Cooperative Edge Caching With Location-Based and Popular Contents for Vehicular Networks

Jiayin Chen, Huaqing Wu, *Student Member, IEEE*, Peng Yang, *Member, IEEE*, Feng Lyu, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—In this article, we propose a cooperative edge caching scheme, which allows vehicles to fetch one content from multiple caching servers cooperatively. In specific, we consider two types of vehicular content requests, i.e., location-based and popular contents, with different delay requirements. Both types of contents are encoded according to fountain code and cooperatively cached at multiple servers. The proposed scheme can be optimized by finding an optimal cooperative content placement that determines the placing locations and proportions for all contents. To this end, we first analyze the upper bound proportion of content caching at a single server, which is determined by both the downloading rate and the association duration when the vehicle drives through the server's coverage. For both types of contents, the respective theoretical analysis of transmission delay and service cost (including content caching and transmission cost) are provided. We then formulate an optimization problem of cooperative content placement to minimize the overall transmission delay and service cost. As the problem is a multi-objective multi-dimensional multi-choice knapsack problem, which is proved to be NP-hard, we devise an ant colony optimization-based algorithm to solve the problem and achieve a near-optimal solution. Simulation results are provided to validate the performance of the proposed algorithm, including its convergence and optimality of caching, while guaranteeing low transmission delay and service cost.**

***Index Terms*—Cooperative edge caching, vehicular network, differential content placement, ant colony optimization.**

I. Introduction

RECENT development of vehicular technologies has led to the era of autonomous vehicles, and both consumers and government authorities are looking forward to autonomous vehicles. Yet, it is challenging to meet the stringent requirements of autonomous vehicles solely based on the on-board sensors in hazardous conditions, e.g., collision avoidance with poor visibility [1]. Information exchange with external entities is also necessary, such as hazard broadcasting and high-quality maps downloading [2]. In addition to driving-related applications, passengers also have strong demands for mobile applications in connected vehicles [3]. Newly developed vehicular applications (e.g., in-car entertainment and mobile advertising) require high-volume data transmission, imposing substantial pressure on vehicular networks that solely rely on cellular networks to fetch data from the Internet. To improve the network capacity, Heterogeneous Vehicular Networks (H-VNets) have been proposed, in which both base stations (BSs) and roadside units (RSUs) can provide network connections, resulting in significantly reduced vehicular communication cost [4], [5].

Edge caching is proposed to reduce both the backhaul traffic and the transmission time for high-volume data delivery [6]. To facilitate content delivery, it is essential to develop a content placement scheme for edge caching servers [7], considering the intermittent connection of moving vehicles. In particular, to satisfy the service delay requirements of the vehicles driving through different edge nodes (e.g., RSUs), edge cooperation has been introduced and the cooperative content placing scheme has been proposed [8], [9]. However, as edge resources are constrained, to adapt to the high mobility of vehicles, caching on both vehicles and RSUs should be considered [10], [11]. Both the cooperation among RSUs and the cooperation between RSUs and vehicles have been investigated. However, as vehicular connections are intermittent due to the limited coverage range of edge servers, content downloading time can be unacceptable. Within the interlaced coverage of multi-tier edge caching servers (e.g., BSs and RSUs) in H-VNets, the cross-tier cooperation among different servers can provide seamless connections to facilitate content downloading. Meanwhile, within the overlapping coverage, differential communication features and caching capacities of multiple servers are considered in content caching. Hence, compared with cooperation between RSUs, a more fine-grained content placement scheme is required, e.g., determining the proportions of cached contents on multi-tier edge servers. In addition, vehicle's high mobility renders the design further intractable, as it constrains the vehicular connection duration. A precise vehicle mobility model is thus required for the connection duration analysis.

Besides, most existing works consider vehicular downloading applications with a single quality of service (QoS) metric, such as the delivery deadline or the downloading rate. However, the QoS metrics for different applications are diverse [12]. For instance, the safety-related vehicular applications call for low latency, while entertainment services desire high throughput. Therefore, different QoS metrics for two content services need to be considered: 1) the location-based contents (e.g., HD map downloaded for driving assistance applications), which require a stringent downloading time, and 2) popular contents (e.g., live

Manuscript received February 25, 2020; revised June 10, 2020; accepted June 15, 2020. Date of publication June 23, 2020; date of current version October 13, 2020. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 91638204 and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The review of this article was coordinated by Dr. F. Tang. *(Corresponding author: Peng Yang.)*

Jiayin Chen, Huaqing Wu, and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: j648chen@uwaterloo.ca; h272wu@uwaterloo.ca; sshen@uwaterloo.ca).

PengYang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yangpeng@hust.edu.cn).

Feng Lyu is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: fenglyu@csu.edu.cn).

Digital Object Identifier 10.1109/TVT.2020.3004720

video streaming), which require to be delivered before a less stringent service deadline. Moreover, through placing the contents on edge servers equipped with low-cost wireless resources (e.g., WiFi), the cost of data transmission can be saved as well. Due to the differential QoS metrics, the objective functions for diverse services should be derived separately, and both cost and QoS should be considered in the content placement problem formulation. This lead to the high complexity of problem formulation and difficulty of multi-objective solution.

In this article, based on the H-VNets with multi-tier edge caching servers (i.e., BS and RSU), a cooperative edge caching scheme is proposed to accommodate the delivery services for both location-based and popular contents. In particular, one content can be cached at multiple servers cooperatively. Considering the cross-tier cooperation and vehicle mobility, the duration of vehicles driving through the coverage of RSU and BS is first analyzed, which determines the upper bound of vehicular downloading time from a single server. To meet differentiated QoS requirements, the content placement schemes of the two services are designed respectively. For a location-based content requiring minimal downloading time, the delay and service cost for each possible content placement scheme are derived. For popular content delivery with a service deadline, we analyze the delay guaranteed minimum content placing requirement for all possible caching modes, i.e., placing one content at both BS and RSU, only at BS, or only at RSU. Accordingly, the service cost is also derived for each mode.

Based on the theoretical analysis, a cooperative content placement problem is then formulated to jointly minimize the transmission delay of the location-based contents and the service cost of both types of contents. The formulated problem turns out to be an NP-hard multi-objective multi-dimensional multi-choice knapsack problem (MMKP), and we propose an ant colony optimization (ACO) based algorithm to solve it. Then, the fine-grained cooperative content placement is obtained, including the caching proportions of each content at different caching servers, which can achieve a near-optimal performance in terms of delay and service cost.

The contributions in this article are as follows:

1) *Cooperative edge caching scheme* – We propose a cooperative caching scheme, in which a vehicle in H-VNet can fetch a content from multiple edge servers while driving along the road. This cooperation among edge servers can not only reduce transmission delay through seamless content delivery, but also save service cost by caching contents in low-cost servers.
2) *Theoretical analysis of delay and cost* – We analyze and derive the content transmission delay for two types of contents. For location-based contents, given a content placement scheme, we derive the delay under the vehicle mobility and communication models, which can be optimized by determining an optimal content placement from the available set. For popular contents with a specified delay requirement, we analyze the delay guaranteed minimum content placing requirement for all possible caching modes. Thus, the placement scheme of each popular content can be optimized by selecting an optimal caching mode. Likewise, the service cost is also derived.
3) *Cooperative content placement algorithm design* – Based on the delay and cost analysis, we formulate an optimization problem of cooperative content placement to jointly minimize the delay and the cost, which turns out to be a multi-objective MMKP. To solve the optimization problem with low complexity, we devise an ACO based algorithm, which can achieve a fine-grained cooperative content placement with near-optimal performance.

The remainder of this article is organized as follows. We review the related work in Section II and describe the system model in Section III. The delay and cost of cooperative content caching are theoretically analyzed in Section IV, followed by the cooperative content placement problem formulation in Section V. In Section VI, we devise the ACO-based algorithm to solve the optimization problem. Simulation results are presented in Section VII to demonstrate the performance of the proposed algorithm. Finally, conclusions and our future work are drawn in Section VIII. Important mathematical symbols are listed in Table I, where V2I stands for vehicle-to-infrastructure communication.

II. Related Work

In this section, we review the related work in two categories: the cooperative caching in vehicular networks and the cross-tier cooperation enabled edge caching.

