Low-latency Caching with Auction Game in Vehicular Edge Computing

Siming Wang, Zehang Zhang, Rong Yu
Guangdong University of Technology, China
Email: simingwang30@163.com,
prestonzzh@163.com, yurong@ieee.org

Yan Zhang
University of Oslo
Oslo, Norway
Email: yanzhang@ieee.org

***Abstract*—With the rapid advances in vehicular technologies and the ever-increasing demands on mobile multimedia services, vehicular networks play a crucial role in intelligent transport systems by providing resilient connection among vehicles and users. Meanwhile, a huge number of parked vehicles may have abundant and underutilized resources in forms of computation, communication and storage. In this paper, we propose a vehicular edge computing (VEC) caching scheme in which content providers (CPs) collaboratively cache popular contents in the storage of parked vehicles located in multiple parking lots. The proposed VEC caching scheme extends the data center capability from the core to the edge of the networks. As a result, the duplicate transmissions from remote servers can be removed and the total transmission latency can be significantly reduced. In order to minimize the average latency to mobile users, we present a content placement algorithm based on an iterative ascending price auction. Numerical results show that the proposed caching scheme achieves a performance gain up to 24% in terms of average latency, compared to the widely-used scheme with most popularity caching.**

***Index Terms*—Vehicular networks, multimedia services, vehicular edge computing, parking lot, edge caching.**

I. INTRODUCTION

The ever-increasing demands for rich multimedia services over vehicular networks has draw significant attention from both industry and academia [1]. However, due to the limited wireless link capacity as well as the explosive growth in mobile traffic, how to deliver the large-sized multimedia contents efficiently becomes an important yet well-addressed issue to improve users' Quality of Experience (QoE).

According to the statistics, there have been 97 million vehicles manufactured worldwide in early 2017, and most of them will be wirelessly connected by 2020 [2]. Most vehicles spend many hours per day in parking garages, parking lots, or driveways [3]. With the tremendous number of parked vehicles in urban cities, the abundance of such on-board resources (e.g. computational, storage and communication resources) are readily available [4]. Due to the characteristics of specific locations and long-time staying, parked vehicles have great potential to share their idle resources during parking [5]. For instance, vehicular fog computing has been proposed to utilize parked vehicles as infrastructure for computation and communication [6]. The existing work [5], [7], [8] exploit parked vehicles to join the urban VANETs and serve as static nodes for content delivery. To accommodate the ever-increasing demands of types of multimedia services, utilizing parked vehicles located in parking lot as infrastructure for edge caching is indeed a promising solution. Caching at the edge brings popular contents physically close to users, which makes it ideal for delay-sensitive multimedia services. By leveraging parked vehicles for edge caching, the transmission latency cost by retrieving contents from remote server can be significantly reduced and users' QoE can be enhanced eventually. In addition, the heavy pressure of transmitting large data volume on the mobile networks can be alleviated.

Existing work have been studied to utilize parked vehicles as infrastructure for storage. The authors in [3] envisioned a vehicular cloud involving cars in the long-term parking lot of a typical international airport. In [9], the authors proposed a two-tier data center architecture leveraging the idle storage resources of the parking lot. The auxiliary vehicular data center formed by parked vehicles alleviates the pressure on the conventional data center and reduces the total communication cost. A novel framework to deliver content over vehicular social networks with parked vehicles based on Device-to-Device (D2D) communications was presented in [10], the storage capacity of vehicular social networks can be increased by using the contents in the parked vehicles.

Although leveraging the parked vehicles located in parking lot as a data center has been discussed recently, few existing studies focus on how to cache popular contents in the storage of parked vehicles efficiently in multiple parking lots. In this paper, we propose a vehicular edge computing caching scheme and concentrate on the scenario, where CPs cache popular contents into the storage of parked vehicles located in multiple parking lots collaboratively. It is essential for CPs to guarantee the QoE of mobile users, such as low latency requirement. Therefore, we formulate the objective function as how to minimize the average latency of mobile users. To solve the caching problem, we design a content placement algorithm based on an iterative ascending price auction game. Numerical results validate the effectiveness of our solution.

The major contributions of this paper are summarized as follow.

- To fully utilize the idle resources of parked vehicles, we propose a low-latency VEC caching scheme that leverages idle storage resources of parked vehicles as infrastructure to improve QoE at network edge.

