Information Sciences 506 (2020) 161–173

Contents lists available at ScienceDirect

Information Sciences

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

A collective filtering based content transmission scheme in edge of vehicles

Xiaojie Wang[a,b], Yufan Feng[b], Zhaolong Ning[a,b,*], Xiping Hu[a,c], Xiangjie Kong[b], Bin Hu[a], Yi Guo[d]

[a] *School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*
[b] *School of Software, Dalian University of Technology, Dalian 116620, China*
[c] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*
[d] *Second Clinical Medical College of Jinan University, Guangzhou 510000, China*

ARTICLE INFO

Article history:
Received 29 October 2018
Revised 22 July 2019
Accepted 25 July 2019
Available online 25 July 2019

Keywords:
Edge of vehicles
Collaborative filtering
Fog computing
Markov chains

ABSTRACT

With the emergence of the ever-increasing vehicular applications and booming Internet services, the requirements of low-latency and high efficient transmission among vehicles become urgent to meet, and their corresponding solutions need to be well investigated. To resolve the above challenges, we propose a fog computing-based content transmission scheme with collective filtering in edge of vehicles. We first provide a system model based on fog-based rode side units by considering location-awareness, content-caching and decentralized computing. Then, a content-caching strategy in RSUs is designed to minimize the downloading latency. Specifically, we model the moving vehicles with the two-dimensional Markov chains, and calculate the probabilities of file caching in RSUs to minimize the latency in file downloading. Each vehicle can also maintain a neighbor list to record the encounters with high similarities, and update it based on the historic and real-time contacts. Finally, we carry on the experiments based on the real-world taxi trajectories in Beijing and Shanghai, China. Simulation results demonstrate the effectiveness of our proposed method.

1. Introduction

Cities are not only a collection of buildings, but also communities where people are closely connected with rich social interactions and interdependences. The closenesses among citizens often make neighboring people have similar interests, such as location-based information and the advertisements of nearby supermarkets [14]. Meanwhile, the wide usages of mobile electronic devices have brought wireless communication and ubiquitous computing to our life. In order to enhance the connections among individuals, information uploading and downloading among devices require to be fulfilled efficiently [22]. This calls for new technologies and deployments of location based community communication.

Nowadays, the number of vehicles has increased dramatically, and a large quantity of data are generated dispersedly. Fog computing extends the facilities of traditional cloud computing to the network edge, largely deducing the latency caused by the long distance from terminals to the cloud server. Specifically, the services and applications of fog computing are widely

* Corresponding author at: School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China.
E-mail addresses: zhaolongning@dlut.edu.cn (Z. Ning), xp.hu@siat.ac.cn (X. Hu), xjkong@dlut.edu.cn (X. Kong), bh@lzu.edu.cn (B. Hu), xuanyi_guo@163.com (Y. Guo).

https://doi.org/10.1016/j.ins.2019.07.083
0020-0255/

decentralized with location-awareness, which provide low-latency and content-aware data dissemination [21] to alleviate the latency and bandwidth limitation problems, since the near-user devices can work in a collaborative and local way.

With the advent of advanced devices and technologies, such as vehicular and other mobile applications, the number of connected devices has grown significantly. Followed that, a great quantity of data generated by devices are immersed in our daily life [25]. Cisco Global Cloud Index estimates that around 500 ZB of data will be generated by intelligent devices by 2019 [6]. Data storage and processing in cloud-based data centers are facing great challenges, such as the high latency due to long transmission distances [9,24]. In order to realize the real-time data transmissions in vehicular networks, fog computing can process the computing tasks at network edges. In fog computing, one or more collaborative end or near-by devices can be leveraged to implement storage, communication, configuration, measurement and management network functions [28]. Hence, data transmission in edge of vehicles is an innovative data sharing mode. In another point of view, a vehicle network also can be seemed as a mobile social network [23], in which vehicle users meet each other and can establish friendships. It is entirely feasible to establish reliable transmission links for wireless transmission among vehicles [4]. Therefore, in order to meet the above requirements, we intend to select vehicles with higher similarities under the fog computing in edge of vehicles.

In this work, we propose a Fog computing-based coNtent transmission scheme with Collaborative Filtering (FNCF) in edge of vehicles. The characteristics of vehicular social behaviors and fog computing are comprehensively considered. The main contributes of this paper are summarized as follows:

- Our method fully considers the social behaviors of vehicles, and the similarities among vehicles are established based on the historic and real-time contacts, with which a user-based collaborative filtering strategy is applied to select neighboring vehicles for priority transmission.
- In edge of vehicles, vehicles and Rode Side Units (RSUs) are both equipped with fog devices with the capacities of content storage and location-awareness. In that case, the content transmission can be implemented in decentralized ways to reduce content transmission latency.
- A content-caching strategy in RSUs is designed to minimize the downloading latency. We model the moving vehicles as the two-dimensional Markov chains. Probabilities of file caching in RSUs can be obtained according to the content popularity and availability in the constructed mobility model.
- Performance evaluations are carried based on the real taxi trajectories in Beijing and Shanghai cities, and the results demonstrate that our scheme outperforms others in both date delivery and downloading latency.

The rest of this paper is organized as follows. The related researches are described in Section 2. In Section 3, we illustrate the system model together with the problem formulation. The implementation details of FNCF are specified in Section 4. In Section 5, we evaluate the presented method and analyze the obtained results. In Section 6, we conclude this paper.

2. Related work

In this section, we provide an overview of the previous researches about content transmission in vehicular networks. The existing content transmission methods in vehicular networks can be roughly divided into: random methods, contact-level mobility based methods, and social-level based methods.

Random methods, such as random walk [2] based algorithms, can be used in data relay situations without information forwarding. A neighboring node is randomly selected as the next-hop content carriers. Random walk strategies can generate moderate network traffic, but often result in large end-to-end latency, such as the epidemic routing in [16]. Specifically, when a message is flooded in network forwarding, epidemic routing may obtain the low end-to-end latency and high delivery ratio. However, random walk strategies can cause unacceptable network overhead [5].

Contact-level mobility based algorithms combine the node mobility characteristics with network structures extensively to facilitate data forwarding. More than that, Markov chains are applied to model and predict the mobility of vehicles [15,20]. In these methods, the utility functions are defined to measure node importance. If one message carrier encounters another neighboring node with a high utility, the message can be forwarded to that neighboring node. With contact-level mobility based strategies, messages can be delivered effectively, since the prior contact information has been attached on nodes.

Social-level based methods are proposed with the social relationship tightness [12,13]. By considering the reliability and latency constraints, power allocation and resource block sharing based on vehicle clusters are studied in [17]. Authors in [27] propose a hybrid approach by combining Tabu search and the artificial bee colony algorithm to solve the vehicle routing problem. The experiments on a benchmark data set verify its efficiency. A cooperative caching method in data broadcasting environments is introduced in [7] to increase the bandwidth efficiency and reduce the data access latency. In addition, a trajectory-based interaction time prediction algorithm is presented in [10] to improve the service quality of vehicular applications. In our work, we capture the vehicle mobility characteristics from both the location and contact aspects.