A. Cooperative Caching for Vehicular Network

Yao *et al.* predicted the probability of vehicles arriving at different hotspots, and selected the vehicles with longer sojourn time as caching nodes [13]. Then, the decision of content replacement is made, considering the file popularity. Zhao *et al.* proposed to cache content at RSUs instead of on the vehicles, and a mobility prediction based content prefetching mechanism was proposed to improve content hit rate and reduce content delivery latency [14]. The frequent disconnection between vehicles and edge nodes makes it difficult to download the complete file during one connection. Thus, coded caching strategies (e.g., maximum distance separable (MDS) code and fountain code [15], [16]) are widely used, since they can improve the delivery reliability/flexibility and cache utilization by dividing the files into small fragments.

B. Cross-Tier Cooperation in Edge Caching

The cooperative caching in vehicular networks mainly focuses on the cooperation between vehicular cloud and RSU cloud [11], while the cooperative caching in mobile user networks takes advantage of multi-tier cellular networks (i.e., macro-cell and small-cell) [17], [18]. The caching strategy was proposed by leveraging the cross-tier cooperation, including the vertical cooperation between a macro base station (MBS) and a small base station (SBS), and the horizontal cooperation between SBSs [19]. Then, the mobile user can download its requested content from one or several BSs that have cached the content. To reduce the content provisioning cost and delivery delay, a cooperative caching strategy was designed, which considers content placement at the centralized MBSs and distributed SBSs [20].

Most existing works on edge caching in H-VNets consider the cooperation between edge caching servers (e.g., RSUs or BSs) and vehicular caching nodes, instead of the cross-tier cooperation between multi-tier servers (e.g., between RSUs and BSs). The cooperation between MBSs and SBSs have been widely investigated for mobile users in low-speed scenarios, in which intermittent connection rarely happens within the course of downloading one content. However, for vehicular users, the high mobility poses challenges to the cooperation, considering the frequent handover between different caching servers. Thus,

TABLE I
SUMMARY OF MATHEMATICAL SYMBOLS

Symbols	Definition
a	Amount of speed variation during each speed update interval
f	File index
l	Size of a coded content packet
n	Possible content placement scheme index
q_0	Exploitation probability of caching mode selection in Algorithm 2
s	Skewness parameter of Zipf-like popularity distribution for contents
s_B^f ($s_{R_w}^f$)	Number of precached encoded packets of file f at BS (RSU W_w)
t_B^f ($t_{R_w}^f$, t_{BL}^f)	Transmission delay for each packet of file f from BS (RSU W_w, the remote server)
$v_i\,(t)$	Speed of VU_i at time t
$x_{f,n}$	Binary variable representing the selection of n-th possible content placement scheme for file f
$B_i\,(t)$	Allocated BS bandwidth for VU_i at time t
D_B (D_R)	Communication ranges of BS (RSU)
$\overline{D}^f\,(n)$ ($\overline{C}^f\,(n)$, $T_{R_w}^f\,(n)$)	Required service delay (cost, access resources) for file f under the n-th possible content placement scheme
F (M)	Total number of files (Number of popular content files)
N (W)	Number of vehicles (RSUs) within the coverage of the BS
N_C^f	Number of possible content placement schemes for file f
P_f (P_{V2I}^f)	Probability of file f being requested (downloaded through V2I connection)
$\overline{R_B^L}$ (R_W)	Average transmission rate from the BS (remote server)
$R_i\,(t)$	Distance between VU_i and BS at time t
S_{MBS} (S_{RSU}, S_{VU})	Storage capacity at each BS (RSU, vehicle)
S_f	Required number of encoded packets for file f recovery
TN_B^f ($TN_{R_w}^f$, TN_{BL}^f)	Number of packets downloaded from BS (RSU W_w, the remote server) of file f
VU_i (W_w)	The i-th vehicle (w-th RSU) of the vehicle (RSU) set
$V_{\min}$ ($V_{\max}$)	Minimal (Maximal) speed of vehicles
W_D (W_C)	Weight for delay (service cost) as performance metric
X_A (X_I)	Number of ants (iterations) in Algorithm 1
λ	Arrival rate of vehicles
σ^2	Additive Gaussian noise power density of link between vehicle and BS
$\tau_{\max}$ ($\tau_{\min}$)	Upper (Lower) bound of pheromone value in Algorithm 1
$\tau_{f,n}$ ($\eta_{f,n}$, $p_{f,n}$)	Pheromone value (Heuristic information, Probability) of choosing mode n for file f
$\mathcal{A}_{ji}$	Amount of data transmitted from VU_j to VU_i
$\mathbb{LF}^w$ ($\mathbb{PF}$)	Set of location-based content files under RSU W_w (popular content files)
ND	Non-dominated content placement scheme set in Algorithm 1
$\mathcal{T}_R$	Amount of downloading data provided by RSU

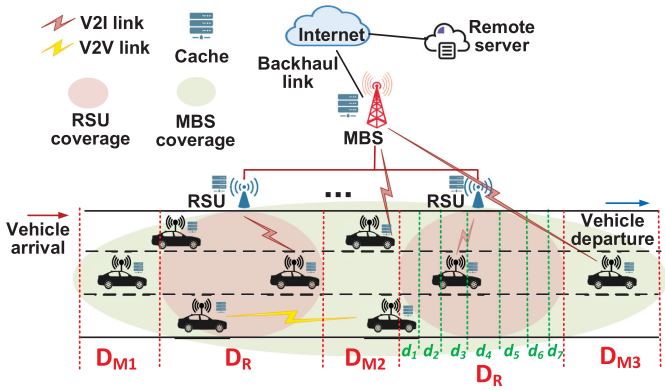


Fig. 1. H-VNet system model.

to guarantee delay requirements and save costs for vehicular content delivery service, we focus on placing contents on cross-tier caching servers cooperatively.

III. System Model

A. System Overview

As shown in Fig. 1, we consider a highway multi-tier H-VNet, including vehicular, RSU, and cellular tiers. For cellular tier, it includes MBS and the backhaul link that connects to the Internet and the remote server. To support the content delivery services for vehicles, we develop a cooperative edge caching scheme for the H-VNets with multi-tier edge caching servers (i.e., at MBS and RSUs), in which one content can be cached at multiple servers cooperatively. If the requested content has been cached at the MBS and/or RSUs, it can be delivered to the vehicle by the MBS and/or RSUs, depending on the trajectory of the vehicle and the locations of cached contents. Compared with fetching content from the remote server, downloading content from edge servers can effectively reduce both delivery delay and cost. Due to the mobility of vehicles, cooperatively caching one content at multiple servers can further improve the caching efficiency, where vehicles can download content when they drive through the coverage area of edge nodes.

Without loss of generality, we focus on the coverage area of one MBS, within which W RSUs are deployed along the road using WiFi access technology. The coverage areas of different RSUs are assumed to be nonoverlapping, and the communication ranges of RSU and MBS are denoted by D_R and D_B, respectively. Let $\mathbb{W}$ be the set of RSUs, i.e., $\mathbb{W} = \{W_1, W_2, ..., W_W\}$. Let $\mathbb{V}$ be the set of N vehicles within the coverage of the MBS, i.e., $\mathbb{V} = \{VU_1, VU_2, ..., VU_N\}$. MBS, RSUs, and vehicles are equipped with caching capability, with the storage capacity denoted by S_{MBS}, S_{RSU}, and S_{VU}, respectively. A network management controller is deployed at MBS, which collects information from vehicles and edge servers and makes decisions on content caching [21]. To proceed, the vehicle mobility model and the communication model of H-VNet will be introduced, followed by the content request and caching mechanism.

B. Vehicle Mobility Model

We consider H-VNet with N moving vehicles requesting file downloading services. We assume the arrival of vehicles follows Poisson distribution with arrival rate λ, and the arriving time interval between two adjacent vehicles, ΔT, follows an exponential distribution, i.e., $\Delta T \sim \exp(1/\lambda)$ [22].

In our model, N vehicles are moving towards the same direction. To characterize the real highway environment, free driving vehicles with speed constrained in $[V_{\min}, V_{\max}]$ are considered [23], and the speed is updated periodically. In particular, consider the time span is slotted with equal slot duration Δt, the speed is updated at the beginning of each slot, and remains invariant subsequently. The speed update of VU_i is

$$v_i\left(t+\Delta t\right) = v_i(t) + \alpha_i(t)\cdot a\cdot \Delta t, \tag{1}$$

where $v_i(t)$ is the speed of VU_i at time t, $\alpha_i(t) \in \{1, 0, -1\}$ is a uniformly distributed random parameter that represents the adjustment of acceleration or deceleration, and a is a constant representing the amount of speed variation [23].