978-1-5386-4502-4/17/$31.00 ©2017 IEEE

- We concentrate on the parking lots caching scenario, where CPs have to pay for the storage resources owned by the parked vehicles located in multiple parking lots.
- In order to minimize the average latency of users, we present an efficient content placement algorithm based on auction game with iterative ascending price.

The rest of this paper is organized as follows. Section II introduces the system model of VEC caching scheme. We formulate the caching problem as how to minimize the average latency of mobile users in Section III. In Section IV, the content placement algorithm based on an iterative auction game is provided. Performance evaluation will be shown in Section V. We conclude our paper in Section VI.

II. SYSTEM MODEL

As shown in Fig. 1, VEC caching system consists of a central cloud and multiple parking lots. The VEC caching system is naturally an extension of the central cloud from the remote data centers to the edge of the networks. The stochastic analysis on storage capacity of the caching system is provided.

A. The Popularity of Contents

We consider there are I parking lots. We model the popularity distribution of the contents. Let us denote by $\mathcal{C} = \{c_1, c_2....c_N\}$ the content set consisting of N contents. The n^{th} content is denoted by c_n. Without loss of generality, we consider that all contents have the same size, denoted by s. The popularity distribution of the contents is represented by $\mathcal{E} = \{e_1, e_2, ..., e_N\}$. The mobile users request for c_n with the probability of e_n. The popularity distribution of the contents are modeled by the Zipf distribution [11] as

$$e_n = \frac{1/n^{\tau}}{\sum_{a=1}^{N} 1/a^{\tau}}, \tag{1}$$

where the exponent $0 < \tau \leq 1$ is a constant which characterizes the popularity of the contents.

The contents are divided into H content groups (CG), with each CG containing content with size $G = N/H$. The n^{th} content, $\forall n \in \{(h-1)G+1, ..., hG\}$, is contained in the h^{th} CG. The set of CG is denoted by $\mathcal{F} = \{f_1, f_2, ..., f_H\}$, where $h = 1, 2, ..., H$. We denote by f_h the h^{th} CG, and by p_h the probability that the mobile users request a content in f_h, and we have

$$p_h = \sum_{n=(h-1)G+1}^{hG} e_n, \quad \forall h. \tag{2}$$

B. Parking Lot Model

1) Benefits for Cooperation: As we know, the parked vehicles in the parking lot are not owned by the SPs. Thus, the on-aborad resources of the parked vehicles are transient. Therefore, some proper incentive scheme should be applied to motivate the parked vehicles to join the VEC caching system. Effective incentive schemes for parked vehicles have been proposed. For instance, parking lot can motivate the parked vehicles to form a resource pool by giving these vehicles a discount of parking fee when they leaves [7]. In addition, to build the connection between the parked vehicles and the parking lot, wired Ethernet connection or free wireless connection (e.g. WiFi) can be provided to the parked vehicles located in the parking lot [3], [7]. Moreover, the parked vehicles should be connected to a power supply. Without any power supply, the parked vehicles are not willing to let their on-board resources be available all the time waiting for requests from the SPs. Therefore, due to the benefits such as the discount of parking fee and the free wireless or wired Ethernet connection as well as the power supply, we assume that the parked vehicles located in the parking lot may be willing to share their idle storage resources to the parking lot.

2) Analysis on the Resources of Caching System: To calculate the amount of storage resources of the parking lot, it is important to characterize the parking occupancy of the parking lot. In [3], [9], the authors proposed that the parking occupancy of the parking lot can be modeled as a stochastic process by Markov chain. The number of the parked vehicles located in the parking lot could be predicted. Here, we follow such a model. For the i^{th} parking lot, the arrival rates of the parked vehicles follows Poisson distribution with the expected value λ_i. The parked vehicles stay for a duration exponentially distributed with a expected value μ_i^{-1} [9]. In this paper, we neglect the mobility of the parked vehicles, we consider that the parked vehicles are static resources provider. The maximal number of the vehicles that the i^{th} parking lot can support is denoted by N_i^p . We define $\eta_i = \frac{\lambda_i}{\mu_i}$. Then, the probability that there are m parked vehicles in the i^{th} parking lot is written as [9]

$$w_{i,m} = \frac{{\eta_i}^m}{m!} \cdot w_{i,0} = \frac{{\eta_i}^m}{m!} \cdot \frac{1}{\sum\limits_{q=0}^{N_i^p} ({\eta_i}^q/q!)}, \tag{3}$$

