T. Huang, R. Xie, J. Liu, V. C. M. Leung, and Y. Liu, "Software defined networking, caching, and computing for green wireless networks," *IEEE Communications Magazine*, vol. 54, pp. 185–193, November 2016.

[6] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Transactions on Information Theory*, vol. 62, 05 2014.

[7] M. Broadbent, D. King, S. Baildon, N. Georgalas, and N. Race, "Opencache: A software-defined content caching platform," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–5, April 2015.

[8] P. Georgopoulos, M. Broadbent, B. Plattner, and N. Race, "Cache as a service: Leveraging SDN to efficiently and transparently support video-on-demand on the last mile," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, Aug 2014.

[9] G. Gür and S. Kafıloğlu, "Layered content delivery over satellite integrated cognitive radio networks," *IEEE Wireless Communications Letters*, vol. 6, pp. 390–393, June 2017.

[10] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X Communications: Design and Optimization for Multi-UAV Networks," *arXiv e-prints*, p. arXiv:1801.05000, Jan 2018.

[11] NLANR (National Laboratory for Applied Network Research)/DAST, "iPerf Tool." Available at https://iperf.fr/, 2019.

[12] A. Neumann, E. López, and L. Navarro, "An evaluation of bmx6 for community wireless networks," in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 651–658, Oct 2012.

[1] Since the content server bearing switch is heavily used due to its function, it was assumed to have a higher initial battery capacity.