This update process can be described as a state transition following Markov model, in which all available speed levels of vehicles are extracted as a set of ordered states with equal transition probabilities. To describe the statistical characteristics of vehicle mobility, we obtain the expectation, $E[v] = (V_{\min} + V_{\max})/2$, and the variance ($Var[v]$) of $v_i(t)$ based on the analysis of the state transition probability matrix,

$$Var\left[v\right] = \frac{n^3 - 2n^2 + 3n - 2}{12n - 8}a^2, \tag{2}$$

where $n = (V_{\max} - V_{\min})/2$.

We consider the headway distance from VU_i to VU_j as a directional variable $D(t)$, which is positive if VU_i is behind VU_j in the moving direction. A G/G/1 queue model is built to predict headway distance, in which the queue length represents the distance [22]. Given the initial headway distance, d_0, the probability density function of $D(t)$ at time t is

$$\begin{aligned} f_D\left(x; d_0, t\right) &= Pr\left\{x \le D(t) \le x + \Delta x | D(0) = d_0\right\} \\ &= \frac{1}{\sqrt{4\pi Var\left[v\right]t}} \exp\left\{-\frac{\left(x-d_0\right)^2}{4Var\left[v\right]t}\right\}. \end{aligned} \tag{3}$$

C. V2I Communication Model

1) V2B Communication Model: If VU_i is served by the MBS, the wireless transmission rate at t, denoted by $R_{B,i}(t)$, is

$$R_{B,i}(t) = B_i(t)\cdot \log_2\left(1 + \frac{P_T \cdot L\left(R_i(t)\right)}{\sigma^2}\right), \tag{4}$$

where $B_i(t)$ is the allocated bandwidth for VU_i at time t, P_T is the transmit power density of the MBS, σ^2 is the additive Gaussian noise power density, $R_i(t)$ is the distance between VU_i and MBS in kilometers at time t, and $L(R_i(t))$ is the path loss between VU_i and MBS [24] given by

$$\begin{aligned} L\left(R_i(t)\right) &= 40(1 - 4\cdot 10^{-3}\cdot Dhb)\log_{10}(R_i(t)) \\ &\quad - 18\log_{10}(Dhb) + 21\log_{10}(f) + 80\text{dB} + X, \end{aligned} \tag{5}$$

where f is the carrier frequency in MHz, Dhb is the antenna height of infrastructure in meters that measured from the average rooftop level, and X in dB represents shadowing channel fading, which follows Log-normal distribution. Note that, vehicles download content through orthogonal channels and the coverage area of different MBSs along the highway are non-overlapping, so the interference is negligible in vehicle to MBS (V2B) communication model.

TABLE II
ADAPTIVE TRANSMISSION RATE OF WIFI ACCESS POINT [25]

Zone	1	2	3	4	5	6	7
d_k (m)	25	30	40	60	40	30	25
R_k $(Mbps)$	1	2	5.5	11	5.5	2	1

We assume the number of vehicles that enter and leave the MBS are equalized, so we use $N = \frac{D_B}{\frac{1}{\lambda}\cdot E[v]}\cdot N_L$ to denote the number of vehicles in MBS coverage, where D_B is the MBS coverage range and N_L is the number of lanes of the highway. Considering the worst case that all the vehicles in the MBS coverage request MBS access, we can get the lower bound of $R_{B,i}(t)$ as $R^L_{B,i}(t)$, where $B_i(t) = B/N$ and B is the available bandwidth of MBS. We use this lower bound to estimate the average transmission rate of MBS as R^L_B. Considering the fixed relative location of MBS and the midpoint of highway, we replace $L(R_i(t))$ by $\check{L}(x)$, where x is the distance from vehicle to the midpoint, $x \in [0, D_B/2]$, it holds that

$$\overline{R^L_B} = \frac{2}{D_B}\int_0^{D_B/2} \frac{B}{N}\cdot \log_2\left(1 + \frac{P_T\cdot \check{L}(x)}{\sigma^2}\right) dx. \tag{6}$$

If the vehicle is connected to the Internet through MBS and its wired backhaul link, the transmission rate is determined by the backhaul link R_W.

2) V2R Communication Model: The communication model between vehicles and RSUs with WiFi access technology is investigated in [25] as an adaptive vehicle to RSU (V2R) transmission model, in which the coverage area is divided into K zones as shown in Fig. 1, and the transmission rate achieved within the k-th zone is denoted as R_k. According to the IEEE 802.11b standard [26], $K = 7$ and the rates through the RSU coverage is symmetric, which is given in Table II with the range of each zone (d_k). To simplify the analysis, we consider a MAC protocol that the connection time for all the vehicles within the coverage is equally allocated, which means the transmission rate of the k-th zone is equally allocated to the associated vehicles. Therefore, the bit rate of VU_i at instant t is expressed as $R_{R,i}(t) = \frac{R_k}{N_k(t)}$, where VU_i is driving through the k-th zone at time t and $N_k(t)$ is the number of vehicles in the k-th zone at time t. Similar to MBS, the number of vehicles that enter and leave the k-th zone are considered to be equivalent, so we use N_k to replace $N_k(t)$ in the following analysis.

Each RSU provides download service to all the vehicles within its coverage, and the amount of data it can provide is

$$\mathcal{T}_R = \sum_{k=1}^{7} N_k\cdot \frac{d_k}{E\left[v\right]}\cdot \frac{R_k}{N_k} = \sum_{k=1}^{7}\frac{d_k\cdot R_k}{E\left[v\right]}, \tag{7}$$

where the duration of VU_i staying in the k-th zone is estimated by $d_k/E[v]$.

Assume that there is an admission control buffer with a size of $\mathcal{T}_R$ in each RSU, which stores the content that will be downloaded by vehicles in its coverage. If a new request of a vehicle is accepted, the requested content will be added to the buffer. Then, during the content transmission, downloaded data will be removed from the buffer. Thus, the admission decision of each vehicle to the RSU can be determined by this buffer. When a new request arrives, it will be accepted if there is enough buffer space for the requested content. A vehicle can always download

the content within the RSU coverage as long as the buffer is not overflowed.

D. V2V Communication Model

When a vehicle requests for content download, it firstly estimates the probability of successful transmission through vehicle-to vehicle (V2V) communication, and then makes the decision on downloading the content either from other vehicles or from edge servers, e.g., the MBS or RSUs.

We consider the V2V communication on the DSRC spectrum from 5.850 to 5.925 GHz. In particular, the physical layer operation is specified by the IEEE 802.11p standard, while for the MAC layer, the IEEE 802.11b DCF scheme is applied for channel contention model. Given that the file size of requested content by VU_i is F_i, the data amount, $\mathcal{A}_{ji}$, transmitted from VU_j to VU_i during the time period of T_d ($T_d > 0$), can be evaluated following [22]. Then, the successful probability of transmitting at least F_i bits of data from VU_j to VU_i with the time constraint T_d is bounded by

$$Pr\left\{\mathcal{A}_{ji} \geq F_i\right\} \geq \frac{\left[E\left(\mathcal{A}_{ji}\right) - F_i\right]^2}{Var\left(\mathcal{A}_{ji}\right) + \left[E\left(\mathcal{A}_{ji}\right) - F_i\right]^2}, \quad (8)$$

where $E(\mathcal{A}_{ji})$ is the mean of $\mathcal{A}_{ji}$, and $Var(\mathcal{A}_{ji})$ is an upper bound of the variance of $\mathcal{A}_{ji}$. Both mean and variance are dependent on the headway distance in (3). This successful probability indicates whether the content can be downloaded from neighboring vehicles within T_d. For location-based content, T_d is set to be proportional to the file size, representing the average transmission delay through V2I connections. For popular content, T_d is set as the delivery delay requirement.

E. Content Request and Caching Model

1) Content Popularity Model: We consider two types of contents that are requested by vehicles: popular contents (e.g., news and video service) and location-based contents (e.g., HD map and local commercial information), denoted by PF and LF, respectively. The location-based contents provide local information, which is always needed by the vehicles around the location. For example, if an RSU is deployed near a shopping center, the advertisements are very likely to be requested by the vehicles driving into the RSU coverage. Thus, we assume a vehicle may request for location-based content when it enters the coverage of an RSU, which is dependent on the location. Within the coverage of RSU W_w, there are N_w location-based content files, and the sets of these files and the corresponding sizes are denoted by $\mathbb{LF}^w = \{LF_1^w, LF_2^w, ..., LF_{N_w}^w\}$ and $\mathbb{S}^{Lw} = \{S_1^{Lw}, S_2^{Lw}, ..., S_{N_w}^{Lw}\}$, respectively. However, the request for popular content may be generated within MBS coverage, and the sets of all the M popular files and the corresponding sizes are denoted by $\mathbb{PF} = \{PF_1, PF_2, ..., PF_M\}$ and $\mathbb{S}^{PF} = \{S_1^{PF}, S_2^{PF}, ..., S_M^{PF}\}$, respectively.

