Vehicular Communications 32 (2021) 100391

Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom

A cooperative approach for content caching and delivery in UAV-assisted vehicular networks ☆

Ahmed Al-Hilo, Moataz Samir, Chadi Assi *, Sanaa Sharafeddine, Dariush Ebrahimi

Concordia University, Canada

ARTICLE INFO

Article history:
Received 22 April 2021
Received in revised form 28 June 2021
Accepted 21 July 2021
Available online 30 July 2021

Keywords:
UAVs
Vehicular networks
Reinforcement learning
Optimization
Content caching

ABSTRACT

Given the advent of Intelligent Transportation Systems (ITSs), drivers and passengers could now spend more of their time enjoying entertainment applications, e.g., watching TV or streaming movies. However, such services can drastically increase the traffic load on the existing network infrastructure (i.e. Roadside Units (RSU) and Cellular Base Stations (CBS)). Recently, Unmanned Aerial Vehicles (UAVs) have been playing a remarkable role in offloading terrestrial networks and providing cellular services thanks to their agility and flexibility. Hence, this paper explores a cooperative approach for content caching and delivery in the context of internet of connected vehicles, where a RSU, having access to a library of contents but with limited communication coverage, collaborates with a UAV to deliver contents to vehicles on a road segment. In this context, the connected RSU is responsible for delivering contents to the UAV cache unit by leveraging passing by vehicles. The RSU loads the contents on these vehicles that in turn upload them to the UAV cache unit. We model this cooperation problem mathematically as a mixed integer non-linear programming (MINLP) problem with the objective to maximize the number of served vehicles. Owing to the complexity of solving this problem, it is alternatively cast as an MDP whose solution is obtained through a Dual-Task Reinforcement Learning method (DTDRL). Simulation results show the superiority of our proposed collaborative solution over non-collaborative methods.

1. Introduction

The Internet of connected vehicles (IoCV) is an emerging technology evolved from vehicular ad-hoc networks (VANETs) to enable a wide range of digital services for vehicles and in-vehicle users. By following Internet of Thing paradigm, IoCV allows to accommodate sophisticated applications and services where network nodes act as smart objects by integrating humans, vehicles, and other objects [1]. Indeed, ubiquitous systems open new grounds in the transportation and automotive sector and vehicles are industrialized with on board communication and computing functionalities to stay connected with various external entities while driving [2].

1.1. Motivation

The demand by consumers for online contents is steadily increasing. A newly published report by Ericsson shows that futuristic cellular average monthly data traffic will triple over the course of the next five years [3]. Hence, the massive data traffic of 5 G and beyond will impose considerable pressure on the existing cellular infrastructure where entities such as Base Stations (BS) and Road side-units (RSU) will not be solely able to cope with it. Further, emerging technologies such as Virtual and Augmented Reality (VR/AR), 4K streaming, besides many others, add a substantial amount of traffic on the network and many of the emerging applications have stringent requirements such as low latency and demands for high data rates. Meanwhile, the rise of autonomous and tele-driving will allow passengers to entertain aboard their vehicles by streaming and watching Video on Demand (VoD) and consuming other digital contents. Consequently, VANETs will experience heavy traffic volumes specially during peak-hours or certain occasions when road densities are high [4]. Additionally, due to the increase in traffic volumes associated with emerging ITS applications and services, particularly content delivery, without a doubt the network will become a major bottleneck. Furthermore, the high mobility of vehicular environment makes it challenging to offer high quality of service to vehicles' users [5]. Therefore, effective solutions and technologies are required to bridge the gap.

Vehicle edge caching has been widely introduced in the literature to improve content delivery services in vehicular networks [6,2]. Caching can alleviate traffic loads and reduce latency across

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.
E-mail address: assi@ciise.concordia.ca (C. Assi).

https://doi.org/10.1016/j.vehcom.2021.100391
2214-2096/© 2021 Published by Elsevier Inc.

the network by avoiding duplicate content retrievals and transmitting contents from nearby units. Drones, or UAVs, on the other hand, are considered as appealing solutions to ameliorate vehicular network performance and overcome the key issues raised earlier, particularly with having backhaul bottlenecks. Given their agility, flexibility, and deployability, UAVs are seen as a solid candidate to help in offloading data traffic from the vehicular networks [7]. As opposed to terrestrial infrastructure, UAVs can better communicate with vehicles as they possess the ability to fly and follow moving objects. UAVs are also known to have much efficient links with higher probability of creating Line of Sight (LoS) since they can adjust their positions to establish better links by reducing distances and avoiding obstacles such as skyscraper and towers. Therefore, in this work, we leverage UAVs and caching principles to design a new paradigm where a UAV is deployed to cooperate and assist a RSU in serving contents [8].

Building upon our previous work [9], we further extend that work by considering a cooperative cache-based system. In this article, we address the issue of accessing popular contents for in-vehicle users. More specifically, we assume that a RSU and a UAV co-exist to serve contents to vehicles navigating on a road segment. The UAV acts as a complement infrastructure to the RSU in a sense of completing services for partially served vehicles and/or probably provide full service for un-served ones. To do so, the UAV is assumed to be equipped with finite cache capabilities to store and serve contents. In addition, due to the complexity of establishing an efficient backhaul link to the UAV, we come up with an alternative method [10]. In a nutshell, we design a relay scheme where the RSU can populate contents on the UAV over the passing vehicles in push-carry-forward fashion.[1] In this way, the loads can be reduced on the backhaul besides maintaining the content list stored on the UAV during the operational phase.

1.2. Challenges

In this article, we identify three main challenges. The first challenge is due to radio resource scheduling. For example, when the RSU serves a vehicle, it has to consider the ability of the UAV to complete the remaining part. This includes answering two questions; whether the requested content is cached on the UAV and whether the UAV has enough resources (spectrum and time) to serve the rest of the contents. The latter question is based on the UAV status (e.g., number of vehicles within its coverage and still need service). More importantly, the RSU should not reject requests while wasting its resource to serve vehicles which can be served through the UAV at a later time. The RSU may also want to populate contents on the UAV cache by serving some vehicles till completion which can subsequently forward/shuttle these contents to the UAV. The second challenge is related to the UAV. The UAV needs to plan its trajectory, radio resources, and decide for contents placement (which contents and how much to cache for each one). The UAV should take into account popularity of contents, and vehicles status (how much each vehicle has been served through the RSU). If a vehicle has been already served partially through the RSU and the UAV is capable of completing the service, the UAV has to satisfy that vehicle. Otherwise, the service is rejected and the RSU would have wasted resources on incomplete tasks. Hence, the overall performance of the network degrades. A third challenge appears as a result of the unknown parameters of the environment. There is a number of uncertainties in our context including the arrivals of upcoming vehicles, their velocities, and their requests and such information is not given in advance.

[1] It is worth noting that the cooperation of vehicles can be motivated by offering some incentive, however, this works focuses on vehicles that are welling to cooperate with the network operator.

In general, the interplay between the RSU and the UAV is of utmost importance. If it is not handled properly, there will be many vehicles leaving the road segment without being served sufficiently. Thus, in order to tackle the challenges mentioned above, we propose a solution approach, Dual-Task Deep Reinforcement Learning (DTDRL), based on Deep Reinforcement Learning (DRL) to control both the RSU and UAV. DRL is an effective method to deal with dynamic environments [11].