The scale of vehicular networks increases dramatically. A large quantity of data are transmitted to and processed in the cloud center. However, the data sources are usually remote to the cloud centers, and cloud-based services cannot guarantee the low-latency requirements in content transmissions. To overcome the issues of storage, retrieval and management mentioned above, fog computing has become a practical solution to enable a smooth convergence between cloud and end-users for content delivery and real-time data processing [19]. Therefore, fog computing is promising to improve the efficiency of data transmissions since the edge severs are distributed over surrounding areas [11]. The parked vehicles are viewed as

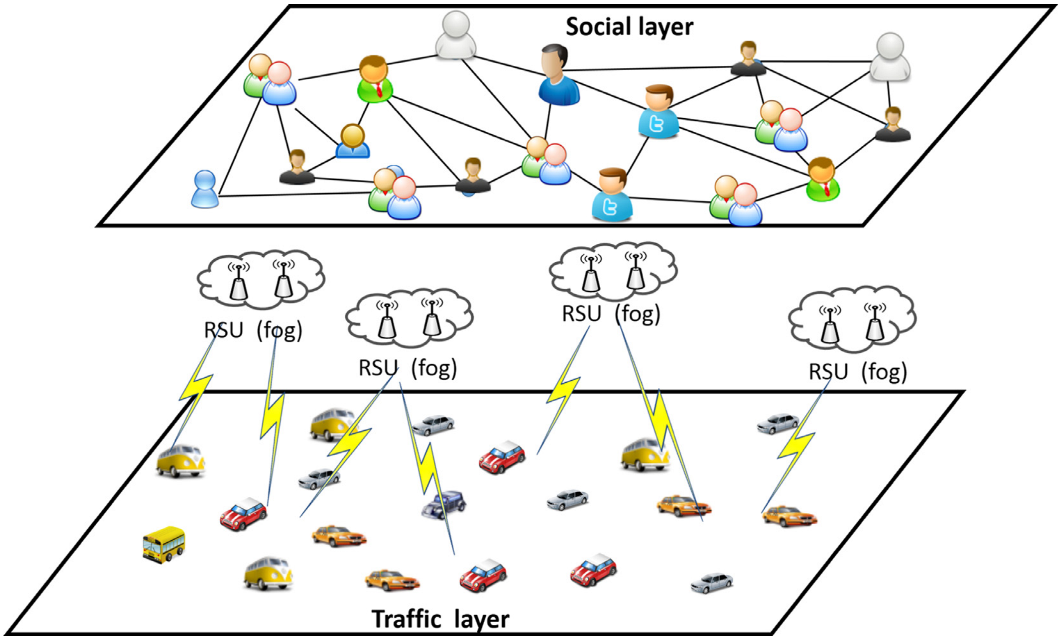


Fig. 1. The network architecture in edge of vehicles.

infrastructures in [3], which provides insights for a novel promising paradigm about vehicular fog infrastructures. According to the previous work, we take advantages of fog computing, and present a perception list of neighboring vehicles for content-precache transmission in edge of vehicles.

3. System model and problem formulation

3.1. System model

The concept of social vehicles comes from the consideration that drivers can share data with their neighbors based on common interests [18]. Social networking in vehicles integrates socially-aware networks with vehicular ad hoc networks. This allows vehicles not only to share contents, but also to select similar neighbor vehicles to improve the communication efficiency.

Fog computing can be viewed as a supplement of core cloud services by transforming traditional data center management to distributed heterogeneous platforms. Hence, fog computing can support edge of vehicles requiring latency-sensitive and context-aware processing, and Internet of Things (IoT) applications in various areas. In our framework, fog computing provides prerequisites for content caching and decision making. This is consistent with the requirements of location-awareness, content caching and high-efficient computing in content transmissions.

According to the description above, a novel social vehicular framework is illustrated in Fig. 1, which can be divided into a traffic physical layer and a mobile social layer. We specify the architecture of FNCF as follows:

- Vehicle nodes: Urban vehicles can act as the content producers, forward units or end-users. They are equipped with on-board units and limited caching storage. Hence, vehicles can not only store the information from several neighboring vehicles, but also carry and transmit contents across the urban areas through Vehicle-to-Vehicle (V2V) and Vehicle-to-RSU (V2R) patterns.
- RSUs: They are edge infrastructures deployed along the roadside, and equipped with the wireless devices to meet the requirements of V2R communications. They can make decisions and retrieve data selectively from moving vehicles. Meanwhile, mobile vehicles can also download contents from RSUs over wireless communications.

Data transmission in our framework depends on wireless technologies. A link between two vehicles is constructed only if their distances is no more than the communication range between vehicles and RSUs (d_{V2R}). The connections of V2V and V2R communications may be interrupted frequently due to the mobility of vehicles. However, the transmission contents are usually in large sizes, and can be divided into several subsegments distributed in different vehicles. With the encounter of vehicles, those segments can be transmitted to the same RSU through moving vehicles, so that they can be merged into the complete original content.

In order to maintain the updated vehicular network topology, beacon messages can be periodically broadcasted by physical links. Vehicles can learn traffic contents from their neighbors, including locations, velocities and other sensed data.

Table 1
Definitions of main notations.

Parameter	Definition
$V = \{v_0, v_1, \ldots, v_m\}$	Local set of vehicles
$F = \{f_0, f_1, \ldots, f_n\}$	Local set of transmission file
$R = \{r_{ij}\}_{m\times m}$	Local similarity matrix among vehicles
$r_i = \{r_{ij}\}$	Similarity vector of v_i, r_{ij} equals the value of similarity between v_i and v_j, $ij \in \{0, 1, \ldots, m\}$
$F_{sim}(v_i, v_j)$	Similarity between v_i and v_j
$H_{sim}(v_i, v_j)$	Historic similarity between v_i and v_j
$RT_{sim}(v_i, v_j)$	Real-time similarity between v_i and v_j
$Nei(v_i)$	Neighbor list of v_i
$D = \{d_{ij}\}_{m\times n}$	File downloading requirement matrix, $d_{ij} \in \{0, 1\}$
$A = \{a_0, a_1, \ldots, a_n\}$	File availability vector
$E = \{e_0, e_1, \ldots, e_n\}$	File popularity vector
buf_{RSU} (buf_V)	Content buffer of RSU (vehicle)
d_{V2V} (d_{V2R})	Communication range of V2V (V2R)
B_{V2V} (B_{V2R})	Normalized transmission capacity of V2V (V2R)
ϑ_{V2V} (ϑ_{V2R})	Mean downloading rate through V2V (V2R) of f_j
ϱ_{V2V} (ϱ_{V2R})	Mean transition rate of file piece downloading through V2V (V2R)

Meanwhile, social contacts among vehicles are established in edge of vehicles. In that case, we can build stable V2V links according to social similarities.

3.2. Problem formulation

Since a file can be downloaded through V2V and V2R, the objective function is to get the minimum file downloading latency. We define $V = \{v_0, v_1, \ldots, v_m, m \geq 0\}$ as a local vehicle set. Similarly, $F = \{f_0, f_1, \ldots, f_n, n \geq 0\}$ is the file set, in which each element is an entire file. The downloading requirement matrix is defined as $D = \{d_{ij}\}_{m\times n}$, where $d_{ij} \in \{0, 1\}$. If d_{ij}=1, it means that v_i requests f_j, and vice versa. We define ϱ_0 and ϱ_1 as the downloading rates of V2V and V2R communications, respectively. By jointly considering V2V and V2R to minimize the downloading latency, the problem is formulated as:

$$\begin{aligned}
&\min \sum_{v_i \in V} \sum_{f_j \in F} d_{ij}\left(\psi_0 \frac{f_j^0}{\varrho_0} + \psi_1 \frac{f_j^1}{\varrho_1}\right),\\
&s.t.:\\
&\quad C1:\ f_j^0 + f_j^1 = f_j,\ f_j > 0,\ f_j^0 \geq 0,\ f_j^0 \geq 0,\ f_j \in F,\\
&\quad C2:\ \psi_0 + \psi_1 = 1,\ \psi_0 \in [0,1]\ \psi_1 \in [0,1],\\
&\quad C3:\ \varrho_0 > 0,\ \varrho_1 > 0,\\
&\quad C4:\ d_{ij} \in \{0,1\},\ i \in \{0,1,\ldots,m\},\ j \in \{0,1,\ldots,n\}.
\end{aligned} \tag{1}$$

If v_i requests to download f_j, it can be divided into two parts, i.e., f_j^0 and f_j^1, depending on V2V and V2R communications, respectively. Variable ψ_0 and ψ_1 are the probabilities that v_i download f_j through V2V and V2R communications, respectively. These are corresponding to constraints $C1$ and $C2$ in Eq. (1). Constraint $C3$ ensures that the downloading rates of V2V and V2R are no less than 0. Constraint $C4$ illustrates the downloading requirements from vehicles. Owning to the mobility of vehicles, the file downloading procedures may be interrupted frequently. As a result, the way to download files is decided by the situation whether vehicles are in the coverage of RSUs containing the requested files. Specifically, as long as v_i is in coverage of an RSU containing f_j, v_i can download f_j through V2R communication. Otherwise, it can obtain f_j merely through V2V communication.