where $m = 1, 2, ..., N_i^p$. Based on $w_{i,m}$, the expected number of parked vehicles in the i^{th} parking lot can be calculated as follows

$$m_i = \sum_{q=0}^{N_i^p} q \cdot w_{i,q}, \tag{4}$$

We consider that the average storage resources offered by each parked vehicle is denoted as r. The total storage resources of the i^{th} parking lot can be estimated by

$$R_i = m_i \cdot r. \tag{5}$$

III. PROBLEM FORMULATION

In the VEC caching scheme, if a mobile user closed to the i^{th} parking lot requests a content that has been cached, the content will be directly sent to the target user without duplicate transmissions from the central cloud. If the requested content can not be found in the i^{th} parking lot, it has to retrieve the content from the other parking lot, which leads to additional latency. If the requested content can not be found in any parking lots, the mobile user will download the content

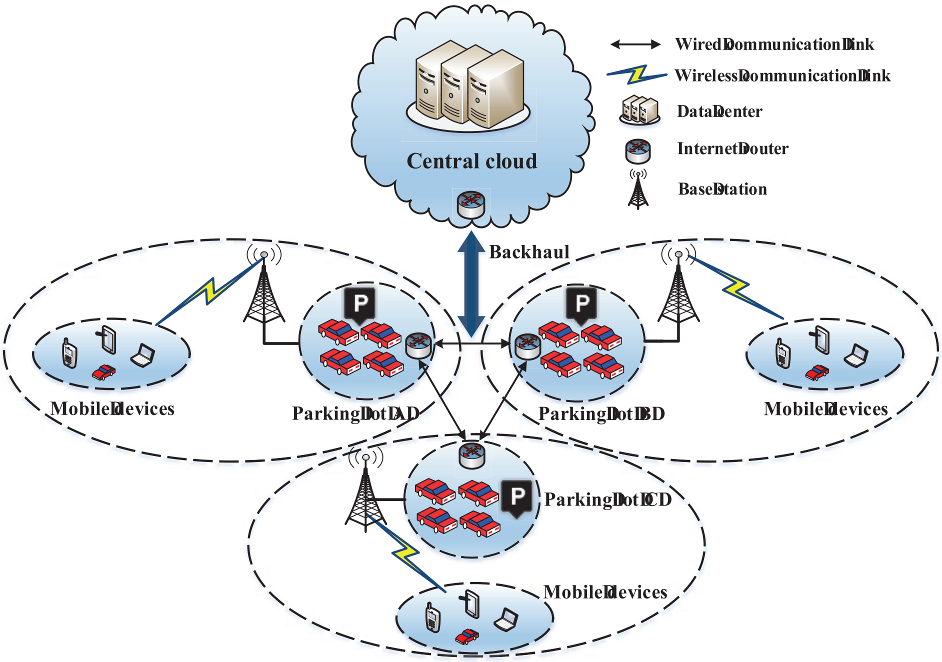

Fig. 1: The architecture of VEC caching system

from the central cloud with high latency. In this paper, we focus on the content access delay model considered in [12], [13].

We set up a content placement matrix X, the 0-1 decision variable $x_{i,n}$ indicates whether the content c_n is cached in i^{th} parking lot or not. The entry of the matrix is defined as

$$x_{i,n} = \begin{cases} 1, & \text{if } c_n \text{ is cached in the } i^{th} \text{ parking lot,} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The 0-1 variable $x_{i,n}^{j}$ and $x_{i,n}^{0}$ are defined as follow [14]

$$x_{i,n}^{j} = \begin{cases} 1, & \text{if request for } c_n \text{ from the } i^{th} \text{ parking lot} \\ & \text{is retrieved from the } j^{th} \text{ parking lot,} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

$$x_{i,n}^{0} = \begin{cases} 1, & \text{if request for } c_n \text{ from the } i^{th} \text{ parking lot} \\ & \text{is retrieved from the central cloud,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

If a mobile user requests c_n from the i^{th} parking lot, the content access delay of the mobile user can be calculated as

$$T_{i,n} = x_{i,n} \cdot t_i^p + \sum_{j \neq i} x_{i,n}^{j} \cdot \left(t_i^p + \min_{\substack{\forall j: \\ j \neq i,\ x_{i,n}^{j}=1}} \{t_{i,j}\} \right) + x_{i,n}^{0} \cdot (t_i^p + t_i^c), \tag{9}$$

where t_i^p is the transmission delay from the i^{th} parking lot to the mobile user. Here $t_{i,j}$ is defined as the transmission delay between the i^{th} parking lot and the j^{th} parking lot, and t_i^c is defined as the transmission delay from the central cloud to the i^{th} parking lot.