1.3. Contributions

This paper presents a scheme for serving vehicles through a cooperative system composed of an RSU and a UAV while populating the UAV cache by content delivered through the vehicles from the RSU. Our contributions are as follows:-

- We present and model a cooperative system where a deployed UAV aides an existing RSU to deliver contents for passing vehicles in a particular road segment. The UAV is used to offload some of the services provided by the RSU and hence enhance the overall network performance. To this end, an optimizer plans how to schedule the RSU and UAV radio resources besides planning the UAV trajectory and maintaining the UAV cache.
- We formulate the joint problem of UAV trajectory, RSU and UAV resource scheduling mathematically as an optimization problem to find the optimal solution that maximizes the concerned utility.
- Given the hardness of the problem and the existing uncertainties, we resort to a solution based on DRL. Thus, we formulate the problem as MDP taking into consideration two sets of actions namely, UAV mobility and RSU service.
- Based on the action space, we develop DTDRL method where the two actions are handled. Our solution approach can handle simultaneously RSU resource scheduling and UAV mobility.
- In addition, we design three effective algorithms to assist the DTDRL agent in allocating the resources of RSU downlink and UAV uplink and downlink.
- We conduct several extensive simulation based experiments using SUMO to study the behaviors of our proposed system and compare our solution approach with other baseline methods.

1.4. Organization

In Section 2, related works are presented. Section 3 explains our system model in details followed by Section 4 which formulate the problem mathematically. In Section 5, we discusses our solution approach. Then, Section 6 presents our numerical results and finally, Section 7 sums up the paper.

2. Related work

A wide spectrum of studies has been devoted to content delivery in vehicular networks for multitude of infotainment services. Some works are based on stationary infrastructures while others investigate the roles of aerial base stations such as UAVs. However, there exists only limited number of papers which address the collaboration among various types of facilities. Here, as this study seeks a collaboration among stationary and non-stationary entities, this section will be more concentrated on the studies that deal with the interplay between UAVs and fixed base stations.

The authors of [12] present a framework to help planning joint UAV and RSU networks for urban areas. The proposed framework takes into account the limited budget, UAV battery capacity, wireless coverage and motivates renewable energy. However, this work does not involve a cooperative planning strategy to deliver contents. In [13], the authors propose a new scheme for cooperative RSUs in order to serve vehicles while considering content popularity for caching purposes. The paper also considers service delay and content encoding technology. The authors in [14] propose a new scheme to jointly optimize content delivery and content caching in vehicular edge networks via deep deterministic policy gradient and formulate as double time-scale Markov decision process. In [15–17], the authors try to improve data dissemination in vehicular networks through considering different types of connections namely, vehicle to vehicle (V2V), vehicle to RSU (V2I), and vehicle to UAV (V2U). Indeed, these three papers focus more on reducing transmission latency rather than serving contents. Next, the authors of [18] provide a platform to enable UAV-assisted vehicular networks by cooperating with other transceivers, specifically, vehicles and RSUs. In spite of the several issues addressed by that work in vehicular networks which include dynamic topology, reliable connectivity and others, it does not elaborate on caching strategies to improve content delivery. Furthermore, the authors of [19] suggest software defined networks to support heterogeneous vehicular networks where RSU and BS are collaborating in multicasting contents to the users. The objective of this work is to improve the utility of each facility, RSU and BS, where each one partially contributes towards serving the demands of vehicles by offering incentives. In addition, the authors of [20] design a framework for cooperative RSU and high-altitude platforms (HAPs) where the latter broadcasts contents to users before the requests arise. In [21], the authors establish a content delivery networks for vehicular environments where vehicles exchange messages with each others and with external infrastructure. The authors in [22] study the importance of caching in vehicular networks and the benefits of reducing the traffic volume on the core networks. In [23], the authors integrate UAV with cluster-based vehicular networks to enhance the network performance in attaining higher packet delivery ratio, throughput and lower delays. In [24], a flooding technique is utilized to improve the reliability of data delivery in vehicular networks considering collaboration between vehicles and UAVs.

Other main limitations of the referenced works are summarized in the following two points. First, some of the works are only applicable for distributing small-size data that, in reality, does not significantly contribute to mitigate network bottlenecks. Second, the assumption of one data/file size may not represent a real scenario, besides, some of the above mentioned works provide a solution for only one static snapshot while the consecutive time slots are remain unaddressed. Unlike the works discussed above, the novelty of this current work concentrates around introducing a cooperative content service mechanism to improve the utility of RSU and UAV and provide efficient solution approach to control the different aspects of the system. Our work mainly focuses on leveraging the flexibility and mobility nature of the UAV besides providing a new method to relay contents to the UAV via passing vehicles.

3. System model

We consider the service of content delivery in vehicular networks as shown in Fig. 1. A set of vehicles, $I = \{0, 1, 2, ..., i, ..., |I|\}$, travel through the considered network and stream contents that are stored at the network edge on caches co-located with the infrastructure, e.g., RSU.[2] Now, since vehicles have only limited residence within the range of the RSU, the content to be streamed may not be completely served by the RSU alone; hence, in this work we exploit cooperative content delivery to offload an overwhelmed RSU with large data demands where a UAV is also leveraged to assist the network in delivering contents, partially or fully to passing by vehicles. The UAV is assumed to be flying within the service area and passing by vehicles are exploited by the RSU to store, carry and forward contents to update the cache at the UAV. Here, we assume that the UAV and RSU can exchange control information over an out of band signaling channel. Accordingly, the UAV through this channel informs the RSU regarding its cache status and request contents to be delivered through passing vehicles. The optimizer can then plan the best trajectory of the UAV.

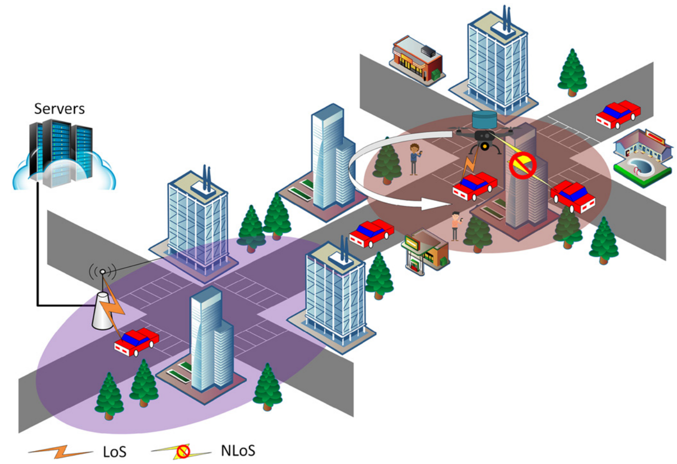

Fig. 1. System model.

Now, upon their deployment, UAVs will generally have limited access to the network through a robust backhaul link, and accordingly, updating the cache of the UAV directly from the network may be difficult. To overcome this issue, one possible solution is to do proactive caching ahead of deployment [25]; however, since the popularity of the content may change with time, and that content may be generated and consumed spontaneously particularly with the emergence of popular social media applications, dynamically adapting the content of the cache becomes critical for improving the quality of experience of end users. We advocate in this paper the cooperation of vehicles with the RSU to establish a link through which the RSU can make the necessary updates on the cache of the UAV; namely, vehicles will act as content mules and the RSU opportunistically will forward through them the contents it needs to deliver to the UAV. A time horizon of length N is considered and time is assumed to be slotted into smaller time slots $\{0, 1, 2, ..., n, ..., N\}$ where each time slot n is of length Δ sec. At each time slot n, there is a subset of vehicles $I^n \subseteq I$; some vehicles are within the range of the RSU and others could also be within the coverage of the UAV. Vehicles are assumed to arrive at the road segment according to a particular arrival rate. The RSU makes decisions to serve requests initiated by each vehicle to download a content, and the RSU also makes decisions to offload content delivery to the UAV through these vehicles. We can summarize the proposed scenario as follows:

- A flow of vehicles arrives to the coverage region of the RSU where each vehicle requests a content.[3]
- The RSU schedules the transmissions to serve vehicles, taking into account the requested content size, available radio resources and vehicles' residence times.

[2] For simplicity, we use one-way one-dimensional road segment, however, the same environment can be extended to multiple dimensions.

[3] For simplicity, we assume each vehicle requests one content.