Consequently, in order to achieve efficient content transmission, both social layer and physical layer information are taken into consideration to optimize the transmission rates of V2V and V2R communications. In terms of V2V, we estimate the vehicular trajectories from the social networking aspect and notice that vehicles preferentially share contents with similar vehicles. Therefore, in our method, a neighbor list is equipped in each urban vehicle. Furthermore, the establishment of V2R links depends on the distances between vehicles and RSUs. Therefore, we introduce a file caching and updating strategy based on the Markov model to simulate the vehicular mobilities and states. The main notations used in this paper are explained in Table 1.

4. Fog computing-based content transmission with collective filtering strategy

This section specifies the implementation details of our designed method, including neighboring vehicle selection based on collaborative filtering strategy, content-caching mechanism based on the two-dimensional Markov model, and computational complexity analysis.

4.1. Neighboring vehicle selection based on collaborative filtering

This subsection evaluates similarities among vehicles based on collaborative filtering strategy. In the social layer, vehicles are more likely to trust and share information with the encounters with high correlations. The social layer connects physically close vehicles, and enables them to share data. The establishment of V2V links is based on the distances among vehicles.

Real-time graphical information, such as GPS and velocity, can be sensed by vehicles. A certain buffer space is also equipped with vehicles to store neighbors with higher similarities. Then, high priorities can be given to those neighbors in content transmission. Vehicles sense and record the graphical information of the surrounding vehicles in real time. The similarities among vehicles are measured based on the historic and real-time contacts.

4.1.1. Similarity evaluation

Each vehicle has a unique ID, which provides the necessary condition for updating the neighbor list from encountered vehicles. For vehicle $v_i \in V$, its neighbor list is defined as $Nei(v_i)$. The relationships among vehicles are represented by matrix $R = \{r_{ij}\}_{m\times m}, i, j \in \{0, 1, \dots, m\}$, and the diagonal elements in R are all equal to 0, i.e., the similarity of the given vehicle and itself is 0. Each row $r_i, i \in \{0, 1, \dots, m\}$ in $R = \{r_0, r_1, \dots, r_m\}^T$ stands for a similarity vector. The value of r_{ij} in r_i equals to the similarity between v_i and v_j. Neighbor list and the corresponding similarities can be updated by the mobility of vehicles.

For any two vehicles v_i, $v_j \in V$, if v_i and v_j encounter in a certain area, $F_{sim}(v_i, v_j)$ can be calculated by Eq. (2). The similarity of two vehicles consists of two parts: real-time similarity $RT_{sim}(v_i, v_j)$ and historical similarity $H_{sim}(v_i, v_j)$, which are expressed in Eqs. 3 and 4:

$$F_{sim}(v_i, v_j) = \alpha \times RT_{sim}(v_i, v_j) + (1-\alpha) \times H_{sim}(v_i, v_j). \tag{2}$$

$$RT_{sim}(v_i, v_j) = \frac{2(vel_i \times vel_j)\cdot \cos(\theta)}{vel_i^2 + vel_j^2}. \tag{3}$$

$$H_{sim}(v_i, v_j) = \frac{r_i \cdot r_j}{\sqrt{(r_i \cdot r_i) \times (r_j \cdot r_j)}}. \tag{4}$$

Herein, vel_i is the velocity of vehicle v_i, and θ is the angular direction difference between v_i and v_j. $RT_{sim}(v_i, v_j)$ depends on the current traffics. If two vehicles have high similarity of direction and velocity, $RT_{sim}(v_i, v_j)$ is close to 1. Eq. (4) represents $H_{sim}(v_i, v_j)$ based on historic contacts. Variable $\alpha \in (0, 1]$ is defined to balance the real-time closeness and historic similarities among vehicles with the user-based collaborative filtering strategy. If v_i is a newly joined vehicle, $Nei(v_i)$ is empty initially. The similarities between v_i and other vehicles are decided by $RT_{sim}(v_i, v_j)$. The similarity calculation among vehicles can be divided into the following two situations:

(1) $Nei(v_j) = Nei(v_i)$: v_i and v_j have the same neighbor lists, i.e., each element in r_i has a counterpart in r_j. Variable $H_{sim}(v_i, v_j)$ can be directly calculated by Eq (3).

(2) $Nei(v_j) \neq Nei(v_i)$: $Nei(v_i)$ and $Nei(v_j)$ are different. Hence, r_i and r_j are updated to $\tilde{r}_i$ and $\tilde{r}_j$ with their common neighbors. For example, r_{ix} and r_{jx} cannot be removed in $\tilde{r}_i$ and $\tilde{r}_j$, only if v_x is their common neighbor. Consequently $H_{sim}(v_i, v_j)$ can be obtained with $\tilde{r}_i$ and $\tilde{r}_j$ by Eq (4). If $|\tilde{r}_i| = |\tilde{r}_j| = 0$, the historical similarity of v_i and v_j equals to 0, and $|\tilde{r}_i|$ is the number of elements in $\tilde{r}_i$.

4.1.2. Top k-nearest neighbor formulation

There are a large number of moving vehicles in certain areas during time intervals. Due to the limited storage in each vehicle, not all information from neighbors can be stored. In that case, a threshold $\lambda_{(v_i)}$ is used to decide whether there exists a strong link between vehicles. We use Eq. (5) to define threshold $\lambda_{(v_i)}$:

$$\lambda_{(v_i)} = \beta \frac{\sum_{\forall F_{sim}(v_i, v_j) > \gamma} F_{sim}(v_i, v_j)}{|\{v_j | F_{sim}(v_i, v_j) > \gamma, \forall v_j \in V\}|}, \tag{5}$$

where $\gamma \in (0, 1)$ is a tiny value used to remove some accidental and weak connections. It can eliminate the effects from accidental weak connections on the overall result, and omit some unnecessary calculations caused by accidental connections, so that the accuracy of neighbor lists based on similarity ranking can be increased. Variable β is a compensation coefficient.

In FNCF, we only select the top-k nearest neighbors, and store them in $Nei(v_i)$, i.e., the size of correlation vectors cannot exceed k. When vehicle v_i first enters into the network, neighbor list vector r_i is empty. As long as the top-k nearest neighbors are selected according to F_{sim} and Eq. (5), neighbor set $Nei(v_i)$ and overall correlation matrix R can be updated. With the k-nearest neighbor lists, vehicles tend to endue high priorities to similar neighbors in content transmission. They have a large possibility to be the same requirement for a certain file, which can reduce the file retrieve latency from numerous encounters.

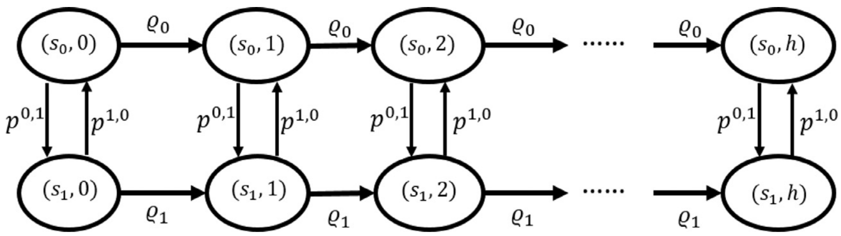


Fig. 2. Two-dimensional Markov model to download file for vehicles.