The mobile users' demands associated with the parking lots are denoted by $\mathcal{K} = \{k_1, k_2, ..., k_I\}$, where k_i is the average number of users' demands associated with the i^{th} parking lot. The total number of the demands from all mobile users can be calculated as

$$K_{total} = \sum_{i=1}^{I} k_i. \tag{10}$$

The main objective of this paper is to minimize the average contents access delay by collaboratively caching the popular contents in multiple parking lots. Based on (1), (9), (10), the objective function can be defined as

$$\min_{X} \quad D = \frac{1}{K_{total}} \sum_{i=1}^{I} \sum_{n=1}^{N} T_{i,n} \cdot k_i \cdot e_n \tag{11}$$

$$s.t. \quad \sum_{i=1}^{I} \sum_{n=1}^{N} x_{i,n} \cdot s \leq R_i, \quad \forall i, \tag{12}$$

$$x_{i,n}^{j} \leq x_{i,n}, \quad i = 1, ..., I, i \neq j = 0, 1, ..., I, \forall n, \tag{13}$$

$$x_{i,n} + \sum_{j \neq i} x_{i,n}^{j} + x_{i,n}^{0} = 1, \quad i = 1, ..., I, \forall n. \tag{14}$$

where (12) represents the limit for storage capacity of each parking lot. Equ. (13) ensures that a content can be retrieved from a cache only if it has been stored in that cache. Equ. (14) ensures that each demand should only be fulfilled. This optimization problem is a NP-complete problem [15], [16].

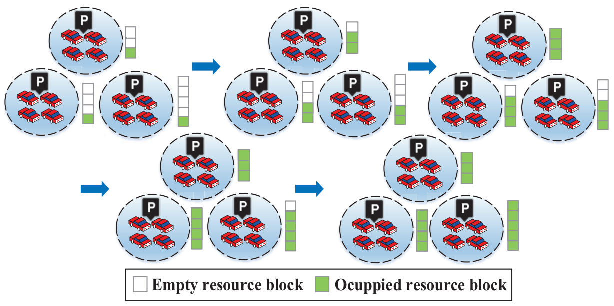

Fig. 2: The caching process of multiple parking lots.

To tackle the problem, we propose a sub-optimal solution by gradually optimizing a sequence of sub-problems.

IV. AUCTION GAME

In this section, an auction game is proposed to solve the caching problem (11)-(14). Two set of players are the CGs and the storage blocks of multiple parking lots. In this paper, we consider that the contents are divided into multiple CGs and cached in the parked vehicles. Thus, we set a common caching size G in the auction game. The storage resources of a parking lot are divided into multiple G-sized blocks. The resource blocks of the parking lots are regarded as objects denoted by $\mathcal{L} = \{l_1, l_2, ..., l_I\}$. The contents are sorted in the descending order according to the popularity distribution $\mathcal{E}$ and divided into multiple G-sized CGs. The set of CGs are regarded as bidders denoted by $\mathcal{F} = \{f_1, f_2, ..., f_H\}$.

A. Utility Function

The caching problem is deal with by multiple rounds of auction games. After each auction, the contents can be placed into the parking lots according to the auction result. Before each auction, CGs have to compute the valuation of caching their contents to each of the parking lots and bid for them. Based on equations (2) and (9), the valuation of caching the h^{th} CG into the i^{th} parking lot can be given by

$$v_h(i) = \sum_{b=1}^{I} -\Delta T_{b,h} \cdot k_b \cdot p_h, \tag{15}$$

which is the decrease of latency for the mobile users requesting contents with popularity of p_h. With the valuation function, we aim at solving the content placement problem through multi-round auction games.

The utility function $U_h(i)$ expresses the satisfaction of bidder f_h for object l_i, which is defined as the difference between the valuation function and the cost paid by the CG for the G-sized storage block

$$U_h(i) = v_h(i) - \rho G. \tag{16}$$

where ρ is the unit price for the resource block, ρG is the cost paid by the CG for the G-sized resource block. The price acts as a control factor to guide bidder to pay for the item with storage cost [17].