Due to the features of LF contents, the file set for a vehicle is dependent on location. If a vehicle sends a content request within the coverage of RSU W_w, an integrated file set $\mathbb{F}^w = \{1, ..., F_w\}$ is established as its overall content set, consisting of $\mathbb{LF}^w$ and $\mathbb{PF}$. The number of files in $\mathbb{F}^w$ is denoted by $F_w = M + N_w$. If a vehicle sends a request at a location not covered by any RSUs, which means the vehicle is not going to request for a location-based content, the file set $\mathbb{F}^0 = \mathbb{PF} = \{1, ..., F_0\}$ is established for the vehicle, where $F_0 = M$. All the file sets are constantly updated by adding new files. For $\mathbb{F}^j, j = 0, 1, ..., W$, the corresponding file request probability is $\mathbb{P}^j = \{P_{j,1}, ..., P_{j,F_j}\}$, which follows Zipf-like distribution [27]. The request probability of the k-th popular content file can be calculated as

$$P_{j,k} = Pr_j \cdot \frac{\frac{1}{k^s}}{\sum_{n=1}^{F_j} \frac{1}{n^s}}, \quad (9)$$

where Pr_j is the probability of vehicles sending the request within RSU W_j if $j = 1, ..., W$ or out of RSU if $j = 0$, F_j is the number of files, k is the rank of popularity, s is the parameter characterizing the skewness of the Zipf distribution.

2) Fountain Coded Caching: Due to the limited transmission range of each edge caching server, it is difficult for the vehicles to download the whole file within the coverage of one single server, especially when the vehicles are with high speed or the size of file is large. By employing coded caching, which divides one content into separated coded packets, it has a higher probability to successfully deliver a coded packet with smaller size within one contact duration between the vehicles and the edge servers. In our coded caching scheme, through random linear fountain coding [16], each content is encoded into independent packets with a size of l bits, which is fixed and equivalent for all contents. If a content has a size of Kl bits, it can be recovered from any set of K' encoded packets, where $K' = K \cdot \sum_{d=1}^{K} z(d)$ is no less than K, and $z(d)$ can be calculated as follows

$$z(d) = \begin{cases} \frac{1}{K} + \frac{S}{K \cdot d} & d = 1 \\ \frac{1}{d(d-1)} + \frac{S}{K \cdot d} & d = 2, 3, ..., (K/S) - 1 \\ \frac{1}{d(d-1)} + \frac{S}{K} \ln\left(\frac{S}{\delta}\right) & d = K/S \\ \frac{1}{d(d-1)} & d = (K/S) + 1, ..., K \end{cases} \quad (10)$$

where $S = c \cdot \ln(K/\delta) \cdot \sqrt{K}$, and δ is the bound on decoding failure probability after receiving K' packets. We set $c = 0.2$ and $\delta = 0.05$ for the fountain code scheme [16].

IV. Delay and Cost Analysis for Cooperative Edge Caching

In this section, we perform the theoretical analysis of content delivery delay and service cost under the cooperative caching scheme. Particularly, we first elaborate on the workflow of cooperative content caching and content delivery. Then, we present the analysis of both location-based and popular contents, considering the differential delivery requirements.

A. Cooperative Content Caching

Denoted by $\mathbb{F} = \{1, 2, ..., f, ..., F\}$, the ground content set consists of $\mathbb{LF}^w$ and $\mathbb{PF}$. The size of $\mathbb{F}$ is given by $F = M + \sum_{w=1}^{W} N_w$. The data size of each file is represented by the required number of encoded packets for data recovery, $\mathbb{S} = \{S_1, S_2, ..., S_f, ..., S_F\}$. The caching capacities for each MBS, RSU, and vehicle, S_{MBS}, S_{RSU}, S_{VU}, are divided by packet size (l bits) to make $packet$ as the unit. Similarly, the size of admission control buffer, $\mathcal{T}_R$, for RSU is modified to packets.

Let the MBS and RSU W_w precache $s_B^f (\leq S_f)$ and $s_{R_w}^f (\leq S_f)$ independent encoded packets of file f, respectively. Considering the limited caching capacities of the MBS and RSUs, the total number of packets cached by each server

has to satisfy the constraint, i.e., $\sum_{f=1}^{F} s_B^f \leq S_{MBS}$ and $\sum_{f=1}^{F} s_{R_w}^f \leq S_{RSU}$, $w = 1, 2, ..., W$. In addition to capacity overhead, caching content also leads to a management cost. We define the price of caching one packet at MBS as CP_B, and CP_R for the RSU.

B. Content Delivery

If a target vehicle sends a content request, the request will be processed by the controller. Based on content placement and network access states, the controller makes the decision on association and content downloading for the vehicle. Then, the vehicle fetches the content following the instruction, including how many packets should be downloaded from other vehicles or edge caching servers.

PF content request can be raised by the vehicle at any locations within the coverage area of MBS. However, due to the dependency between location-based popularity of LF content and RSU coverage, the LF request can be raised by the vehicles entering the coverage of RSU W_w. The request is processed by the controller with the following steps:

1) *Availability of V2V transmission* – Check whether it is possible to transmit the content through V2V connections. Find the nearest vehicle holding the requested content and calculate the successful V2V transmission probability based on (8).
2) *Availability of edge cached content* – Obtain the list of MBS or RSUs that simultaneously cache the requested content and are available to the vehicle. Based on the moving speed and the duration of request, the edge caching servers (MBS and/or RSUs) that the vehicle will drive through can be determined. For an RSU, if serving the newly requested content makes its admission buffer overflow, the RSU should not be included in the list.
3) *Access to the remote server* – Make the decision on whether the target vehicle needs to fetch content from the remote server. Based on the total available cached content in the list, if the vehicle does not get sufficient packets for decoding by the transmission deadline, it will download the content from the remote server regardless of which connection is being utilized currently.

Next, we analyze the average download delay and cost for the request, which are taken as performance metrics for content placement scheme. Note that, content delivery through V2V connection is not considered when designing the content placement scheme, because V2V transmission performance is not affected by the content placement.

We assume contents held in the vehicles follow the file popularity distribution. If the content is available from neighboring vehicles, the successful V2V transmission probability is given by the lower bound in (8). Accordingly, the probability that the vehicle downloads file f through V2I communication (P_{V2I}^f) can be calculated. If the lower bound is higher than a threshold ξ, the vehicle is arranged to download from the other vehicles, $P_{V2I}^f = 1 - Pr\{\mathcal{A}_{ji} \geq S_f\}$. Otherwise, the target vehicle needs to fetch content through V2I connections, i.e., $P_{V2I}^f = 1$.

C. Delay Analysis of Content Delivery

For a vehicle served by V2I connections, the transmission process of file f can be divided into several segments depending on the handover of data providers. N_t^f denotes the number of segments that the vehicle can get connected, which is determined by the size and delay requirement of file f, and the list of edge servers caching the file f. In addition, the duration of each segment is defined as $T_n^f, n = 1, 2, ..., N_t^f$ and the downloaded data volume in packets during T_n^f is denoted as $S_n^f, n = 1, 2, ..., N_t^f$.

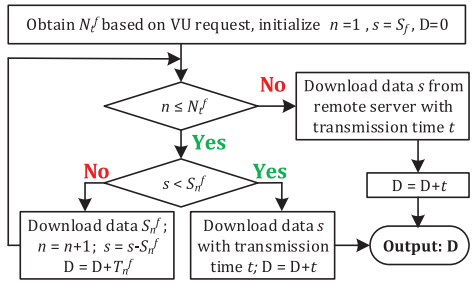


Fig. 2. Flowchart of downloading content through V2I connection.

Based on the analysis of V2R communication, a vehicle can download the requested content from RSU W_w within the coverage, once it successfully accesses to the RSU. Thus, if the vehicle accesses to RSU W_w during the n-th segment, the amount of downloaded data of the n-th segment is $S_n^f = s_{R_w}^f$ and the transmission delay for each packet can be defined as $t_{R_w}^f = T_n^f / s_{R_w}^f$. If the vehicle accesses to MBS during the n-th segment, the transmission delay for each packet can be defined as $t_B^f = l/\overline{R_B^L}$, and the downloaded data volume $S_n^f = \left\lfloor T_n^f / t_B^f \right\rfloor$, where $\lfloor \cdot \rfloor$ is the floor function.