4.2. Mobility model for vehicles

Various contents can be transmitted through vehicles and RSUs. Files can be stored in RSUs. In this part, we specify the content caching procedure of FCNF. The mobility of vehicles is modelled to be a two-dimensional Markov chain. Following that, the average file downloading rate through V2V and V2R can be obtained. Finally, the file caching and updating strategies are designed based on the Markov model.

Without loss of generality, we assume that each RSU has a finite and identical file buffer, denoted by buf_{RSU}, and buffer size is represented by $|buf_{RSU}|$. Vehicles can download contents from RSUs or other vehicles. For example, vehicle $v_i \in V$ requests file $f_j \in F$. An RSU can transmit f_j to v_i directly, if $f_j \in buf_{RSU}$. Otherwise, it can retrieve the vehicles within its communication range and send the vehicle identities with the cached contents to v_i. In that case, all pieces of f_j can be downloaded through V2V. The retrieving of RSUs realizes the directed information sharing among vehicles. The transmission connections are established unstablely due to the high mobility of vehicles. Therefore, the file downloading procedure can be composed of sub-downloading sessions from different vehicles or RSUs. In order to avoid interruptions, each file can be divided into several pieces. A vehicle can finish file downloading as long as it has collected all the file pieces.

In an urban content transmission system, files are stored in RSUs according to the real-time traffic to minimize the download latency of mobile users. To be specific, a file can be cached in RSUs only if it is popular and high available within a particular location. The proportion of vehicles storing files reflects file availability. Popularity indicates the proportion of vehicles for file requesting. Let $D = \{d_{ij}\}_{m\times n}$ be the file downloading requirements from vehicles, satisfying $d_{ij} \in \{0, 1\}, i \in \{0, 1, \ldots, m\}$ and $j \in \{0, 1, \ldots, n\}$. If $d_{ij} = 1$, it means that v_i issues a request for downloading file f_j, and vice verse. The availability and popularity vectors of file set F are denoted by $A = \{a_0, a_2, \ldots, a_n\}$ and $E = \{e_0, e_2, \ldots, e_n\}$, respectively. The above two vectors are stored in local RSUs. For file $f_j \in F$ and a target RSU, V represents the local vehicle set, in which vehicles are in the coverage of RSUs. Therefore, a_j and e_j can be calculated by Eqs. (6) and (7), respectively:

$$a_j = \frac{|\{v_i|\ f_j \in buf(v_i)\ and\ v_i \in V\}|}{|V|}, \tag{6}$$

$$e_j = \frac{\sum_{v_i\in V}\sum_{i<m,j<n} d_{ij}}{|\{v_i|\ \forall f_q \in F,\ \ v_i \in V,\ \ d_{i,q} > 0\}|}, \tag{7}$$

where $buf(v_i)$ is the content buffer of v_i, and d_{ij} in Eq. (7) represents a local download request. Availability a_j can be measured by the number of vehicles interested in f_j, including the carriers and the sources of f_j. Meanwhile, its popularity lies on the number of vehicles requesting f_j.

In order to improve efficiency of V2R, we update availability vector A and popularity vector E with local RSUs in a distributed manner. RSUs collect the caching and requesting information of files from passing vehicles. Then, they update the availability and popularity vectors and send them to vehicles.

As shown in Eq. (1), we aim to minimize file downloading latency. When $d_{ij} = 1$ and v_i is within the coverage of one RSU containing f_j, v_i can access f_j through V2R communication pattern. Otherwise, it can only communicate with vehicles. We utilize a parameter $y_i(t) \in \{0, 1\}$ to mark the states whether v_i is in the RSU communication range. At time t, if v_i is within the RSU communication range, $y_i(t)$ equals to 1. Then, we can model file downloading process with a two-dimensional Markov chain. The file are divided into h pieces and encoded from 0 to $h - 1$.

At each time t, moving vehicle v_i can be denoted by $\pi_i(t) = (y_i(t), \chi_i(t))$, in which $y_i(t) \in \{0, 1\}$ represents the vehicle state, and $\chi_i(t) \in \{0, 1, \ldots, h\}$ is the observation set, representing the number of pieces that v_i has downloaded. Vehicle v_i changes its state π_i dynamically over time. When $\chi_i(t) = h$, the whole Markov chain of v_i terminates. The vehicular Markov chain is shown in Fig. 2. The parameter $p^{0,1}$ represents the probability that a vehicle changes its state from $y(t) = 0$ to $y(t) = 1$, so does $p^{1,0}$. The two probabilities depend on the circumstances that whether v_i is in the communication range of a target RSU. Based on the Markov model and file distribution, the average file-downloading rate can be calculated. Then, the probability of an RSU caching a file can be obtained. When $y_i = 0$, ϱ_0 is the transition rate for one file piece downloading through V2V before v_i finishes downloading file f_j. On the other hand, when v_i is in the coverage of a target RSU, ϱ_1 is the average downloading rate for one file piece through V2R. We define B_{V2V} and B_{V2R} as the normalized transmission capacities of V2V and V2R, respectively. For example, a V2V link L_{V2V} can transmit l file pieces, meaning that the transmission capacity of L_{V2V} is l. We consider that there exists u_1 vehicles within the V2V communication range, and vehicles share information with others within their communication ranges. Let ϑ_{V2V} denote the average downloading rate, and the average downloading rate ϑ_{V2V} through V2V connections can be expressed as:

$$\vartheta_{V2V} = \frac{B_{V2V}}{u_1+1} \times (1-(1-a_j)^{u_1}), \tag{8}$$

where file availability a_j represents the probability that f_j is stored in a random selected vehicles. The probability that at least one vehicle exists to cache f_j can be calculated by $(1-(1-a_j)^{u_1})$. We suppose that the time for one file piece downloading through V2V communication corresponds to an exponential distribution, that is $1/\varrho_0 = (1/h)/\vartheta_{V2V}$, where $\varrho_0 = h \times \vartheta_{V2V}$.

When v_i is within the communication range of a target RSU, v_i can download f_j through V2R communication as long as the RSU contains f_j. In this case, all the vehicles that have requested f_j will compete for the download channels. Similar to the V2V case, consider that there are u_2 vehicles completing for the V2R channels with v_i, the average download rate ϑ_{V2R} of v_i is expressed as:

$$\vartheta_{V2R} = e_j \frac{B_{V2R}}{u_2+1} + (1-e_j) \times \vartheta_{V2V}, \tag{9}$$

where e_j is the popularity of f_j. In this case, we approximate that the time of downloading one file piece is also exponentially distributed. Therefore, we can obtain $\varrho_1 = h \times \vartheta_{V2R}$.