- Given the short residence time of vehicles within the coverage of the RSU and the limited wireless resources of the RSU, the RSU may leverage a UAV to cooperate in delivering contents to the vehicles.
- It is assumed that vehicles leaving the coverage region of the RSU will pass through the coverage of the UAV.
- The UAV is assumed to be deployed with some cached contents; as explained, the RSU opportunistically selects some vehicles to transfer contents (and hence update) to the UAV by pushing the entire contents on the vehicles. The selected vehicles will *store* the contents in their local buffers and *carry* them until they become in contact with the UAV where they *forward* the buffered contents.
- To facilitate the communication between the UAV and passing by vehicles, the RSU manages the mobility of the UAV such that the UAV can fetch contents from vehicles to update its cache. Besides, the RSU tries to give sufficient contact time between the UAV and the vehicles to receive full service.[4]
- Once the UAV cache is loaded with contents, it is ready to participate in collaborating with the RSU in delivering contents to the vehicles. Thus, the RSU will take into consideration the ability of the UAV to serve the demands.

The RSU aims at satisfying the vehicles by serving the contents in whole before they leave the considered area [9].

3.1. Content model

We assume a library containing a particular number of contents (J), each of size ($Z_0, ..., Z_j, ..., Z_J$). The popularity of contents depicted by its access frequency which is assumed to be derived from a Zipf distribution with a skewness parameter α which characterizes the distribution.

3.2. Communication model

The RSU and the UAV are both assumed to be operating on the cellular spectrum and they communicate with vehicles using cellular VANETs [26]. We assume to use Time-Division Multiple Access (TDMA) where each time slot is allocated at most to one vehicle. Both infrastructures, RSU and UAV, are assumed to be capable of transmitting simultaneously. That said, each one can only communicate with vehicles within its coverage area. Typically, the communication channel between RSUs and end users is usually modeled as large- and small-scale fading. However, in the context of RSU where the RSU is situated such that there exists a clear link between transmitter and receiver [27]. Thus, the communication link can be characterized by strong line-of-sight. Consequently, the channel gain between the RSU and vehicle i at time slot n can be written as follows:

$$h^n_{R\to i} = h_0\left(\sqrt{(L^n_i - w^n)^2}\right)^{-\tau}, \forall i \in I^n, \forall n \tag{1}$$

where h_0 is the mean path gain at reference distance = 1 m and τ denotes path loss exponent. Also, w^n is the UAV location at time slot n and L^n_i denotes vehicle i location at time slot n.

Then, the instantaneous throughput of the RSU downlink to vehicle i can be formulated as follows:-

$$T^n_{R\to i} = W \log(1 + \frac{h^n_{R\to i} P_R}{\sigma^2}), \forall i \in I^n, n, \tag{2}$$

where W is the available bandwidth, P_R denotes the transmission power of the RSU, and σ is the thermal noise power.

For the UAV, we use channel model for the UAVs in urban area where high-rise buildings and other objects appear which could disturb the links between the UAV and receiving vehicles. Thus, we assume that the link propagation is characterized by both Line-of-Sight (LoS) and non Line-of-Sight (NLoS). Here, $S^n_{U\to i} \in \{\text{LoS, NLoS}\}$ indicates the state of the channel between the UAV and vehicle i at time slot n. The probability of having LoS link adopted in this paper is similar to [28]. Then, we can find the probability of channel states between the UAV and vehicle i.

$$Pr(S^n_{U\to i} = \text{LoS}) = \frac{1}{1 + \eta_1 e^{(-\eta_2(\theta^n_{U\to i} - \eta_1))}}, \forall i \in I^n, n, \tag{3}$$

where η_1 and η_2 are constant parameters of the environment. $\theta^n_{U\to i} = \frac{180}{\pi} \arctan(\frac{z_U}{d^n_{U\to i}})$ is the angle degree between vehicle i and the UAV at n. Meanwhile, z_U denotes the height of the UAV antenna and $d^n_{U\to i}$ is the horizontal distance between vehicle i and the UAV at time slot n. Moreover, the probability of having NLoS can be found from $Pr(S^n_{U\to i} = \text{NLoS}) = 1 - Pr(S^{i,n}_U = \text{LoS})$. Next, the channel power gain for each vehicle $i \in I^n$ at time slot n is computed as:

$$h^n_{U\to i} = \begin{cases} (D^n_{U\to i})^{-\beta_1} & S^n_{U\to i} = \text{LoS}, \\ \beta_2 (D^n_{U\to i})^{-\beta_1} & \text{otherwise}, \end{cases} \tag{4}$$

where $D^n_{U\to i}$ is the Euclidean distance between the UAV and vehicle i at time slot n; $D^n_{U\to i} = \sqrt{(d^n_{U\to i})^2 + z^2_U}$. β_1 denotes the path loss exponent and β_2 is the attenuation factor for NLoS. Thus $h^n_{U\to i}$ can also be rewritten as:

$$h^n_{U\to i} = p(S^n_{U\to i} = \text{LoS})(D^n_{U\to i})^{-\beta_1} + (1 - Pr(S^n_{U\to i} = \text{LoS}))\beta_2 (D^n_{U\to i})^{-\beta_1} \tag{5}$$

Then, the instantaneous throughput of the UAV downlink to vehicle i can be formulated as follows:-

$$T^n_{U\to i} = W \log(1 + \frac{h^n_{U\to i} P_U}{\sigma^2}), \forall i \in I^n, n, \tag{6}$$

where P_U is the UAV transmission power. Likewise, we can find the uplink throughput between vehicle i and the UAV at time slot n.

$$T^n_{i\to U} = W \log(1 + \frac{h^n_{U\to i} P_V}{\sigma^2}), \forall i \in I^n, n, \tag{7}$$

where P_V is the vehicle i transmission power. To maintain a free interference communication (i.e., the interference below a certain threshold), a minimum distance between the RSU and the UAV is considered. Therefore, interference-free communication is considered in this work. In other words, the trajectory of the UAV maintains a distance away from the RSU to keep the interference below the threshold. Also, in the case when they operate on the same spectrum, we assume that the UAV and RSU are equipped with directional antennas where the coverage range of each one does not overlap with that of the other or they communication quality should be greater than a certain threshold [29]. Next, we constrain the mobility of the UAV such that it does not get closer to the RSU. Also, as this work focuses on providing service to a certain road segment, the UAV is assumed to move only within the boundaries of that area.

Finally, the UAV position is determined by its trajectory where the UAV can move back, forth, or hover in its place at each time slot. Let $(w^{n+1} - w^n)$, the distance that UAV moves during one time

[4] The RSU controls the wireless resources of both the RSU and UAV, UAV mobility, and UAV cache units to serve vehicles.

slot, denote the UAV speed and direction at time slot n. For example, if $(w^{n+1} - w^n) > 0$, the UAV has moved forth, else, the UAV has moved back. The maximum UAV speed, V_{max}, is predetermined and should not be violated.