If the remaining required data, s (packets), is less than S_n^f, which means the transmission will terminate during segment T_n^f, the transmission delay for remaining s can be calculated as $s \cdot t_B^f$ or $s \cdot t_{R_w}^f$, depending on the n-th vehicle access server. Otherwise, if $s > 0$ after the vehicle goes through all N_t^f segments, the vehicle will download remaining data from the remote server, and the transmission delay is $T_{n+1}^f = s \cdot t_{BL}^f$, where $t_{BL}^f = l/R_W$ is the backhaul link transmission delay for each packet. The delay D can be derived by the process shown in Fig. 2. Considering the coverage ranges of RSU and MBS [23], we deploy two RSUs along the highway as an example, but the analysis method and the content placement scheme design can be extended to scenarios with more RSUs.

For LF content downloading, the vehicle is expected to fetch the content as soon as possible, so we evaluate the average download delay and design the caching scheme to minimize the delay. In order to evaluate the delay performance, we calculate the mean of total content download delay for LF files ($\overline{D}$), the details are given in Appendix A. However, for PF content downloading, the vehicle always prefers to download the content within a latency requirement. We define three caching modes for PF content: *Mode 1* - only caching at MBS, *Mode 2* - caching at both MBS and RSUs, and *Mode 3* - no packet cached at MBS or RSUs. In Appendix B, for each mode, we evaluate the volume of downloaded data before the deadline of data delivery, and a content placement scheme is designed to ensure it is sufficient for data recovery.

D. Cost Analysis of Content Delivery

The cost of content service can be divided into two parts, the cost of caching the content and the cost of transmitting the data to vehicles.

In terms of caching cost, the prices for MBS, RSU W_w caching one packet are denoted as CP_B, CP_R, and $CP_B > CP_R$. Thus, the caching cost for file f, C_C^f, is given by

$$C_C^f = CP_B \cdot s_B^f + CP_R \cdot \sum_{w=1}^{W} s_{R_w}^f. \tag{11}$$

The total caching cost is denoted as $C_C = \sum_{f=1}^{F} C_C^f$, where F is the total number of files.

In terms of data transmission cost, the prices for MBS, RSU W_w and the remote server transmitting one packet are denoted as TP_B, TP_{R_w}, TP_{BL}, and $TP_{BL} > TP_B > TP_{R_w}$. In order to calculate the total transmission cost, numbers of packets downloaded by vehicle through these three methods need to be evaluated, which are denoted as $TN_B^f, TN_{R_w}^f, TN_{BL}^f, f = 1, 2, ..., F$. In addition, the number of vehicles requesting file f is $N \cdot P_f$, where P_f is the probability of file f requested by vehicle.

For the vehicle requesting file $f \in \mathbb{LF}^w$, it firstly downloads the cached packets, then from the remote server if necessary, thus we have $TN_B^f = s_B^f, TN_{R_w}^f = s_{R_w}^f, TN_{BL}^f = S_f - s_B^f - \sum_{w=1}^{W} s_{R_w}^f, f \in \mathbb{LF}^w$.

For vehicle requesting file $f \in \mathbb{PF}$, we first evaluate the average download duration of each method, $\overline{H_i^f}, i \in \{B, R_w, BL\}, w = 1, 2, \ldots, W$, which can be determined by D_R^f and handover duration sequences. Then, we can obtain its average number of downloading packets, $TN_i^f = \min(\lfloor \overline{H_i^f}/t_i^f \rfloor, s_i^f), i \in \{B, R_w, BL\}, w = 1, 2, \ldots, W$.

Thus, the average transmission cost for file f, C_T^f, is given by

$$C_T^f = N \cdot P_f \cdot \sum_{i \in \{B, R_w, BL\}} TP_i \cdot TN_i^f. \tag{12}$$

Then, the total transmission cost, C_T, is $C_T = \sum_{f=1}^{F} C_T^f$, where F is the total number of files.

Based on (11) and (12), the service cost, including both the caching and transmission cost, of all the files can be obtained. To reduce the total service cost, popular contents need to be cached at and transmitted from low-cost servers. In what follows, we achieve a low-cost content placement scheme via solving an optimization problem.

V. Cooperative Content Placement Problem

A. Multi-Objective Cooperative Content Placement Problem

For each file, its content placement is denoted as $\mathbf{s}^f = (s_B^f, s_{R_w}^f), w = 1, 2, ..., W$. With the objective of jointly minimizing service cost and delay requirement of content service, the cooperative content placement problem can be formulated as $\mathbf{P1}$. Since the objective of popular content service is downloading before the deadline, mode-based content placement schemes are designed for popular contents based on the delay analysis in Section IV-C. In $\mathbf{P1}$, the solution set for each popular content consists of its mode-based placements, which guarantee the delay requirements. For location-based contents, to achieve minimal downloading delay, we consider $\overline{D}$ obtained from delay analysis in the objective function. Meanwhile, the service costs for both types of contents are considered, and jointly minimized with the delay for location-based contents. In $\mathbf{P1}$, (13a) and (13b) are set to avoid redundant content caching for the MBS and RSUs, (13c) and (13d) reflect the caching capacity constraints of the MBS and RSUs, respectively, while (13e) is based on the admission capacity discussed in (7).

$$(\mathbf{P1}): \min_{\{\mathbf{s}^f\}} \left(\overline{D}, C_C + C_T\right) \tag{13}$$

$$\text{s.t.} \begin{cases} s_B^f \le S_f, f = 1, 2, ..., F & \text{(13a)} \\ s_{R_w}^f \le S_f, w = 1, ..., W, f = 1, 2, ..., F & \text{(13b)} \\ \sum_{f=1}^{F} s_B^f \le S_{MBS}, & \text{(13c)} \\ \sum_{f=1}^{F} s_{R_w}^f \le S_{RSU}, w = 1, ..., W & \text{(13d)} \\ \sum_{f=1}^{F} N \cdot P_f \cdot TN_{R_w}^f \le \mathcal{T}_R, w = 1, ..., W & \text{(13e)} \end{cases}$$

In order to solve the problem, content placement design should consider the tradeoff between content diversity, service cost, and download delay. For the RSUs, if more packets for one file ($s_{R_w}^f$) are cached, the vehicle can download each packet with lower delay ($t_{R_w}^f$). Thus, larger $s_{R_w}^f$ contributes to faster download rate for the vehicle, but it reduces the content diversity of the RSU, resulting in a low cache hit rate. For the MBS, it guarantees a high hit rate by providing a large access coverage whereby vehicles can fetch the content anywhere, but the transmission cost of the MBS is higher than that of the RSU.

In addition, the intermittent connection during transmission should be considered. For RSU transmission, the caching resource would be wasted if excessive packets of one file are cached but only part of these packets can be downloaded for vehicles within the RSU's coverage.

To achieve the objective of PF and LF content delivery services, we design content placement for these two types of content in different ways. For LF content, the objective is a joint minimization of delay and service cost. To minimize the transmission cost, the LF file $f \in \mathbb{LF}^w, w = 1, 2, ..., W$ should be cached at RSU W_w, because RSU has a lower transmission cost price than the MBS. However, the transmission delay of RSU may be larger than that of the MBS for files with small S_f. For PF content, the objective is to minimize the service cost within the download latency constraint. Based on the analysis in Section IV, given a delay constraint, the required caching data placement is deterministic for each case. Then, the service cost can be determined accordingly. Thus, the content placement for PF content is simplified to a selection of caching mode.

Although the content placement principle is different for PF and LF content, they share both the caching capacity and access resources. Thus, a joint design of PF and LF content placement is a requisite. Another challenge of this problem is the cooperation, as the MBS and RSUs may cooperatively cache different packets of one file. Therefore, the placement problem has an unacceptable solution set, which causes the curse of dimensionality.