According to the transmission rate analysis above, vehicle v_i starts with state $\pi_i(t_0) = (y_i(t_0), \chi_i(t_0))$, and its states change dynamically until all h file pieces are downloaded. In our Markov model, the durations of v_i in two states exhibit exponential distributions with $1/\zeta$ and $1/\rho$. We suppose that a target RSU caches file f_j with probability p_j, which is decided by the local availability, popularity and vehicle downloading requirements of f_j. Therefore, the probability that v_i with $\pi_i(t) = (1, q)$, $q \le h$, can access f_j through V2R is p_j. In addition, we can indicate that $p_j h$ pieces of files can be downloaded through V2R. Variable Dt_{ij} denotes the latency of downloading one piece of f_j. Under this circumstance, we consider the vehicular mobility and Markov model comprehensively, aiming to minimize the subjective function in Eq. (1). In Fig. 2, the mobility of v_i based on the Markov model can be expressed recursively as:

$$\begin{aligned}
&\pi_i^{0,q} = \frac{1}{\zeta+\varrho_0} + \frac{\varrho_0}{\zeta+\varrho_0}\pi_i^{0,q+1} + \frac{\zeta}{\zeta+\varrho_0}\pi_i^{1,q},\\
&\pi_i^{1,q} = \frac{1}{\rho+\varrho_1} + \frac{\varrho_1}{\rho+\varrho_1}\pi_i^{1,q+1} + \frac{\rho}{\rho+\varrho_0}\pi_i^{0,q},\\
&\Rightarrow \zeta\pi_i^{0,q} + \rho\pi_i^{1,q}\\
&= \frac{\Lambda}{L} + \frac{\zeta+\rho}{L}(\zeta\varrho_0\pi_i^{0,q+1} + \rho\varrho_1\pi_i^{1,q+1}) + \frac{\varrho_0\varrho_1}{L}(\zeta\pi_i^{0,q+1} + \rho\pi_i^{1,q+1}),\\
&\Rightarrow \zeta\varrho_0\pi_i^{0,q} + \rho\varrho_1 = (h-q)(\zeta+\rho),
\end{aligned}$$

where $\Lambda = (\zeta+\rho)^2 + \zeta\varrho_1 + \rho\varrho_0$, $L = \zeta\varrho_0 + \rho\varrho_1 + \varrho_0\varrho_1$,

Hence, we have:

$$\begin{aligned}
Dt_{ij} &= \frac{1}{\zeta+\rho} \times (\zeta \times \pi_i^{0,q} + \rho \times \pi_i^{1,q})\\
&= \frac{\zeta+\rho}{\zeta\varrho_0+\rho\varrho_1} + \frac{1}{\zeta+\rho} \times \frac{\zeta\varrho_1+\rho\varrho_0}{\zeta\varrho_0+\rho\varrho_1}\\
&= \dots\\
&= \frac{\zeta+\rho}{\zeta\vartheta_0+\rho\vartheta_1} + \frac{1}{\zeta+\rho} \times \frac{\zeta\vartheta_1+\rho\vartheta_0}{\zeta\vartheta_0+\rho\vartheta_1}.
\end{aligned} \tag{10}$$

According to Eq. (10), Eq. (1) can be converted into $\sum_{v_i\in V}\sum_{f_j\in F} d_{ij} \times Dt_{ij}$. If a vehicle downloads file f_j through V2V with probability p_j, latency $T = \sum_{v_i\in V}\sum_{f_j\in F} d_{ij} \times Dt_{ij}$, which is a twice-differential convex function of p_j. Therefore, the minimization problem can be converted into a convex optimal problem. With the Karush–Kuhn–Tucker (KKT) conditions, we can obtain the optimal caching probability p_j of file f_j by:

$$p_j = \frac{\sum_{v_i\in V}\sqrt{H_j d_{ij}}}{\sum_{v_i\in V}\sum_{f_s\in F}\sqrt{H_s d_{ij}}}, \tag{11}$$

where $H_j = (\zeta+\rho)/\rho \times \vartheta_{V2V} + (1/\zeta - 1/\rho)$. In order to obtain the value of H_j, ϑ_{V2V} needs to be calculated by file availability a_j and popularity e_j, Specifically, each RSU maintains a buffer and tends to cache files with a large value of $\sqrt{H_j d_{ij}}$.

4.3. FNCF Implementation

In this part, we specify the implementation of FNCF. With the mobility of vehicles, social behaviors and the precaching strategy are both taken into consideration. At first, moving vehicles update their neighbor lists according to the similarities. RSUs cache and update files with probabilities according to Eq. (11).

4.3.1. FNCF algorithm

In FNCF, vehicles can be the sources and carriers of files. If vehicle v_i requests file f_j with h pieces, a downloading request can be sent to other vehicles and RSUs. The whole downloading procedure is demonstrated in Algorithm 1. The local connection topologies and the content requesting collection are inputed to run Algorithm 1 at edges. Then, the content transmission links can be awared and established to improve the efficiency. Finally, the output parameters are whether the whole file piece can be downloaded, and the uncached number of requests.

Algorithm 1 Content transmission in FNCF.

Input: $V, F, D, Request_{f_j}^{v_i}$
Output: $isDownloaded$, $UncachedNum$

isDownloaded=false
for v_x in V **do**
  if $d_{v_i,v_j} < d_{V2V}$ **then**
    $F_{sim}(v_i, v_x) = \alpha \times RT_{sim}(v_i, v_j) + (1-\alpha) \times H_{sim}(v_i, v_j)$
    $r_{ij} = F_{sim}(v_i, v_j)$
  end if
end for
$Nei(v_i) = topK(r_{ij})$
while isDownloaded==false **do**
  \∗ no feedback from local RSU
  for v_x in $Nei(v_i)$ **do**
    Download $Request_{f_j}^{v_i}$ from v_x
    Remove the downloaded file piece from $Request_{f_j}^{v_i}$
  end for
  if $Request_{f_j}^{v_i} == \phi$ **then**
    isDownloaded=true
    break
  end if
  \∗ received feedback from local RSU
  if $Request_{f_j}^{v_i}$ in buf_{RSU} **then**
    download $Request_{f_j}^{v_i}$ from RSU
    finishDownload=true
  else
    uncachedNum++
    Retrieve f_j from V send retrieval results to v_i
  end if
end while
if uncachedNum$\geq \mu |buf_{RSU}|$ **then**
  Update buf_{RSU} with availability and popularity vectors, i.e. A and E
end if
return

When a new file is generated or uploaded to vehicle v_i, it will be broadcasted to neighboring vehicles. Hence, v_i can download f_j from its top-k nearest neighbors in $Nei(v_i)$. We suppose that v_i can obtain h' file pieces of f_j from the neighboring vehicles through stable V2V links. If $h' >= h$, the downloading process is finished. Otherwise, v_i can only download the contents from RSUs or other vehicles. However, if local RSUs do not contain f_j, it can be obtained through V2V communication.

When a file downloading requests from v_i has been sent to an RSU, the RSU retrieves f_j in its own buffer. If $f_j \in buf_{RSU}$, the file pieces can be transmitted to v_i through V2R communication. Otherwise, the RSU retrieves f_j pieces among all the vehicles within its coverage. Then, the file pieces can be transmitted to the requesting vehicles through V2V communication. Meanwhile, v_i can record the neighboring vehicles that have transmitted f_j through V2V communication in the log files. With the file retrieval from the RSU, the availability and popularity of vectors can be updated. When the number of uncached file pieces is larger than $\mu|buf_{RSU}|$, the content buffer is updated through the file probabilities defined in Eq. (11), where $|buf_{RSU}|$ is the size of RSU buffer, and $\mu \in (0, 1)$ is a parameter defined for file updating. However, if v_i is out of the RSU coverage, the file downloading requests should be broadcasted before v_i finishes content downloading.

Table 2
Default settings of vehicles and RSUs.

Content cache storage of RSUs	400 pieces
File content cache storage of vehicles	80 pieces
Communication range of V2V d_{V2V}	250 m
Communication range of V2R d_{V2R}	300 m
Capacity of V2V communication B_{V2V}	30 pieces/s
Capacity of V2R communication B_{V2R}	40 pieces/s
γ expressed in Eq. (5)	0.05
β expressed in Eq. (5)	0.5
Message time to live (TTL)	600 min
Message piece size	No more than 5MB

4.3.2. Complexity analysis

In FNCF, we fulfill the content transmission procedure through V2V and V2R communications. Specifically, the most time-consuming procedures are neighbor list updating of vehicles and cached content updating of RSUs. The computation complexity of FNCF is analyzed as follows:

Consider there are n vehicles and m files in the edge of vehicles, the complexity of similarity calculation by Eq. (2) is a constant value $O(Ck^2)$, because the size of vehicle similarity vectors is k at most. The top k nearest neighbors are selected with the complexity $O(log(k)n)$. Hence, the complexity of neighbor updating procedure is $O(Clog(k)n)$. Meanwhile, vehicles can obtain the cached contents through V2R communication before finishing content downloading. RSUs retrieve files from all the n vehicles within their coverage. Therefore, the complexity of file retrieval is $O(nm)$. In the final part, we update buffers according to the Markov model, the availability and popularity vectors. The two vectors are updated in the procedure of file retrieval. In that case, we can update RSU buffers with $O(|buf_{RSU}|m)$, where $|buf_{RSU}|$ is a constant value. In summary, the complexity of FNCF is $O(Cnm)$.