4. Mathematical formulation

This section formulates the system model described above mathematically. Let us define $x^n_{i,j}$ as a scheduling variable for the wireless resources of the RSU, where:

$$x^n_{i,j} = \begin{cases} 1 & \text{vehicle } i \text{ is served content } j \text{ by the RSU at time slot } n, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Likewise, we define y^n_i for the UAV wireless resources:

$$y^n_i = \begin{cases} 1 & \text{vehicle } i \text{ is served by the UAV at time slot } n, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Then, we define $v^n_{i,j}$ for the uplink resources from vehicle i to the UAV for content j:

$$v^n_{i,j} = \begin{cases} 1 & \text{vehicle } i \text{ is sending content } j \text{ to the UAV at time slot } n, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Now, we compute the total amount of content j served to vehicle i through the RSU at each time slot n ($U^n_{R\to i,j}$).

$$U^n_{R\to i,j} = x^n_{i,j} \Delta T^n_{R\to i}, \forall i \in I^n, n \in N, j \in J \tag{11}$$

where Δ is the length of one time slot and J is the library of contents. Similarly, we compute $U^n_{i\to U,j}$ which denotes the total amount to upload content j to the UAV through vehicle j during time slot n.

$$U^n_{i\to U,j} = v^n_{i,j} \Delta T^n_{i\to U}, \forall i \in I^n, n \in N, j \in J. \tag{12}$$

Then, we compute the amount served to vehicle i through the UAV during time slot n.

$$U^n_{U\to i} = y^n_i \Delta T^n_{U\to i}, \forall i \in I^n, n \in N. \tag{13}$$

Vehicle i is considered served if and only if it downloads its desired content fully from the RSU, UAV, or both. Here s_i indicates whether vehicle i is served or not as follows:

$$s_i = \begin{cases} 1 & \sum_{j=0}^{J} C_{i,j} Z_j \le \sum_{n=0}^{N} (\sum_{j=0}^{J} (U^n_{R\to i,j} C_{i,j}) + U^n_{U\to i}), \forall i \in I, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Next, we suppose that the UAV does not have the requested contents. Therein, we assume that the UAV has a certain cache capabilities that can be filled with contents until fullness.

Let us define $\zeta^n_{i,j} \in \{0,1\}$ as an indicator holding value 1 if content j on vehicle i is fully fetched by the UAV before or at time slot n and 0 otherwise.

$$\zeta^n_{i,j} = \begin{cases} 1 & Z_j \le \sum_{n'=0}^{n} U^{n'}_{i\to U,j}, \forall i \in I^n, n, j \in J, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Next, we introduce $m^n_j \in \{0,1\}$ as a binary variable where it is equal to 1 if content j is available on the UAV at time slot n and 0 otherwise. The value of m^n_j has three cases; it is either equal to 1 when the content is just fetched or 0 if it is removed in the previous time slot. The third case is that it remains the same if no change occurs on it.

$$m^n_j = \begin{cases} 1 & \sum_{i=0}^{I} v^n_{i,j} \zeta^n_{i,j} = 1, \forall n, j \in J, \\ 0 & f^{n-1}_j = 1, \\ m^{n-1}_j & \text{otherwise,} \end{cases} \tag{16}$$

where $f^n_j \in \{0,1\}$ is a decision variable that equals to 1 if content j is decided to remove at the end of time slot n and 0 otherwise. Next, we introduce $\hat{s}_{i,j} \in \{0,1\}$ which holds value of 1 if vehicle i has downloaded the entire content j from the RSU only.

$$\hat{s}_{i,j} = \begin{cases} 1 & Z_j \le \sum_{n=0}^{N} U^n_{R\to i,j}, \forall i \in I, j \in J, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Now, we can write the optimization problem as follows:

$$\max_{x,y,v,f,w} \quad \sum_{i=0}^{I} s_i \sum_{j=0}^{J} C_{i,j} Z_j \tag{18a}$$

$$\text{s.t.} \quad \sum_{n=0}^{N} y^n_i \le 1, i \in I^n, \tag{18b}$$

$$\sum_{n=0}^{N} \sum_{j=0}^{J} x^n_{i,j} \le 1, i \in I^n, \tag{18c}$$

$$\sum_{n=0}^{N} \sum_{j=0}^{J} v^n_{i,j} \le 1, i \in I^n, \tag{18d}$$

$$v^n_{i,j} \le \hat{s}_{i,j}, \forall n, i \in I^n, j \in J, \tag{18e}$$

$$y^n_i \le \sum_{j=0}^{J} m^n_j C_{i,j}, \forall n, i \in I^n, \tag{18f}$$

$$\sum_{j=0}^{J} m^n_j Z_j \le \eta, \forall n, \tag{18g}$$

$$f^n_j \le m^n_j, \forall j \in J, n. \tag{18h}$$

$$|w^{n+1} - w^n| \le V_{max}, \forall j \in J, n. \tag{18i}$$

$$|w^n - \varpi| \ge \Lambda, \forall n. \tag{18j}$$

Since we assume TDMA as transmission access then only one transmission is allowed at a time as in Constraint (18b) and (18c). Constraint (18e) limits vehicle i upload transmission for the UAV to the contents that has been downloaded fully from the RSU. Constraint (18f) is added to ensure vehicle i is not being served through the UAV if the latter does not possess the requested content. Constraint (18g) prevents violating the limited cache capacity of the UAV where η denotes UAV cache size. Also, constraint (18h) states that a content cannot be removed from the UAV cache if it does not exist in the cache beforehand. Constraint (18i) prevents violating UAV max speed. And finally, Constraint (18j) restrains UAV mobility to not cross a certain point such that no overlap occurs between the coverage ranges of the RSU and UAV where ϖ and Λ denote RSU position and the minimum distance between the two entities, the RSU and UAV, respectively.

Now, if we look at the objective function with its constraints, we can see that the presented problem contains several binary variables alongside a real-value decision variable w^n. In addition, our objective function is non-convex owing to the UAV trajectory. Hence, our problem is MINLP which is known to be difficult to solve. Furthermore, there are two unknown parameters; $C_{i,j}$ and L^n_i. These two parameters belong to the vehicles and in such context, it is impractical to assume the upcoming vehicles and their demands are given beforehand. Despite that there are some heuristic methods to solve such types of problems, they are still inefficient as they cannot considers all the possible scenarios [30].

Therefore, we suggest to use DRL in order to learn the environment aspects and solve our presented problem. The complete implementation of our solution approach is explained in the next section.

5. Solution approach

To solve the given problem, we design DTDRL (based on DRL) besides cooperative and full-service content delivery algorithms. First, we formulate the Markov Decision Process (MDP) which is represented by a tuple $(\boldsymbol{S}, \boldsymbol{A}, \gamma, \boldsymbol{P}, \boldsymbol{R})$ where:

- $\boldsymbol{S}$ is a set of states, also known as state space, that includes all the possible states $s^n \in \boldsymbol{S}$ at any time slot n.
- $\boldsymbol{A}$ is a set of possible actions, also known as action space, that agent can take at each time slot n which is denoted by a^n.
- γ is the discount factor satisfying $0 \le \gamma < 1$ and it specifies how much the decision maker cares about rewards in the distant future relative to those in the immediate future.
- $\boldsymbol{P}$ is the transition probability of being in next state given the current state and current action $Pr(s^{n+1} \mid s^n, a^n), \forall s^n, s^n \in \boldsymbol{S}, a^n \in \boldsymbol{A}$.
- $\boldsymbol{R} : \boldsymbol{S} \times \boldsymbol{A} \to \mathbb{R}$ is a reward function where $r^n = r(s^n, a^n, s^{n+1})$ denotes the single-step reward of the system for transitioning from state s^n to state s^{n+1} due to action a^n.