1) LF Content Cooperative Placement Subproblem: The objective of LF content caching is to jointly minimize the transmission delay and service cost, considering the limited caching capacity of both MBS and RSUs, and admission limitation of RSUs. For $f \in \mathbb{LF}^w, w = 1, 2, ..., W$, since the delay, service cost, required caching capacity, and access resources are determined by the content placement $(s_B^f, s_{R_w}^f), w = 1, 2, ..., W$, we build a matrix for each file to record the delay, service cost, and required access resources. Note that, for $f \in \mathbb{LF}^w, w = 2, ..., W$, content transmission start under RSU W_w, so $s_{R_w}^f = 0, w = 1, ..., w-1$. To reduce the size of this matrix, the following principles are applied to build the matrix elements:

1) Avoid redundant content caching, $s_B^f + \sum_{w=1}^{W} s_{R_w}^f \le S_f$;

2) Satisfy the caching capacity constraint, $s^f_{R_w} \leq S_{RSU}$ and $s^f_B \leq S_{MBS}$;
3) Satisfy the admission capacity, $N \cdot P_f \cdot s^f_{R_w} \leq \mathcal{T}_R$.

Based on the principles, we get the matrix for file $f \in \mathbb{LF}^w$, in which each column represents a possible placement scheme. We use N^f_C to denote the number of possible content placement schemes for file f. Then, we calculate the average delay, service cost, and access resources for each scheme. Given a placement scheme $(s^f_B(n), s^f_{R_w}(n)), w = 1, 2, ..., W, n = 1, 2, ..., N^f_C$, the corresponding average delay $\overline{D}^f(n)$ is calculated according to Section IV-C. The average service cost $\overline{C}^f(n) = C^f_C + C^f_T$, where C^f_C and C^f_T are discussed in Section IV-D. The required access resources $T^f_{R_w}(n) = N \cdot P_f \cdot TN^f_{R_w}, w = 1, 2, ..., W$. In addition, required caching resources can be determined by the content placement scheme. This matrix will be utilized as input information in the cooperative content placement algorithm.

2) PF Content Cooperative Placement Subproblem: Different from LF content, the objective of caching PF content is minimizing the service cost with guaranteed delay. Based on previous discussions, the content placement for PF content is simplified to a selection of caching modes. We get the matrix for file $f \in \mathbb{PF}$, in which each column corresponds to one caching mode. Firstly, for each mode n, based on latency requirement, we determine its content placement scheme $(s^f_B(n), s^f_{R_w}(n)), w = 1, 2, ..., W, n = 1, 2, 3$ $(N^f_C = 3)$. Then, the average service cost, $\overline{C}^f(n)$, and the access resources, $T^f_{R_w}(n)$, for each mode are calculated. In addition, $\overline{D}^f(n)$ is set to 0 in accordance with the LF matrix.

B. Multi-Objective MMKP Formulation for Cooperative Content Placement

The cooperative content placement problem $\mathbf{P1}$ can be transferred to a multi-objective MMKP, as shown in $\mathbf{P2}$, which is a variant of the knapsack problem. There are F groups of items, and the f-th group includes N^f_C items.

The objective function requires the joint minimization of delay and service cost, where $x_{f,n}$ is a binary variable representing the designed content placement scheme for file f, and $x_{f,n} = 1$ if scheme n is selected, and $x_{f,n} = 0$ otherwise. Due to the limited resources, $2W + 1$ constraints are considered in $\mathbf{P2}$, including the caching capacity of MBS in (14b) and RSUs in (14c), and admission limitation for RSUs in (14d).

$$(\mathbf{P2}): \min_{\{x_{f,n}\}} \left(\sum_{f=1}^{F} \sum_{n=1}^{N^f_C} \overline{D}^f(n) \cdot x_{f,n}, \sum_{f=1}^{F} \sum_{n=1}^{N^f_C} \overline{C}^f(n) \cdot x_{f,n} \right) \tag{14}$$

$$\text{s.t.} \begin{cases} \sum_{n=1}^{N^f_C} x_{f,n} = 1, x_{f,n} \in \{0,1\}, f = 1, 2, ..., F; & (14a) \\ \sum_{f=1}^{F} \sum_{n=1}^{N^f_C} s^f_B(n) \cdot x_{f,n} \leq S_{MBS}; & (14b) \\ \sum_{f=1}^{F} \sum_{n=1}^{N^f_C} s^f_{R_w}(n) \cdot x_{f,n} \leq S_{RSU}, w = 1, ..., W; & (14c) \\ \sum_{f=1}^{F} \sum_{n=1}^{N^f_C} T^f_{R_w}(n) \cdot x_{f,n} \leq \mathcal{T}_R, w = 1, ..., W. & (14d) \end{cases}$$

Dynamic programming (DP) is a widely used method to solve the knapsack problem. However, DP is inefficient for large scale problems due to the complex constraint calculation and considerable state storage requirements, which is known as the curse of dimensionality. In what follows, we propose a content placement algorithm based on ACO to find near-optimal solutions to the multi-objective MMKP. Furthermore, to evaluate the gap between this near-optimal solution and the optimum, we obtain a lower bound of the objective value by relaxing $\mathbf{P2}$ to a linear programming (LP) problem.

VI. ACO Based Algorithm Design

The ACO was first proposed as an approximate method for solving complex optimization problems, inspired by how ant colonies find the path from food source to the nest [28]. At the beginning stage of foraging, the ants explore paths randomly and leave pheromone when they move on the ground. The quantity of pheromone is inversely proportional to the length of the path, which means a shorter path has more pheromone. Then, the ants can choose a path according to the pheromone, and they always prefer the path with stronger pheromone. This is how ants exchange information with each other through pheromone, and find the shortest path cooperatively. Note that, heuristic information, such as the potential gain of choosing a certain step along the path, will be also used by the ants in addition to pheromone. The problem formulated in $\mathbf{P2}$ is a multi-objective minimization problem, which can be solved by multi-objective evolutionary algorithms (MOEAs). The concept of dominance is widely used in MOEAs through establishing a non-dominated solution. We will first introduce the concept of dominance and non-dominated solutions, then present the dominance-based ACO algorithm for cooperative content placement problem.

A. Non-Dominated Solution

Consider a multi-objective minimization problem with n objectives ($\mathbf{g} = \{g_1, g_2, ..., g_n\}$) and m decision variables ($\mathbf{x} = \{x_1, x_2, ..., x_m\}$). Thus, the solution can be denoted by $\mathbf{x}$ and its corresponding objective vector is $\mathbf{g}(\mathbf{x})$. Based on the definition in [29], if $\mathbf{x}_1$ is not worse than $\mathbf{x}_2$ in any objective and strictly better than $\mathbf{x}_2$ in at least one objective, the solution $\mathbf{x}_1$ is defined to dominate $\mathbf{x}_2$. If a solution is not dominated by any other solutions, it is defined as a non-dominated solution. The corresponding objective points of all non-dominated solutions form a front in the objective space, which is the Pareto optimal front [30]. Thus, the multi-objective minimization problem can be solved by finding the non-dominated solution set, which is also the set of Pareto optimal solutions.

B. Dominance-Based ACO Algorithm

In order to find the solution to $\mathbf{P2}$, we propose a dominance-based ACO algorithm, which optimizes multiple objectives by combining dominance with the ACO. Multiple objectives and dominance are incorporated in the following phases:

1) *Pheromone update:* The pheromone is updated every iteration, including general pheromone evaporation and additional pheromone that is incrementally deposited by the selected solutions from the non-dominated set;
2) *Definition of pheromone and heuristic information:* Pheromone and heuristic information are used by ants to make probabilistic decisions at each step. Due to the

multiple objectives, pheromone and heuristic information can be stored in multiple matrices, each of which corresponds to one objective. Since the ant has to aggregate the pheromone/heuristic matrices when making the decisions, we calculate a weighted sum of multiple matrices to aggregate multiple objectives.

C. Dominance-Based ACO Content Placement Algorithm

Consider an ant colony with X_A ants, each ant has the capability of constructing a feasible content placement scheme. After all X_A ants constructing their schemes, the non-dominated content placement scheme set, $\mathbf{ND}$, can be updated, in which the delay and cost for each newly established scheme are compared with the schemes in current $\mathbf{ND}$ to determine the updated non-dominated scheme. Then, based on $\mathbf{ND}$, we can update the pheromone matrix to simulate the evaporation and accumulation of pheromone, and the updated pheromone will be used during the next iteration. The proposed algorithm terminates after X_I iterations.