5. Simulation validation

We evaluate the proposed FNCF algorithm based on python and matlab. The simulations are carried on data sets with the realistic vehicular traces containing GPS information, velocities and directions. The data sets contain Beijing taxi trajectories in November, 2012 and Shanghai taxi trajectories in April, 2017.

5.1. Simulation setup

RSUs are deployed uniformly in an urban area, and the distance between two RSUs is between 300 and 500 m. We focus on public data delivery, such as traffic and weather information in video and text formats. The communication ranges of V2V and V2R are 250 m and 300 m, respectively [26]. The default settings of our experiment are illustrated in Table 2.

We first carry on the experiment to train parameters α, μ and k, with which FNCF is simulated on the one month data set. During the simulations, each vehicle updates the top-k nearest neighbor list with the collaborative filtering strategies in average 2 minutes.

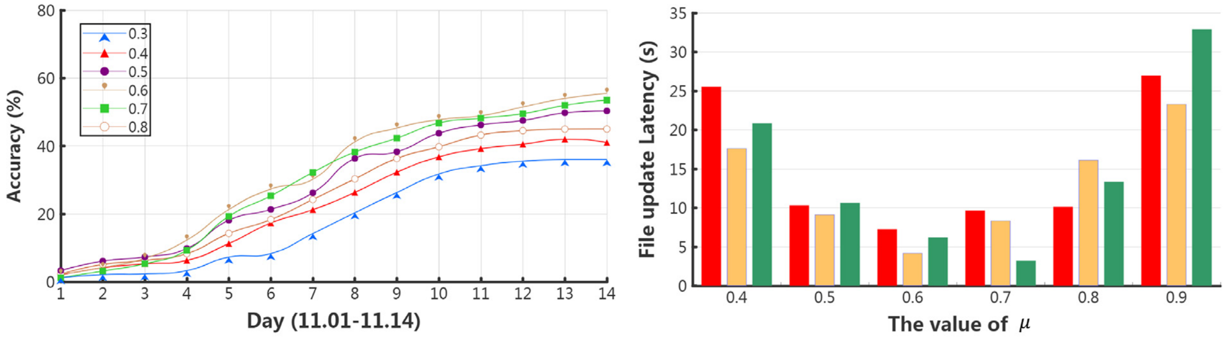
5.2. Performance metrics and comparing algorithms

Parameter α is used to balance the influences from historic and real-time contacts. If the value of α is too large, the accuracy can be low due to the accidental contacts from unknown vehicles. Otherwise, the cold starting problem can affect the accuracies of neighbor lists, resulting in that the vehicles are not sensitive to the traffic environment. On the other hand, neighbor list updating process is also time consuming. If k is too small, almost all neighboring vehicles will be replaced whenever $Nei(v_i)$ updates; otherwise, the updating process can cause a large delay due to the large similarity matrices.

In our experiment, parameter α, neighbor list size k and uncached ratio μ are set based on the data set. In the parameter training process, average neighbor accuracy Acc_{Nei} is applied to evaluate the accuracy of the neighbor prediction. For $v_i \in V$, average neighbor accuracy $Acc_{Nei}(v_i)$ is a ratio, aiming to evaluate the performance of neighbor prediction strategies. For vehicle v_i requesting f_j, there exists $s^0_{v_i}$ neighbors containing f_j, and $s^1_{v_i}$ neighbors caching f_j can be added to the top-k nearest neighbor list. Then, we can obtain the average accuracy of v_i by $Acc_{Nei}(v_i) = s^1_{v_i}/s^0_{v_i}$. Since the number of moving vehicles is large, we can calculate the average accuracy of all the moving vehicles over a day by:

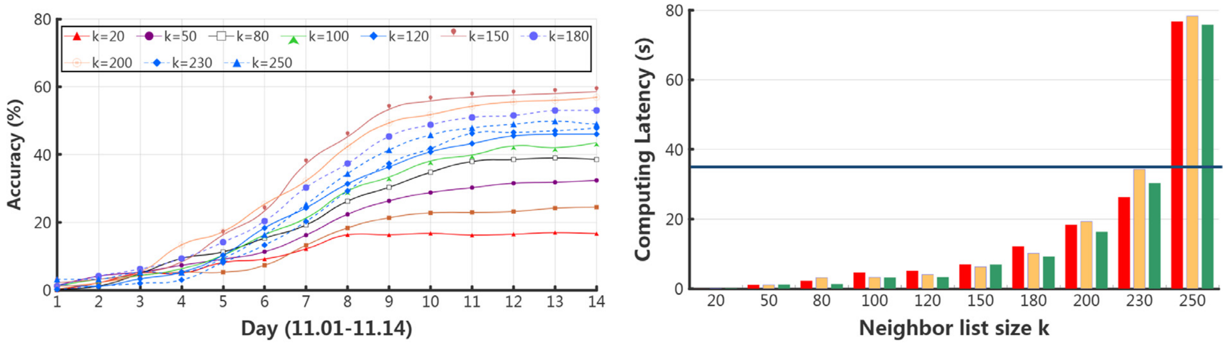
$$Accuracy = \frac{\sum_{t<=24h} \sum_{v_i \in V} Acc_{Nei,t}(v_i)}{|V|}. \tag{12}$$

The vehicle ID retrieval procedure is time consuming due to the large number of encounters. As the value of k increases, the latency of each updating process also increases. Therefore, we can make a tradeoff to evaluate the value of k between accuracy and latency.



(a) Parameter training: α with $k = 150$ (b) Parameter training: μ with $\alpha = 0.65,\ k = 150$

Fig. 3. Parameter training: α and μ.



(a) Nei accuracies with different k (b) Time cost with different k

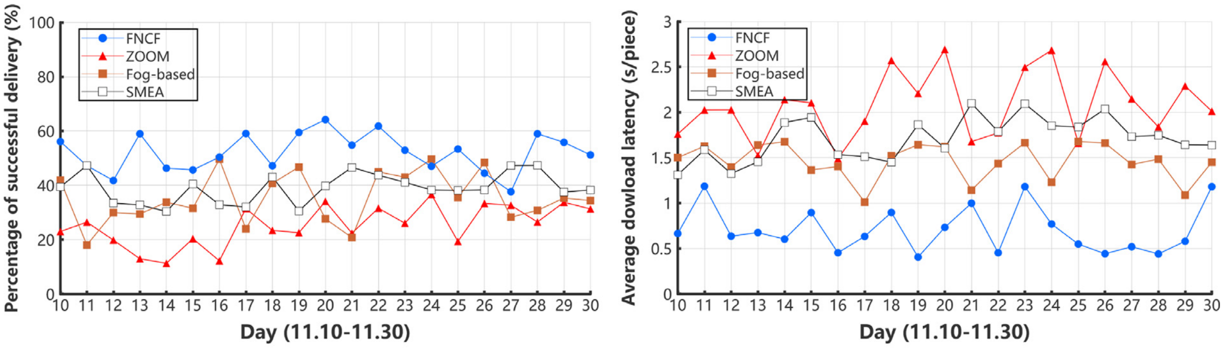
Fig. 4. Parameter training k with $\alpha = 0.65$.