Given the above mentioned MDP, we explain the action, state, and reward as follows:

- State $\boldsymbol{S}$: The state at time slot n, $s^n \in \boldsymbol{S}$, is defined as:

$$s^n = [w^n, L^n, R^n, C^n, D^n, U^n, K^n], \tag{19}$$

  where w^n denotes the current position of the UAV. L^n is a vector that contains the position of each vehicle at time slot n. R^n is a vector that contains the content requested by each vehicle at time slot n. D^n is a vector indicates how much each vehicle has downloaded at time slot n. U^n is a vector indicates how much each vehicle has uploaded to the UAV at time slot n. K^n is a vector that contains the indices of cached contents on the UAV at time slot n.
- Action $\boldsymbol{A}$: The action taken at time slot n, $a^n \in \boldsymbol{A}$, is composed of two sub-actions; a_1^n and a_2^n. The first sub-action, $a_1^n \in \{0, 1\}$, is for the RSU and it specifies whether to continue or cut the service for the vehicle chosen by Algorithm 2. We design Algorithm 2 to reduce the action space where this algorithm selects a vehicle to serve each time the RSU has no vehicle scheduled to serve.[5] However, it is still the agent responsibility to decide whether to continue or stop the service for the scheduled vehicle through a_1^n. The second sub-action, a_2^n, is an integer number to select one velocity out of a list of different velocities for the UAV. Hence, the action space is of size $2Y$ where Y denotes the total discrete UAV velocities including hovering.
- Reward: The immediate reward, r^n, is the sum of positive rewards due to serving vehicles sufficiently. Here, the decision maker/agent selects vehicle i which lies in the coverage region of the RSU to be served. Vehicle i must still be in need for content at time slot n. Similarly, the solution approach also chooses vehicle within UAV coverage to transmit or receive (if the UAV has the required content cached). For the sake of simplicity, we assume that each vehicle can only download one content. Thus, the immediate reward can be written as:

$$r^n = \begin{cases} \sum_{j=0}^{J} C_{i,j} Z_j & \begin{array}{l}\sum_{i=0}^{I}\sum_{j=0}^{J} C_{i,j} Z_j \le \sum_{n'}^{n}(\sum_{j=0}^{J} \\ (U_{R\to i,j}^{n'} C_{i,j}) + U_{U\to i}^{n'}),\end{array} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Intuitively, we may design random or greedy policy. For example, the agent probably selects vehicles to serve at random or based on the strength of the communication channel between the RSU and vehicles to be served. However, such policies may lack substantial benefits owing to the complexity of the proposed system. Specifically, concerning the cooperation between the RSU and UAV is overlooked in such basic policies. Thereby, deep reinforcement learning can better explore and build knowledge about the environment and then exploit based on the enhanced policy it has developed throughout the learning phase.

As illustrated in Algorithm 1, DTDRL interacts with the environment to collect the samples through several iterations and reveal the actual rewards. First, DTDRL initializes random sampling policy and value function for the neural networks as in line 3. Then, in line 4, the agent starts to interact with the environment through several iterations. For each iteration, the agent first observes the environment, then selects the action based on the current policy $\pi_{\theta_{old}}$. In order to achieve dual task, the agent splits the action a^n into two sub actions, a_1^n and a_2^n, using division with floor operations as in lines 8 and 9. By dividing the action a^n to the Y, the agent can work out the first action which corresponds to the RSU wireless scheduling. Next, the agent take the division reminder of a^n to Y which gives the second action which corresponds to the UAV velocity. The RSU decides whether to serve the selected vehicle ζ or not based on a_1^n. The vehicle ζ is chosen by Algorithm 2. This algorithm selects a vehicle which recently entered the area and thus has the longest time to remain in contact with the RSU. If the selected vehicle is fully served or the agent decides to cut service from it, by setting $a_1^n = 1$, another vehicle is selected based on the same policy. When the selected vehicle is fully served, the agent is rewarded based on that service. Likewise, the UAV adjusts its position based on a_2^n and then schedules the wireless resources based on Algorithm 3 and 4. If the selected vehicle to be served through the UAV has received full service, this is also considered as step reward. After gathering the set of samples and computing the rewards, DTDRL finds out the advantage function for all the iterations (line 27). In this paper, we use Proximal Policy Optimization (PPO) as our deep reinforcement learning technique. Thus, the agent computes the surrogate loss function and optimizes it via Adam optimizer as in line 28. In PPO, similar to Trust Region Policy Optimization (TRPO), the policy update is done only if the difference between the old policy and the new policy does not exceed certain threshold. The policy is updated if the advantage function is within reasonable value. If it is very large, it means the action becomes much more likely after the last step of gradient ascent and, hence, it might become out of its trusted region, so we should not update to avoid having worse results in the future. In addition, the same applies if the value becomes much less probable in the current step.

To understand the efficiency of DTDRL, we may need to compute its complexity. Based on [31], the complexity of connected network with P layers is $O(\sum_{p=0}^{P} n_p n_{p-1})$ where n_p denotes the total number of neurons in layer p.

Concerning the uplink, as laid out in Algorithm 3, the UAV checks each existing vehicle and selects the one which satisfies its criteria. The vehicle should hold a popular content which is not cached on the UAV before. The content popularity is measured based on most frequently used (MFU). If the cache has no space to cache a content, then it can replace contents based on least recently used, however, the fetched content has to be more popular. The complexity of Algorithm 3 is $O(\partial |J|)$ where ∂ represents the

[5] This algorithm approximates the solution and without it, the agent will need to learn for longer time. The algorithm also helps the agent to provide full content service in a prompt way rather than randomly schedule vehicles.

Algorithm 1 DTDRL.

Inputs: L, C, N, Learning Rate, γ, ϵ.
Outputs: The dual policy of RSU resource allocation and UAV velocity control.
Initial policy π with random parameter θ and threshold ϵ
for each episode $k \in \{0, 1, 2, ...\}$ **do**
  for $n : \{0, 1, 2, ..., N\}$ **do**
    Observe state $Z_j, w^n, L_i^n, C_{i,j}, U_i^n, D_i^n, \forall i \in I^n$.
    Select action a^n from $\pi_{\theta_{old}}$
    Extract first action, $a_1^n = \lfloor \frac{a^n}{Y} \rfloor$
    Extract second action, $a_2^n = a^n \% Y$
    Serve vehicle that is selected by a_1^n via the RSU.
    if $a_1^n = 1$ **then**
      Set $\zeta = None$
    Call Algorithm 2
    if the selected vehicle is fully served at n **then**
      Add the reward to r_n based on (20).
    Move the UAV according to $w^{n+1} = w^n + a_2^n$.
    if UAV is outside the road segment **then**
      Keep the UAV at the current position.
    Set $v_{i,j}^n \in \{0, 1\}$ based on Algorithm 3.
    Set $y_i^n \in \{0, 1\}$ based on Algorithm 4.
    if the selected vehicle is completely served **then**
      Calculate step reward as in Eq (20).
    if vehicle i content is fully received by the UAV **then**
      if vehicle i content needs space **then**
        Remove other content(s) as in 5-C.
      Store vehicle i content in the UAV cache.
  Compute advantage estimate $\hat{A}$ for all epochs.
  Optimize surrogate loss function using Adam optimizer.
  Update policy $\pi_{\theta_{old}} \leftarrow \pi_\theta$.

Algorithm 2 RSU to Vehicle Service.

Inputs: n, Z_j, $R_{i,j}$.
Outputs: ζ as the chosen vehicle to commit serving.
if $\zeta = None$ OR Vehicle ζ has left the RSU coverage OR fully served **then**
  Set $\zeta = None$
  for each vehicle i within RSU coverage at time slot n **do**
    if vehicle i is not completely served and has the longest remaining time to stay within the RSU coverage **then**
      Set $\zeta = i$

Algorithm 3 Popularity-Based Content Fetching (PBCF).

Inputs: n, Z_j, $C_{i,j}$.
Outputs: Π as the vehicle to fetch from at n.
Set $\Pi = None$.
Sort cache items by popularity
for $\forall i \in I$ where $a_i < n < d_i$ **do**
  if i's content does not exist in the UAV cache and the content is not fully fetched yet **then**
    Compute the total size of cached contents on the UAV.
    if the vehicle i content cannot fit in the cache **then**
      Define $R = 0$ as the amount to remove in order to cache the new content.
      for each content $j : k_j^n = 1$ **do**
        if vehicle i content has higher MFU value than Π and j **then**
          Set $R = R + Z_j$
          if vehicle i content size $\leq R$ **then**
            Set $\Pi = i$
            break
        else
          break
    else
      if Π' content is less popular than i's **then**
        Set $\Pi = i$

maximum number of vehicles present simultaneously in the road segment and $|J|$ is the number of contents in the library. The value of ∂ depends on the road density and, in reality, $\partial \ll I$.