A set of pheromone vectors $[\tau_{f,1}, \tau_{f,2}, ..., \tau_{f,N_C^f}], f = 1, 2, .., F$ are built at the initialization stage, and each element is set to be $\tau_{\max}$, which is the upper bound of pheromone value. At the end of each iteration, the pheromone vectors are updated. First, in order to simulate the pheromone loss caused by evaporation, all elements are decreased by multiplying $(1-\rho)$, where $\rho \in [0, 1]$. Then, based on the updated non-dominated scheme set, all the pheromone values of elements (f, n) (selected by the scheme $\mathbf{x}_i, \mathbf{x}_i \in \mathbf{ND}$) are increased by multiplying $(1+\gamma_i)$. γ_i describes how good the performance of $\mathbf{x}_i$, i.e.,

$$\gamma_i = \frac{\mathcal{F}(\mathbf{x}_i)}{\sum_{\mathbf{x}_j \in \mathbf{ND}} \mathcal{F}(\mathbf{x}_j)}, \tag{15}$$

where $\mathcal{F}(\mathbf{x}_i) = 1/(W_D \cdot \overline{D}(\mathbf{x}_i) + W_C \cdot \overline{C}(\mathbf{x}_i))$ is the performance evaluation function of the scheme (i.e., the reciprocal of weighted summation of delay and service cost after applying the scheme $\mathbf{x}_i$), and W_D and W_C are the weights for delay and service cost, respectively.

Therefore, the update of pheromone follows

$$\tau_{f,n}(t+1) = \tau_{f,n}(t) \cdot (1-\rho) \cdot \prod_{\mathbf{x}_i \in \mathbf{ND}_{f,n}} (1+\gamma_i), \tag{16}$$

where $\tau_{f,n}(t)$ is the pheromone value in the t-th iteration, and $\mathbf{ND}_{f,n} \subseteq \mathbf{ND}$ is the subset of non-dominated schemes that contains the choice of (f, n). Thus, the pheromone value $\tau_{f,n}$ represents how good the performance achieved by the element (f, n) in previous iterations, which can be used to guide the selection for the next iteration.

In order to balance the exploitation and exploration (i.e., selecting the known good-performance scheme and choosing the under-explored schemes), we set the upper and lower bound of pheromone value as $\tau_{\max}$ and $\tau_{\min}$, respectively. This can avoid the ACO algorithm entering a stagnation situation by bounding the level of exploration. A pseudo-random proportional rule is also used in content placement construction for balancing exploration and exploitation. With the exploitation probability q_0, the ant selects the content placement scheme with the best potential performance; otherwise, the decision is made probabilistically, where the probability of a scheme being selected is proportional to its potential performance.

In terms of heuristic information, we combine the two objectives (delay and service cost) together as a weighted sum. In addition to the objective functions, heuristic information $\eta_{f,n}$ (representing the preference of mode n is chosen by the file f) also depends on the resources (including caching and admission) consumed by this choice. The ratio between the consumed resources and the remaining resources is defined as $RC_{f,n}$, which represents the tightness of selecting mode n on resource constraints.

$$RC_{f,n} = \frac{S_B^f(n)}{RS_{MBS}} + \sum_{w=1}^{W} \left(\frac{S_{R_w}^f(n)}{RS_{R_w}} + \frac{T_{R_w}^f(n)}{RT_{R_w}} \right). \tag{17}$$

Then, $\eta_{f,n}$ is calculated as

$$\eta_{f,n} = 1/\left[\left(W_D \cdot \overline{D}^f(n) + W_C \cdot \overline{C}^f(n) \right) \cdot RC_{f,n} \right]. \tag{18}$$

From (18), if the mode achieves better performance (i.e., lower weighted summation of delay and cost) and consumes less resource (i.e., lower $RC_{f,n}$), then $\eta_{f,n}$ will be higher.

Probabilistic decisions are made by ants to construct a content placement scheme. For file f, the ant first finds out its feasible mode set $\mathcal{O}_f$, which are determined by resource constraints, (i.e., the remaining caching and admission resources are sufficient for mode $n \in \mathcal{O}_f$). After that, the probability of selecting feasible mode n is

$$p_{f,n} = \frac{\tau_{f,n}^{\alpha} \cdot \eta_{f,n}^{\beta}}{\sum_{l \in \mathcal{O}_f} \tau_{f,l}^{\alpha} \cdot \eta_{f,l}^{\beta}}, \tag{19}$$

where $\tau_{f,n}$ and $\eta_{f,n}$ are the updated pheromone and heuristic information, respectively, and α and β determine the impact of pheromone and heuristic information on the decision.

It has been proved that, when applied to combinatorial optimization problems, ACO algorithms can achieve the best performance in conjunction with random local search method [31]. Starting from an initial solution, local search method tries to find an optimal solution within the predefined neighborhood of the initial point. Since ACO algorithms perform a rather coarse-grained search for the optimal solution, quality of the solutions can be enhanced with the aid of local search. The initial solution is the content placement scheme produced by each ACO iteration, then it can be locally optimized through local search. Since the pheromone is utilized by following ACO iterations, its update based on the locally optimized solutions will have a long-term impact on the performance.

The proposed ACO algorithm for solving $\mathbf{P2}$ is described in Algorithm 1, and the ant constructs content placement scheme following Algorithm 2.

VII. Performance Evaluation

In this section, we first evaluate the convergence performance of the proposed algorithm. Then, we compare the caching performance, including the average delay and the service cost, of our algorithm with the benchmark methods.

A. Simulation Setup

In our simulation scenario, 2 RSUs are deployed in the coverage of one MBS along the highway. For V2R communication, the setting of RSU follows [23], and the transmission rates for each zone are shown in Table II. The configurations of V2B and V2V communication follow the 3GPP standard and [22], respectively. The detailed parameters of V2B communication and vehicle mobility are given in Table III.

Algorithm 1: ACO-based Content Placement

Input: Parameters of the system model, including vehicle mobility model, V2I and V2V communication models;
Output: Content placement decisions $\mathbf{x} = \{x_{f,n}\}$
Initialization:
Objectives: delay vectors $\left\{\overline{D}^f\right\}$; cost vectors $\left\{\overline{C}^f\right\}$
Required resources: caching $S^f_{R_w}$, S^f_B; admission $T^f_{R_w}$
$\mathbf{ND} = \{\}$; Pheromone matrix $\tau = \tau_{\max}$
for $i = 1 : X_I$ **do**
  for $k = 1 : X_A$ **do**
    Ant k constructs scheme $\mathbf{x}_k$ (following Algorithm 2)
  Update non-dominated content placement scheme in $\mathbf{ND}$
  Update pheromone value following (16)
  if *Pheromone value* $< \tau_{\min}$ *or* $> \tau_{\max}$ **then**
    Set pheromone value to $\tau_{\min}$ or $\tau_{\max}$
Evaluate performance ($\mathcal{F}(\mathbf{x})$ in (15)) for all schemes in $\mathbf{ND}$, and find the scheme $\mathbf{x}$ with the best performance

Algorithm 2: Content Placement Scheme Construction

Initialize remaining caching and admission resources to S_{MBS}, S_{RSU}, and $\mathcal{T}_R$; and randomly order the files
for $f = 1 : F$ **do**
  for $n = 1 : N^f_C$ **do**
    if *Remaining resources are sufficient for* (f, n) **then**
      Obtain selecting probability following (19);
  if $q < q_0$, *where* $q \sim U(0, 1)$ **then**
    Exploitation: mode with the highest probability
    else
      Exploration: probabilistic mode selection
  Update remaining resources
Update $\mathbf{x}_k$ by $Local\ search(\mathbf{x}_k)$ (optional)

TABLE III
PARAMETERS OF SYSTEM MODEL

V2B	Carrier Freq.	Bandwidth	Tx Power	Coverage
	2000 MHz	20 MHz	43 dBm	1000 m
Mobility	$V_{\min}$	$V_{\max}$	a	λ
	20 m/s	30 m/s	2 m/s^2	1/3 sec^{-1}

The files requested by the vehicles are classified as three sets, i.e., PF and $LF_w, w = 1, 2$. We consider the same data size for files belonging to one set and their packets are encoded by fountain code. To analyze the impact of different file size and caching capacity, we conduct simulation under several cases, as shown in Table IV, both file size and caching capacity are in the unit of packet and s is the parameter characterizing the skewness of Zipf-like popularity distribution. These cases represent varying proportions of PF and LF files (by changing the size of LF files) at different levels of caching capacity.

TABLE IV
PARAMETERS OF FILES AND CACHING CAPACITIES

	Caching capacity		PF		LF		s
	MBS	RSU	Size	Num.	Size	Num.	
Case1	50	30	5	10	14	5	0.7
Case2	100	40	5	10	22	5	1
Case3	50	40	5	10	18	6	1

To evaluate the service cost of caching and transmission, we set the price of caching at MBS and RSU to be 0.3 and 0.5, respectively, and the price of transmission to be 0.8, 0.5, and 1.5, corresponding to the MBS, RSU, and backhaul link, respectively. In addition, we set the equal weights of delay and service cost in the objective function, i.e., $W_D = W_C = 0.5$. In our simulation results, we use cost to represent this weighted summation of delay and service cost.