Algorithm 1: The Iterative Ascending Price Auction-Based Caching Algorithm

Input: $\mathcal{E}$, $\mathcal{F}$, $\mathcal{L}$
Output: $X_{I\times N}$
Begin
Stage 1. Initialization
Parking lots announce the common size of block G;
Reorder the contents in the descending order according to the popularity distribution $\mathcal{E}$;
The contents are divided into $H = N/G$ CGs.
Stage 2. Resource Blocks Auction
for $j \leftarrow 1$ *to max* $\{R_i/G\}$ **do**
  The j^{th} resource blocks of all parking lots are regarded as objects;
  Set round index $t = 0$, initial price ρ^0, and fixed price step $\Delta > 0$;
  repeat
    The CGs calculate the utility of caching the set of contents to each resource block;
    if $U_h(i, \rho^t) > 0$ **then**
      Bidder h submits bid $b_h(i) = U_h(i, \rho^t)$;
    else
      Bidder h submits 0;
    end
    The auctioneer collects bids and selects the highest one $b_{h*}^t(i^*) = \max_{f_h \in \mathcal{F}}(b_h^t)$;
    Allocate l_{i^*} to bidder h^*;
    $\mathcal{L} = \mathcal{L} - l_{i^*}$;
    Update $\rho^{t+1} = \rho^t + \Delta$;
    Set $t = t + 1$;
  until $\mathcal{L} = \emptyset$;
end
The matrix $X_{I\times N}$ shows the caching result;
End

B. Content Placement Algorithm

In this part, we apply an iterative ascending price auction based algorithm to solve the caching problem [17]. The caching scheme is determined by multi-round auction games. We set a common block size G in the round-based auction games. According to the popularity distribution of the contents, we sort the contents by the descending order of popularity. Then, the contents are divided into $H = N/G$ CGs. On the other hand, the storage resources of all parking lots are divided into multiple storage blocks with the size of G. As shown in Fig. 2, the resource blocks will be auctioned for max$\{R_i/G\}$ times, where the j^{th} resource block of all parking lots are auctioned off in the j^{th} auction [18].

In each round of auction game, the bidders calculate the utility of each object. If the utility function is non-negative, the bidder bids with their utility function, otherwise the bidder bids with 0. The bids are collected by auctioneer and the highest one is selected. The bidder who bids with the highest one will get the object, and the object is removed from $\mathcal{L}$.

TABLE I: SIMULATION PARAMETERS

Parameter	Value
Size of the contents s (MB)	30
Common size of block G (GB)	4
Average resources offered by vehicle r (GB)	1
Total number of contents N	from 10000 to 30000
Number of the users' demands $\mathcal{K}$	uniform in $[100, 200]$
Download delay from parking lot t_i^p(ms)	uniform in $[5, 15]$
Transmission delay within parking lots $t_{i,j}$(ms)	uniform in $[30, 50]$
Transmission delay from central cloud t_i^c(ms)	uniform in $[80, 200]$

The auctioneer sets $\rho^{t+1} = \rho^t + \Delta, t = t+1$, then the auction moves to the next round. The auction game repeats until $\mathcal{L} = \emptyset$. We list an overview of the algorithm in Algorithm 1.

C. Strategy-proof

Proposition 1: The content placement algorithm based on iterative ascending price auction is strategy-proof.

Proof: For the resource block l_i in round t, bidder h can bid with $b_h(i)$ or 0. Two cases for bidding strategy of bidder h are considered as follow:

1) At the beginning round of the auction game, the utility function of bidder h satisfies $U_h(i, \rho^t) > 0$, if bidder h bids with 0, it will quit this round and lose the i^{th} resource block which can minimize the content access delay.

2) In the process of the proposed algorithm, the price ρ^t will gradually increase, the utility function of bidder h satisfies $U_h(i, \rho^t) < 0$. If bidder h bids with $U_h(i, \rho^t)$ and wins the i^{th} resource block, it will obviously have a negative effect on the final utility.