For the downlink, the UAV is committed to serve a vehicle as long as it resides within the coverage area of the UAV and not fully served. The UAV also checks if it has already scheduled a vehicle to serve previously so it continues the service if that vehicle still satisfies the conditions above. However, if no vehicle has been selected previously or the service terminated, the UAV starts to look for another vehicle to serve. The UAV gives priority to vehicles which have already downloaded some of their requests. In such way, the UAV will enable the cooperation with the RSU in order to fulfill vehicles needs. The UAV also takes into consideration the availability of requested contents in its cache as it cannot serve vehicles that it does not hold along their requests in the first place. The complexity of Algorithm 4 is $O(\partial)$.

Algorithm 4 Committing Content Service (CCS).

Inputs: n, Z_j, $R_{i,j}$, G.
Outputs: G as the chosen vehicle to commit serving.
if $G = None$ OR Vehicle G has left the UAV coverage OR fully served **then**
  Set $G = None$
  for each vehicle i within UAV coverage at time slot n **do**
    if vehicle i requested a content that is cached on the UAV, not completely served, has more content amount downloaded previously, and the remaining time slots are enough to receive the whole content **then**
      Set $G = i$

Table 1
Simulation Parameters.

Parameter	Value
RSU and UAV coverage range	200 m
Δ	1 sec [35]
Number of Contents	20
Bandwidth	40 MHz [36]
Zipf (α)	1.5
$P_{U\to V}^{i,n}$, $P_{V\to U}^{i,n}$	10 W
UAV discrete velocities	[0, 10] m/s
Inter-arrival time of vehicles	[0.5-3] s
UAV altitude	20 m [37]
η_1, η_2 (for dense urban area)	11.95, 0.136

Finally, the cache replacement policy used in this work is as follows. Whenever the cache is full and a new content is fetched, the UAV starts freeing up space by removing contents which are less popular until enough room is made to store the recently fetched content.

6. Evaluation

6.1. Simulation setup

For the Deep reinforcement learning, 3 linear layers are used with tanh as activation function for the middle layers and softmax for the output layer. Internal layers contain 64 units each and Adam optimizer is incorporated to minimize the loss function. Learning rate is set to 0.002, γ to 0.08, and clip to 0.02. On the other hand, we use SUMO to mimic the vehicular environment with a road of 1 Km. SUMO has been used widely for traffic simulation in thousands of articles [32–34]. The simulation is run on a lab computer with Core i7 processor and 16 GB ram. Python 3 is used to implement DRL while PyTorch 1.4, a well-know machine learning library, is used to deploy deep neural networks. Additionally, the generation of content requests follows Zipf distribution. The key simulation parameters are listed in Table 1.

To the best of knowledge, there is no work in the literature that targets similar problem. Thus, we use baseline methods similar to [13] to compare with our solution approach.

- NonCoop: Non-cooperative method where the RSU and UAV aim to maximize their own gain independently. That is, the RSU greedily selects vehicles to serve such that it can fully satisfy and generate revenue. Similarly, the UAV does the same and it is assumed stationary.
- No UAV: As its name suggests, there is no UAV while the RSU alone will try to maximize the revenue. This will let us

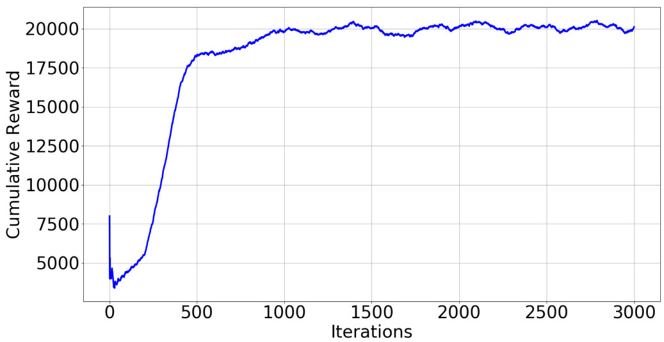

Fig. 2. DTDRL convergence.

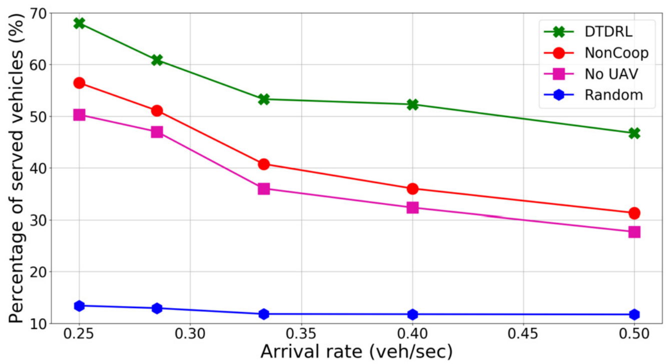

Fig. 3. Service rates of vehicles for different vehicle arrival rates.

understand how influential the UAV is on the overall system performance.

- Random: The RSU schedules resources to serve vehicles randomly.

6.2. Evaluation

We conduct five experiments to examine the various aspects of the proposed system.

6.2.1. DTDRL convergence

as DTDRL encounters more observation samples and takes actions, it can learn to perform better as demonstrated in Fig. 2. One can observe that DTDRL converges after approximately 1000 iterations.

6.2.2. Vehicle arrival rate

In order to measure the impacts of high dense and low dense road conditions, we vary the arrival rate and compute the ratio of total vehicles served. In Fig. 3, we plot the results of various arrival rates ranging from 0.25 veh/sec to 0.5 veh/sec. In this experiment, we set content size between 1.8 to 2.2 Gb and cache size to 18 Gb. As it can be observed, the percentage of vehicles served goes down as the arrival rate increases. In fact, as the density of the road increases, the requests will increase and the RSU and UAV will be less able to serve all the demand. One can also notice that our solution approach has significant advantage over the baselines. This advantage becomes more apparent as the arrival rate increases. That is, our solution approach can better adapt to a highly dense environment by enabling efficient cooperation between the UAV and the RSU. Besides, the mobility of UAV in DTDRL has a great impact on the system performance.

At arrival rate of 0.25 veh/sec, our proposed DTDRL attains 10% more vehicles served than the second highest which is NonCoop. This performance gap grows up to reach above 15% at arrival rate of 0.5 veh/sec. On the other hand, there is a considerable difference between NonCoop and No UAV. This difference is due to the existence of the UAV; i.e., the UAV can increase the performance by around 15%. In the meantime, Random seems to provide poor solutions as it cannot serve most of the vehicles due to the interrupted service provided.

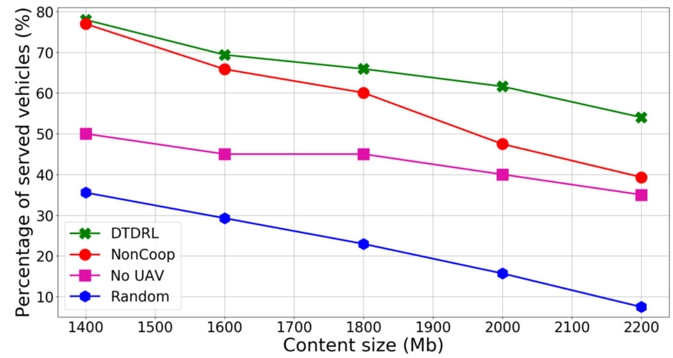

Fig. 4. Service rates of vehicles for various content sizes.