We compare FNCF with three approaches, i.e., ZOOM [29], a fog-based caching algorithm [1] and SMEA [8]. ZOOM is a cloud-based method, and the social connections among vehicles are considered. The fog-based caching algorithm [1] is a decentralized content-precached strategy, but it overlooks the availability and popularity of real-time contents. In SMEA [8], the public-private key encryption-decryption mechanism is used to transmission data in channels. We verify the efficiency of our method from three aspects, including the cache hit ratio, ratios of successful content delivery and the download latency for each file piece.

5.3. Simulation result analysis

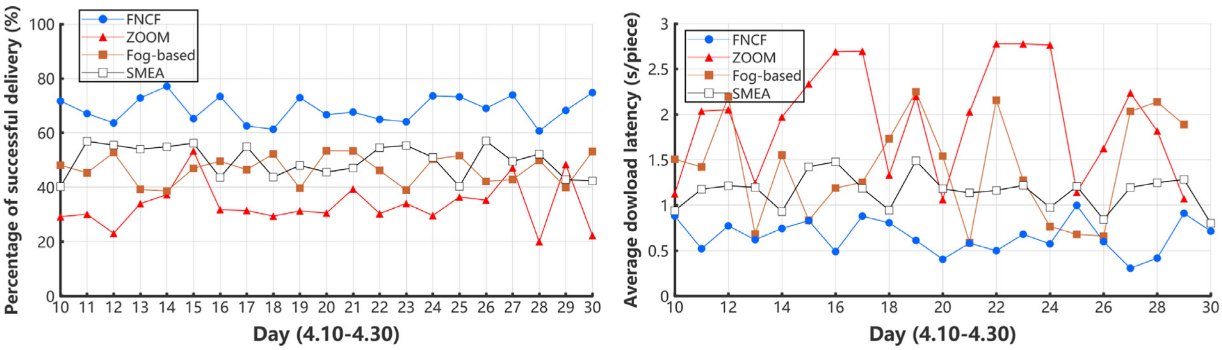
We use data of two weeks (2012.11.01–2012.11.14) in the data set of Beijing taxi trajectories to train the three parameters. We first set $k = 150$ to train α. The results are shown in Fig. 3(a). We can see that when the values of α are in the range of 0.3 and 0.6, the average accuracy increases with the increasing of α. However, when α is larger than 0.6, the average accuracies decrease as α increases. This indicates that the proper range of α is [0.6, 0.7]. According to the description above, we set $\alpha = 0.65$ and $k = 150$. Parameter μ is trained to optimize the file updating procedure in RSUs. With part of the data set (2012.11.01–2012.11.03), the results are shown in Fig. 3(b). In order to minimize the file updating delay, we set μ in the range of [0.5, 0.7]. Similarly, the training results of k are shown in Fig. 4. From Fig. 4(a), we can observe that when k is less than 150, the average accuracy increases as k increases. Meanwhile, the latency also enhances as k increases in Fig. 4(b). We set a limitation that the computing latency cannot exceed 30% of the trajectory data interval. In that case, k can be no more than 230. By considering the two criteria comprehensively, we set k within [150, 200].

We compare FNCF with three other algorithms from the aspects of cache hit ratio, delivery ratio and latency. The simulation results are shown in Fig. 5, Figs. 6 and 7. According to Figs. 5 and 6, our method outperforms others in both successful delivery ratio and latency. This shows that our proposed strategy can improve the performance of vehicular content transmissions through V2V and V2R communications. In addition, we explore that how the collaborative filtering neighbor updating policies influence the content transmission. Neighboring node prediction makes full use of social behaviors that similar vehicles tend to share the same message, by which the latency of information transmission can be reduced and the successful transmission ratios can be improved. As Fig. 7 shows, FNCF achieves a earlier and larger convergence in cache hit ration comparing with that of the fog-based method.



(a) Delivery ratio comparison (b) Latency comparison for each file piece

Fig. 5. Performance evaluation in Beijing taxi data set: Fig. 5(a): Comparison of successful delivery ratios over days (2012.11.10–2012.11.30); Fig. 5(b): Comparison of average time latency for downloading a file piece over days (2012.11.10–2012.11.30).



(a) Delivery ratio comparison (b) Latency comparison for each file piece

Fig. 6. Performance evaluation in Shanghai taxi data set: Fig. 6(a): Comparison of mean successful delivery ratios over days (2017.4.10–2017.4.30); Fig. 6(b): Comparison of average time latency for downloading a file piece over days (2017.4.10–2017.4.30).

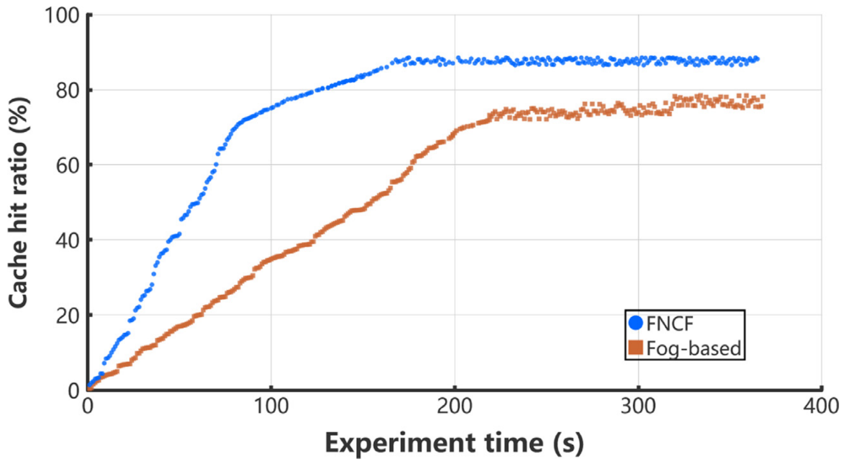


Fig. 7. Cache hit ratio comparison in Beijing taxi data set (2012.11.15).

As the prediction accuracy increases in Fig. 8, the latency of content transmission decreases gradually, and it is reduced obviously when the accuracy is between 20% and 40%. Similarly, with the increasing of accuracy, the successful data delivery ratios increase, and begin to converge when the accuracy is close to 40%. This indicates that the neighboring file sharing based on collaborative filtering can improve the information transmission effectively in the edge of vehicles. The reason for the occurrences of convergence is that some neighboring nodes are requiring the target files at the same time, and they cannot share files.

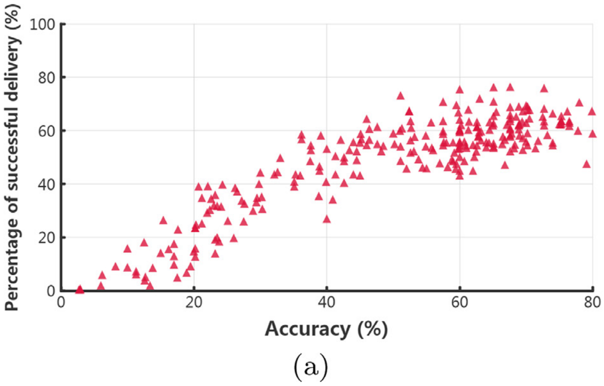


Fig. 8. Performance evaluation with prediction accuracy: Fig. 8(a): Relationship between neighbor prediction accuracy and download latency; Fig. 8(b): Relationship between neighbor prediction accuracy and successful delivery ratio.

6. Conclusion

In order to minimize the content transmission latency in edge of vehicles, we construct a novel architecture by considering fog-computing and collaborative filtering strategy to optimize the content delivery based on V2V and V2R communications. The V2V communication is designed with a neighbor list kept in vehicles, which stores the information of top-k closest neighboring vehicles and can be updated with the collective filtering algorithm. Moreover, a two-dimensional Markov chain is modelled for the moving vehicles. Under the premise of minimizing the downloading latency of V2R communications, the probabilities of files stored in RSUs are obtained in the Markov model. As a result, the contents are cached and updated under fog-based RSUs in a decentralized way. The simulation results verify the effectiveness and efficiency of our method. In the future work, we will consider data permissions to access content downloading.