In summary, we can draw the conclusion that the optimal strategy for bidder h is to bid with its true demand. If bidder h bids untruthfully, it will lead to revenue loss and fail to decrease the content access delay. Therefore, the proposed storage resource allocation algorithm is strategy-proof. ■

D. Complexity

The proposed algorithm requires $\beta = \max\{R_i\}/G$ times of iterations to allocate the resource blocks. While in the proposed algorithm, each CG calculates its utility during each iteration t, which takes $\mathcal{O}(H * I)$. The complexity of selecting the maximal bid by auctioneer is $\mathcal{O}(H * I)$ during each iteration t. If the total number of iterations of each auction game is t, the algorithm takes $\mathcal{O}(2t(H * I))$ in each auction. Therefore, the total complexity of the algorithm is $\mathcal{O}(2\beta t(H * I))$.

V. NUMERICAL RESULTS

In this section, we evaluate the average user delay performance of the proposed caching scheme, as well as the impact of system parameters. We consider a VEC caching system consisting of 3 parking lots, each deployed with WiFi infrastructure for wireless connection of parked vehicles. The size of CG is the same as the size of storage block. The detailed simulation parameters are shown in Table I.

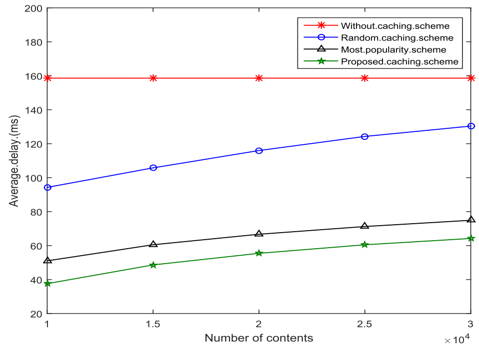

Fig. 3: Average delay versus number of contents.

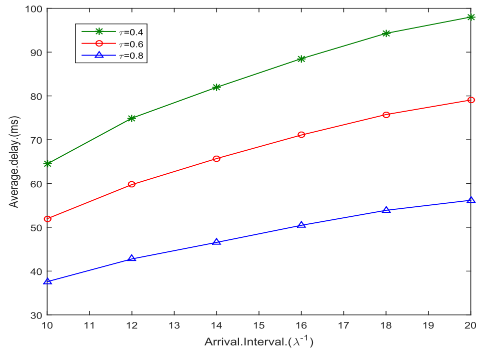

Fig. 4: Average delay versus arrival interval.

The first simulation compares the performance of schemes without caching, with random caching, with most popularity caching and the proposed caching scheme. Fig. 3 shows the results in terms of average delay versus number of contents. We set $\lambda = 10^{-1}$ and $\mu = 650^{-1}$ with the Zipf exponent parameter $\tau = 0.8$ [9]. Random caching scheme means caching the contents in each parking lots randomly. Most popularity caching scheme means caching the most popular contents in each parking lot. In Fig. 3, the performance of the average delay values, achieved by schemes with without caching, random caching, most popularity caching and the proposed caching scheme, shows as an increasing function of the contents number. The proposed caching scheme achieves a performance gain up to 24% and 118% in terms of average latency compared to the most popularity caching scheme and random caching scheme. Compared with no caching scheme, random caching scheme and most popularity caching scheme, the performance of our mechanism can greatly reduce the average delay of users. Therefore, these results indicate that the proposed content caching scheme is an efficient solution.

The second simulation investigates the influence of λ. We set $\mu = 650^{-1}$, $N = 30000$ contents with Zipf exponent parameter $\tau = 0.4, 0.6, 0.8$ respectively. The performance of average delay as a function of λ^{-1} is plotted out. The arrival rates of vehicles follows Poisson distribution with λ, and λ^{-1} is the arrival interval of the vehicles. Fig. 4 shows that when

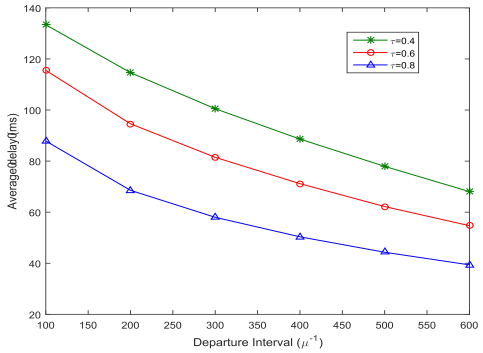

Fig. 5: Average delay versus departure interval.

λ^{-1} gets greater, the average delay increases. As λ^{-1} gets greater, the expected number of parked vehicles will decrease, which leads to resources reduction of the parking lot. Then, fewer contents can be cached in the parking lot. Therefore, more requests must be satisfied by retrieving contents from other parking lots or from remote server. It can also be observed that a greater Zipf exponent τ makes it easier to achieve low latency. Because a higher τ means that most popular contents account for the majority of download requests.