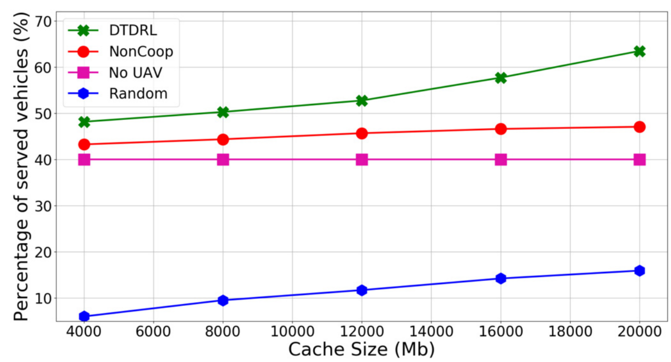

Fig. 5. Service rates of vehicles for different cache capacities.

6.2.3. Content sizes

One of the key factors that impacts the performance of the system is the size of contents. When the content sizes are small, the RSU and UAV can readily serve them completely to the requesters. The cooperation becomes more necessary as the content size increases where the RSU needs assistance from the UAV to pursue the service until completion. In light of above, we conduct this study where we vary the content size from 1.4 to 2.2 Gb. Note that arrival rate of vehicles is set to 0.33 veh/sec and cache capacity can store at most 10 contents.

As shown in Fig. 4, with small contents, DTDRL and NonCoop almost achieve similar performance since the cooperation and UAV mobility is less important. However, the superiority of DTDRL appears as the content size increases. We can see that at content size of 2.2 Gb, the difference becomes around 15%. Meanwhile, No UAV obtains less gain by around 50% to 40% vehicles served. Random, as usual, comes in the last place and the performance sharply decreases with content size.

6.2.4. Cache capacity

The cache unit capacity plays key role in the proposed system as it impacts the capability of the UAV to contribute in serving vehicles. A UAV with small cache unit can store only few contents and consequently serve less number of vehicles. While a UAV with larger cache unit would more effectively cooperate with the RSU in delivering contents. In this study, we vary the cache unit size in the range of 4 to 20 Gb while content size is set to 2 Gb and arrival rate of vehicles is 0.33 veh/sec. As illustrated in Fig. 5, larger cache

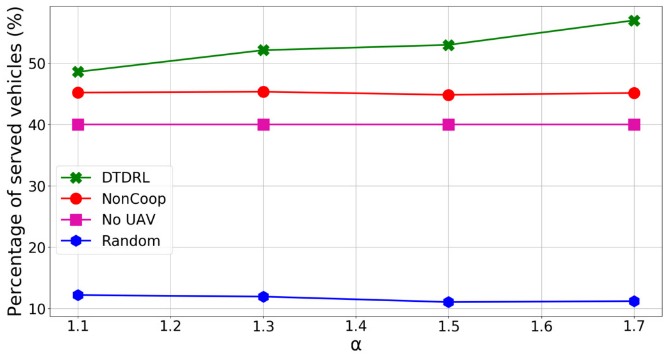

Fig. 6. Service rates of vehicles for Zipf skewness values.

unit results in higher overall performance. At 20 Gb, it attains 15% gain higher than NonCoop. Meanwhile, No UAV is not impacted by the cache capacity changes since it does not include a UAV in the first place. One can also notice that even Random experiences better performance with bigger cache units.

6.2.5. *Content popularity distribution*

the popularity distribution characteristics is, in general, very critical in cache based systems as it determines the average number of hits each content will receive and the shape of hits distribution. Thus, in this study we are going to vary the skewness parameter, α, of Zipf and observe the behaviors of the system. Hence, we change α between 1.1 to 1.7. At 1.1 the hits will be more fairly distributed among contents while in 1.7 the hits will be concentrated on the top popular contents.

As seen in Fig. 6, when α is as small as 1.1, the service provided is relatively limited. This is due to that the UAV can less contribute in serving vehicles as more requests are not cached. In this experiment we set the content size to 2 Gb and cache unit to 10 Gb while arrival rate is 0.33 veh/sec. However, as α increases, one can observe that more vehicles are being served by DTDRL. As α goes higher, popular contents will receive higher requests as well. Thus, if those contents are cached, the UAV will be able to serve them. We can notice that our proposed method makes most benefit of this feature while the other methods achieve very limited gain. That is, our method can better take advantage of the cooperation and the existence of the UAV and offload more traffic volume from the RSU.

6.2.6. *UAV mobility*

needless to say, UAV trajectory has profound influences on the overall performance. Thus, in order to study the performance of our solution approach in terms of steering the UAV, we conduct a study where we compare with three different well-known counterparts.

- SUAV: Stationary UAV similar to NonCoop.
- RW: Random walking UAV.
- CW: Constant walking UAV where the UAV moves at the maximum speed. Once the UAV reaches either end of its designated coverage segment, it reverses the direction.

In this experiment, we set cache size to 10 Gb, content size to 2 Gb, and vehicle arrival rate to 0.3 veh/sec. Additionally, we perform two tests, one with content updates through vehicles and the second one without updates. In the latter, we assume there are random contents stored in the UAV cache and will not be updated. Through this, we can evaluate the impacts of the proposed fetching methods.

In Fig. 7, we present the contributions of the UAV in serving vehicles. This is computed as the total gain achieved through the UAV over the total gain. As demonstrated in the bar chart of Fig. 7, our proposed method generates much higher gain from the UAV than using a UAV with static cache. In general, it is very noticeable that DTDRL contributes higher than the other methods at 35% of total vehicle service and about 42% better than SUAV. SUAV, in its turn, comes in the second place with slightly better performance than RW. Meaning, that a stationary UAV can better establish stable connection with vehicles than other methods that do not take smart decisions. We can also observe that RW has slightly better performance than CW due to that we use uniform random walking where the UAV circulates around the center of its coverage. That will allow the UAV to establish opportunistically longer contact times with some vehicles. In contrary CW keeps the UAV moving far from the center of the road and thus the connection will experience several interruption events.

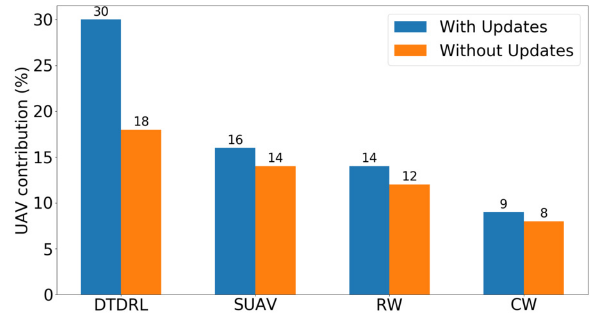

Fig. 7. Comparison of UAV shares in serving vehicles for different trajectory techniques.

Moreover, one can also observe that if the UAV does not update its cache through fetching contents, there will be significant loss. Our DTDRL demonstrates the highest utility of the fetching with more than 40% increase in the UAV contribution when fetching contents from the passing vehicles. Meanwhile, the other UAV mobility methods show only limited enhancement with content updates.

7. Conclusion

In this paper, we presented a cooperative framework where a RSU and a UAV, equipped with a cache capability, cooperate to support content delivery service in vehicular networks. DTDRL algorithm is exploited to learn the vehicular environment and its dynamics in order to control the scheduling and caching mechanism. Simulation results show that the cooperation between the two infrastructures is very lucrative in terms of number of requests satisfied. The numerical results also demonstrate that the mobility of the UAV plays a key role in the overall performance. In addition, this article presents a new technique to populate contents on a flying UAV cache via passing vehicles and show how useful it is to leverage such technique. Finally, we demonstrate that the DTDRL approach achieves the highest performance compared to the other counterpart methods. In terms of future work, other metrics can be investigated. For example, Mean Opinion Score (MOS) can be considered in order to improve users' QoE. A multi-UAV scenarios can also be considered where several UAVs collaborate with the RSU to provide enhanced quality of service for the vehicles. Moreover, the willingness of the vehicles to cooperate and the related privacy issues with the RL agent can also be studied.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Abu Talib, S. Abbas, Q. Nasir, M.F. Mowakeh, Systematic literature review on Internet-of-vehicles communication security, Int. J. Distrib. Sens. Netw. 14 (12) (2018) 1550147718815054.