Declaration of Competing Interest

We declare no conflict of interest exits in the submission of this manuscript, and the manuscript is approved by all authors for publication. We would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

Acknowledgments

This work is supported by the National Nature Science Foundation of China under Grants 61632014, 61671092, 61771120 and 61802159, China Postdoctoral Science Foundation under Grant 2018T110210, Fundamental Research Funds for the Central Universities under Grant DUT19JC18 and Grant DUT18JC09, and Science and Technology planning project of Shenzhen (JCYJ20170818111012390).

References

[1] E. Bastug, M. Bennis, M. Debbah, Living on the edge: the role of proactive caching in 5G wireless networks, IEEE Commun. Mag. 52 (8) (2014) 82–89, doi:10.1109/MCOM.2014.6871674.
[2] C. Gkantsidis, M. Mihail, A. Saberi, Random walks in peer-to-peer networks, in: Proceedings of the Joint Conference of the IEEE Computer and Communications Societies, 2004, pp. 241–263, doi:10.1109/INFCOM.2004.1354487.
[3] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: a viewpoint of vehicles as the infrastructures, IEEE Trans. Veh. Technol. 65 (6) (2016) 3860–3873, doi:10.1109/TVT.2016.2532863.
[4] S. Huang, B. Guo, W. Ju, X. Zhang, J. Han, C. Phillips, J. Zhang, W. Gu, A novel framework and the application mechanism with cooperation of control and management in multi-domain WSON, J. Netw. Syst. Manag. 21 (3) (2013) 453–473.
[5] S. Huang, Y. Zhou, S. Yin, Q. Kong, M. Zhang, Y. Zhao, J. Zhang, W. Gu, Fragmentation assessment based on-line routing and spectrum allocation for intra-data-center networks with centralized control, Opt. Switch. Netw. 14 (2014) 274–281.
[6] G.C. Index, Cisco global cloud index: forecast and methodology, 2016–2021 White paper.
[7] H. Ji, V.C.S. Lee, C.Y. Chow, K. Liu, G. Wu, Coding-based cooperative caching in on-demand data broadcast environments, Inf. Sci. s 385–386 (C) (2017) 138–156, doi:10.1016/j.ins.2017.01.012.
[8] T. Limbasiya, D. Das, Secure message transmission algorithm for vehicle to vehicle (V2V) communication, in: Proceedings of the IEEE Region 10 Conference (TENCON), 2016, pp. 2507–2512, doi:10.1109/TENCON.2016.7848485.
[9] Y. Liu, Y. Peng, B. Wang, S. Yao, Z. Liu, Review on cyber-physical systems, IEEE/CAA J. Autom. Sin. 4 (1) (2017) 27–40, doi:10.1109/JAS.2017.7510349.
[10] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, M.S. Obaidat, A cooperative quality-aware service access system for social internet of vehicles, IEEE Internet Things J. 5 (2018) 2506–2517, doi:10.1109/JIOT.2017.2764259.
[11] Z. Ning, X. Kong, F. Xia, W. Hou, X. Wang, Green and sustainable cloud of things: enabling collaborative edge computing, IEEE Commun. Mag. 57 (1) (2019) 72–78, doi:10.1109/MCOM.2018.1700895.
[12] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, W. Zhang, CAIS: a copy adjustable incentive scheme in community-based socially aware networking, IEEE Trans. Veh. Technol. 66 (4) (2016) 3406–3419.

[13] Z. Ning, X. Wang, X. Kong, W. Hou, A social-aware group formation framework for information diffusion in narrowband internet of things, IEEE Internet Things J. 5 (3) (2017) 1527–1538.
[14] Z. Ning, F. Xia, N. Ullah, X. Kong, X. Hu, Vehicular social networks: enabling smart mobility, IEEE Commun. Mag. 55 (5) (2017) 16–55.
[15] Q. Song, Z. Ning, S. Wang, A. Jamalipour, Link stability estimation based on link connectivity changes in mobile ad-hoc networks, J. Netw. Comput. Appl. 35 (6) (2012) 2051–2058.
[16] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Efficient routing in intermittently connected mobile networks: the multiple-copy case, IEEE/ACM Trans. Netw. 16 (1) (2008) 63–76, doi:10.1109/TNET.2007.897964.
[17] W. Sun, D. Yuan, E.G. Strm, F. Brnnstrm, Cluster-based radio resource management for D2D-supported safety-critical V2X communications, IEEE Trans. Wirel. Commun. 15 (4) (2016) 2756–2769, doi:10.1109/TWC.2015.2509978.
[18] A.M. Vegni, V. Loscr, A survey on vehicular social networks, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2397–2419, doi:10.1109/COMST.2015.2453481.
[19] T. Wang, Z. Peng, S. Wen, Y. Lai, W. Jia, Y. Cai, H. Tian, Y. Chen, Reliable wireless connections for fast-moving rail users based on a chained fog structure, Inf. Sci. 379 (2016) 160–176, doi:10.1016/j.ins.2016.06.031.
[20] X. Wang, Z. Ning, X. Hu, E. Ngai, L. Wang, B. Hu, Y. Kwok, A city-wide real-time traffic management system: enabling crowdsensing in social internet of vehicles, IEEE Commun. Mag. 56 (2018) 19–25, doi:10.1109/MWC.2018.1700441.
[21] X. Wang, Z. Ning, L. Wang, Offloading in internet of vehicles: a fog-enabled real-time traffic management system, IEEE Trans. Ind. Inform. 14 (10) (2018) 4568–4578, doi:10.1109/TII.2018.2816590.
[22] X. Wang, Z. Ning, M. Zhou, X. Hu, L. Wang, B. Hu, R.Y.K. Kwok, Y. Guo, A privacy-preserving message forwarding framework for opportunistic cloud of things, IEEE Internet Things J. 5 (6) (2018) 5281–5295, doi:10.1109/JIOT.2018.2864782.
[23] X. Wang, Z. Ning, M.C. Zhou, X. Hu, L. Wang, Y. Zhang, F.R. Yu, B. Hu, Privacy-preserving content dissemination for vehicular social networks: challenges and solutions, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1314–1345, doi:10.1109/COMST.2018.2882064.
[24] X. Wang, C. Wang, J. Zhang, M. Zhou, C. Jiang, Improved rule installation for real-time query service in software-defined internet of vehicles, IEEE Trans. Intell. Transp. Syst. 18 (2) (2017) 225–235, doi:10.1109/TITS.2016.2543600.
[25] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, X. Shen, Internet of vehicles in big data era, IEEE/CAA J. Autom. Sin. 5 (1) (2018) 19–35, doi:10.1109/JAS.2017.7510736.
[26] L. Yang, J. Xu, G. Wu, J. Guo, Road probing: RSU assisted data collection in vehicular networks, in: Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, 2009, pp. 2717–2720, doi:10.1109/WICOM.2009.5303353.
[27] D. Zhang, S. Cai, F. Ye, Y.W. Si, T.T. Nguyen, A hybrid algorithm for a vehicle routing problem with realistic constraints, Inf. Sci. 394–395 (C) (2017) 167–182, doi:10.1016/j.ins.2017.02.028.
[28] P. Zhang, M. Zhou, G. Fortino, Security and trust issues in fog computing: a survey, Future Gener. Comput. Syst. 88 (2018) 16–27, doi:10.1016/j.future.2018.05.008.
[29] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, X.S. Shen, Zoom: scaling the mobility for fast opportunistic forwarding in vehicular networks, in: Proceedings of the IEEE INFOCOM, 2013, pp. 2832–2840, doi:10.1109/INFCOM.2013.6567093.