In the third simulation, the impact of μ is evaluated, where we set $\lambda = 10^{-1}$, $N = 30000$ contents with Zipf exponent parameter $\tau = 0.4, 0.6, 0.8$, respectively. The performance of average delay as a function of μ^{-1} is shown in Fig. 5. The parked vehicles stay for a duration exponentially distributed with a mean value μ^{-1}, and μ^{-1} means the departure interval. Fig. 5 shows that the performance of average delay decreases with the increase of μ^{-1}. As μ^{-1} gets greater, the expected number of parked vehicles will increase. The storage resources of the parking lot increase, and more contents can be cached in the parking lot. Therefore, more requests can be satisfied by retrieving contents locally.

VI. Conclusion

In this paper, we proposed a VEC caching system that leverages idle storage resources in multiple parking lots as the infrastructure for auxiliary edge caching. We concentrated on the parked vehicles caching scenario and formulate the caching problem as how to minimize the average latency of mobile users. To solve the caching problem, we designed a content placement algorithm based on an iterative ascending price auction game. Simulation results have shown that the proposed content caching scheme is an efficient solution. In the future work, we will focus on incentive scheme to motivate parked vehicles to share their idle resources to the parking lots.

Acknowledgment

Rong Yu is the corresponding author of this paper. This work was supported in part by programs of NSFC under Grant nos. 61422201, 61370159 and U1301255, U1501251, the Science and Technology Program of Guangdong Province under Grant no. 2015B010129001, Special-Support Project of Guangdong Province under grant no. 2014TQ01X100, High Education Excellent Young Teacher Program of Guangdong Province under grant no. YQ2013057, Science and Technology Program of Guangzhou under grant no. 2014J2200097.

References

[1] Z. Su, Y. Hui, and Q. Yang, "The next generation vehicular networks: a content-centric framework," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 60–66, 2017.

[2] The Statistics Portal http://www.statista.com/.

[3] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, and I. Khalil, "Datacenter at the airport: Reasoning about time-dependent parking lot occupancy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2067–2080, 2012.

[4] S. Abdelhamid, H. Hassanein, and G. Takahara, "Vehicle as a resource (vaar)," *IEEE Network*, vol. 29, no. 1, pp. 12–17, 2015.

[5] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "Pva in vanets: Stopped cars are not silent," in *INFOCOM, 2011 Proceedings IEEE*, pp. 431–435, IEEE, 2011.

[6] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technologies*, 2016.

[7] Z. Su, Q. Xu, Y. Hui, M. Wen, and S. Guo, "A game theoretic approach to parked vehicle assisted content delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, 2016.

[8] A. B. Reis, S. Sargento, and O. K. Tonguz, "Parked cars are excellent roadside units," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[9] L. Gu, D. Zeng, S. Guo, and B. Ye, "Leverage parking cars in a two-tier data center," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 4665–4670, IEEE, 2013.

[10] Z. Su, Y. Hui, and S. Guo, "D2d-based content delivery with parked vehicles in vehicular social networks," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 90–95, 2016.

[11] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, pp. 126–134, IEEE, 1999.

[12] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[13] X. Li, X. Wang, S. Xiao, and V. C. Leung, "Delay performance analysis of cooperative cell caching in future mobile networks," in *Communications (ICC), 2015 IEEE International Conference on*, pp. 5652–5657, IEEE, 2015.

[14] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.

[15] T. X. Tran and D. Pompili, "Octopus: A cooperative hierarchical caching strategy for cloud radio access networks," *arXiv preprint arXiv:1608.00067*, 2016.

[16] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Wireless On-demand Network Systems and Services (WONS), 2017 13th Annual Conference on*, pp. 165–172, IEEE, 2017.

[17] F. Wang, C. Xu, L. Song, Z. Han, and B. Zhang, "Energy-efficient radio resource and power allocation for device-to-device communication underlaying cellular networks," in *Wireless Communications & Signal Processing (WCSP), 2012 International Conference on*, pp. 1–6, IEEE, 2012.

[18] Z. Hu, Z. Zheng, T. Wang, L. Song, and X. Li, "Caching as a service: Small-cell caching mechanism design for service providers," *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6992–7004, 2016.