[2] A. Al-Hilo, D. Ebrahimi, S. Sharafeddine, C. Assi, Revenue-driven video delivery in vehicular networks with optimal resource scheduling, Veh. Commun. 23 (2020) 100215.

[3] M. Pätzold, 5 g unlocks its power for global mobile connectivity [mobile radio], IEEE Veh. Technol. Mag. 15 (2) (2020) 5–11.

[4] A. Al-Hilo, et al., Cooperative content delivery in UAV-RSU assisted vehicular networks, in: Proc. 2nd ACM MobiCom Workshop, DroneCom, 2020, pp. 73–78.

[5] O.A. Wahab, H. Otrok, A. Mourad, Vanet qos-olsr: Qos-based clustering protocol for vehicular ad hoc networks, Comput. Commun. 36 (13) (2013) 1422–1435.

[6] Z. Ning, K. Zhang, X. Wang, M.S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, R.Y. Kwok, Joint computing and caching in 5 g-envisioned Internet of vehicles: a deep reinforcement learning-based traffic control system, IEEE Trans. Intell. Transp. Syst. (2020) 1–12.

[7] H. Kim, L. Mokdad, J. Ben-Othman, Designing UAV surveillance frameworks for smart city and extensive ocean with differential perspectives, IEEE Commun. Mag. 56 (4) (2018) 98–104.

[8] O.S. Oubbati, M. Atiquzzaman, T.A. Ahanger, A. Ibrahim, Softwarization of UAV networks: a survey of applications and future trends, IEEE Access 8 (2020) 98073–98125.

[9] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, UAV-assisted content delivery in intelligent transportation systems-joint trajectory planning and cache management, IEEE Trans. Intell. Transp. Syst. (2020) 1–13.

[10] M. Najafi, et al., Statistical modeling of the FSO fronthaul channel for UAV-based communications, arXiv preprint, arXiv:1905.12424, 2019.

[11] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, M.J. Kochenderfer, Combining planning and deep reinforcement learning in tactical decision making for autonomous driving, IEEE Trans. Intell. Veh. 5 (2) (2019) 294–305.

[12] M.C. Lucic, H. Ghazzai, Y. Massoud, A generalized dynamic planning framework for green UAV-assisted intelligent transportation system infrastructure, IEEE Syst. J. 14 (4) (2020) 4786–4797.

[13] J. Chen, H. Wu, P. Yang, F. Lyu, X. Shen, Cooperative edge caching with location-based and popular contents for vehicular networks, IEEE Trans. Veh. Technol. 69 (9) (2020) 10291–10305.

[14] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, N. Ansari, Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks, IEEE Int. Things J. 7 (1) (2019) 247–257.

[15] R. Zhang, et al., UAV-aided data dissemination protocol with dynamic trajectory scheduling in vanets, in: Proc. IEEE Int. Conf. Commun., ICC, 2019, pp. 1–6.

[16] R. Lu, et al., UAV-assisted data dissemination with proactive caching and file sharing in V2X networks, in: Proc. IEEE GLOBECOM, 2019, pp. 1–6.

[17] F. Zeng, et al., UAV-assisted data dissemination scheduling in vanets, in: 2018 IEEE International Conference on Communications, ICC, IEEE, 2018, pp. 1–6.

[18] W. Shi, et al., Drone assisted vehicular networks: architecture, challenges and opportunities, IEEE Netw. 32 (3) (2018) 130–137.

[19] Y. Hui, Z. Su, T.H. Luan, Collaborative content delivery in software-defined heterogeneous vehicular networks, IEEE/ACM Trans. Netw. 28 (2) (2020) 575–587.

[20] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, X. Shen, Air-ground integrated vehicular network slicing with content pushing and caching, IEEE J. Sel. Areas Commun. 36 (9) (2018) 2114–2127.

[21] C.M. Silva, F.A. Silva, J.F. Sarubbi, T.R. Oliveira, W. Meira Jr, J.M.S. Nogueira, Designing mobile content delivery networks for the Internet of vehicles, Veh. Commun. 8 (2017) 45–55.

[22] F. Malandrino, C.-F. Chiasserini, S. Kirkpatrick, The impact of vehicular traffic demand on 5 g caching architectures: a data-driven study, Veh. Commun. 8 (2017) 13–20.

[23] H. Abualola, H. Otrok, H. Barada, M. Al-Qutayri, Y. Al-Hammadi, Matching game theoretical model for stable relay selection in a UAV-assisted Internet of vehicles, Veh. Commun. 27 (2021) 100290.

[24] O.S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, A. Rachedi, Uav-assisted supporting services connectivity in urban vanets, IEEE Trans. Veh. Technol. 68 (4) (2019) 3944–3951.

[25] M. Chen, et al., Caching in the sky: proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience, IEEE J. Sel. Areas Commun. 35 (5) (2017) 1046–1061.

[26] P. Wang, B. Di, H. Zhang, K. Bian, L. Song, Cellular v2x communications in unlicensed spectrum: harmonious coexistence with vanet in 5 g systems, IEEE Trans. Wirel. Commun. 17 (8) (2018) 5212–5224.

[27] W.S. Atoui, W. Ajib, M. Boukadoum, Offline and online scheduling algorithms for energy harvesting rsus in vanets, IEEE Trans. Veh. Technol. 67 (7) (2018) 6370–6382.

[28] M. Mozaffari, W. Saad, M. Bennis, M. Debbah, Unmanned aerial vehicle with underlaid device-to-device communications: performance and tradeoffs, IEEE Trans. Wirel. Commun. 15 (6) (2016) 3949–3963.

[29] L. Yang, W. Zhang, Beam tracking and optimization for UAV communications, IEEE Trans. Wirel. Commun. 18 (11) (2019) 5367–5379.

[30] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, A. Ghrayeb, Age of information aware trajectory planning of UAVs in intelligent transportation systems: a deep learning approach, IEEE Trans. Veh. Technol. 69 (11) (2020) 12382–12395.

[31] M.S. Shokry, D. Ebrahimi, C. Assi, S. Sharafeddine, A. Ghrayeb, Leveraging UAVs for coverage in cell-free vehicular networks: a deep reinforcement learning approach, IEEE Trans. Mob. Comput. (2020).

[32] M.J. Khabbaz, W.F. Fawaz, C.M. Assi, A simple free-flow traffic model for vehicular intermittently connected networks, IEEE Trans. Intell. Transp. Syst. 13 (3) (2012) 1312–1326.

[33] W.M. Griggs, R.H. Ordóñez-Hurtado, E. Crisostomi, F. Häusler, K. Massow, R.N. Shorten, A large-scale sumo-based emulation platform, IEEE Trans. Intell. Transp. Syst. 16 (6) (2015) 3050–3059.

[34] L. Bedogni, M. Gramaglia, A. Vesco, M. Fiore, J. Härri, F. Ferrero, The bologna ringway dataset: improving road network conversion in sumo and validating urban mobility via navigation services, IEEE Trans. Veh. Technol. 64 (12) (2015) 5464–5476.

[35] M. Samir, S. Sharafeddine, C.M. Assi, T.M. Nguyen, A. Ghrayeb, Uav trajectory planning for data collection from time-constrained iot devices, IEEE Trans. Wirel. Commun. 19 (1) (2019) 34–46.

[36] Y.K. Tun, A. Ndikumana, S.R. Pandey, Z. Han, C.S. Hong, Joint radio resource allocation and content caching in heterogeneous virtualized wireless networks, IEEE Access 8 (2020) 36764–36775.

[37] Z. Qiu, X. Chu, C. Calvo-Ramirez, C. Briso, X. Yin, Low altitude UAV air-to-ground channel measurement and modeling in semiurban environments, Wirel. Commun. Mob. Comput. 2017 (2017).