We compare the proposed cooperative content placement scheme (denoted by coop) with two benchmark methods. In noncooperative caching scheme (denoted by noncoop), there is no cooperation between MBS and RSUs, which means a file can be cached at either MBS or RSUs. In greedy caching scheme (denoted by greedy), PF and LF files are greedily cached at MBS and corresponding RSUs based on file popularity, respectively. Then, we compare the caching performance of our algorithm with the lower bound achieved by the LP method (denoted by LP). Comparing $\mathbf{P2}$ and its linear relaxation ($\mathbf{P3}$), we can see that the feasible solution set of $\mathbf{P2}$ is a subset of the feasible solution set of $\mathbf{P3}$ [32]. Thus, the solution to $\mathbf{P3}$ can be used as the lower bound for the solution to $\mathbf{P2}$.

Linear relaxation can be achieved by replacing (14a) with (20a), shown at the bottom of this page. Multiple constraints are combined in (20b), shown at the bottom of this page, where ω_B, ω_{RS}, and ω_{RT} are surrogate multipliers for MBS caching, RSU caching, and RSU access resources, respectively, which are positive real numbers. These multipliers can be obtained through the method proposed in [32], and they can be seen as the shadow price of its related constraint in the relaxation.

B. Simulation Results

We first illustrate the convergence of the proposed method. Then, we evaluate the performance of the proposed algorithm via both numerical results and Monte Carlo simulation results. Furthermore, the impacts of caching capacity resource allocation, weight parameters, and file set on the performance are analyzed. Caching performance is evaluated by the value of the objective function in $\mathbf{P3}$, a lower value reflects a better performance.

1) Convergence Performance: For the ACO-based cooperative content placement algorithm design, the number of ants has an impact on the convergence. We can observe from Fig. 3 that how the convergence speed changes with the number of ants, i.e., 10, 30, and 50. Both ACO with and without local

$$(\mathbf{P3}): \min_{\{x_{f,n}\}} \left(W_D \cdot \sum_{f=1}^{F} \sum_{n=1}^{N^f_C} \overline{D}^f(n) \cdot x_{f,n} + W_C \cdot \sum_{f=1}^{F} \sum_{n=1}^{N^f_C} \overline{C}^f(n) \cdot x_{f,n} \right) \tag{20}$$

$$\text{s.t.} \begin{cases} \sum_{n=1}^{N^f_C} x_{f,n} = 1, x_{f,n_c} \in [0,1], f = 1, 2, ..., F; & \text{(20a)} \\ \sum_{f=1}^{F} \sum_{n=1}^{N^f_C} \left[\omega_B \cdot s^f_B(n) + \sum_{w=1}^{W} \left(\omega_{RS} \cdot s^f_{R_w}(n) + \omega_{RT} \cdot T^f_{R_w}(n) \right) \right] x_{f,n} \leq \omega_B S_{MBS} + 2\omega_{RS} S_{RSU} + 2\omega_{RT} \mathcal{T}_R & \text{(20b)} \end{cases}$$

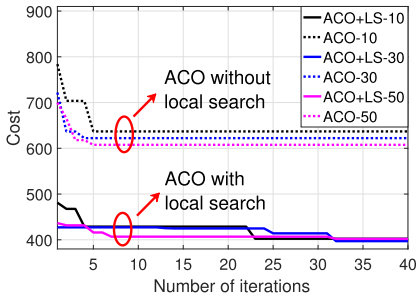


Fig. 3. Convergence speed.

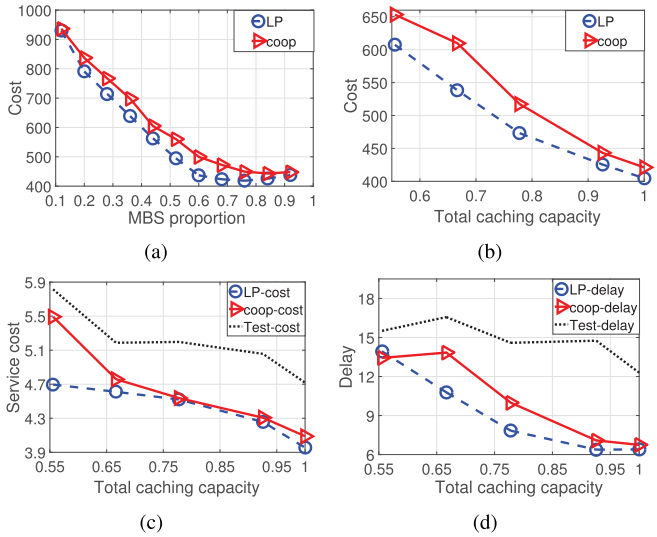


Fig. 4. Performance changing with caching capacity: (a) MBS caching; (b) total caching; (c) service cost performance; (d) delay performance.

search are simulated with file assumption following Case 1 in Table IV. Since the convergence of ACO is mainly dependent on the number of solutions generated, i.e., $number\,of\,ants * number\,of\,iterations$, more iterations are required if we use fewer ants, as shown in Fig. 3. In addition, we can also observe that the ACO algorithm with local search (denoted by ACO+LS) achieves a much better result than ACO without local search, due to the effectiveness of local optimization. In our following simulation, we set the number of ants to be 30 and run the ACO algorithm with local search for 35 iterations, which is sufficient to achieve the convergence.

2) Impact of Caching Capacity: The total caching capacity is defined as the ratio of total capacity, i.e., the summation of caching capacities at MBS and RSUs, to the total size of PF and LF files. To study how caching capacity affects the performance, we consider both the proportion of different edge caching servers' caching capacity and the amount of total caching capacity. We use the setting of Case 2 in Table IV.

In Fig. 4(a), the total capacity is fixed at 0.78, and the proportion of MBS varies from 0.05 to 0.9. With an increased proportion of MBS caching capacity, the performance keeps improving until the MBS proportion reaches over 0.7. Due to the limited admission resources of RSU, the bottleneck of RSU caching performance is admission resources when the MBS proportion is under 0.7 (i.e., RSU proportion is over 0.3). In this region, more caching resources should be allocated to the MBS to improve the caching resource utilization efficiency. However,

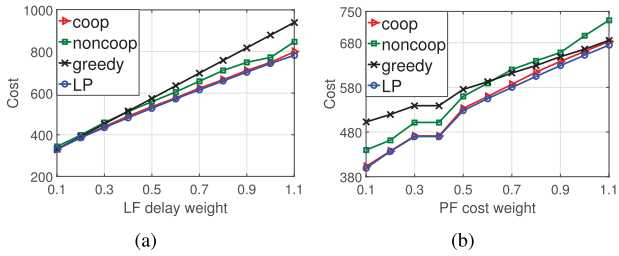


Fig. 5. Performance changing with weights: (a) delay weight; (b) cost weight.

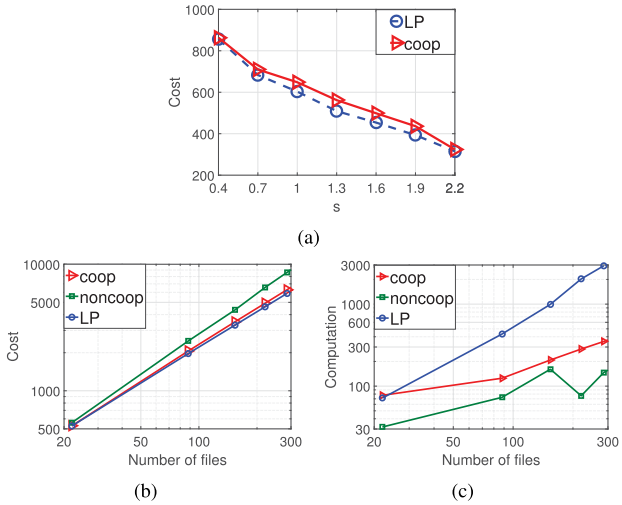


Fig. 6. Performance changing with file parameters: (a) popularity skewness s; (b) cost; (c) computation complexity.

if the MBS proportion is over 0.7, increasing the MBS proportion will cause a slight performance degradation, which is caused by the higher transmission cost of MBS compared with RSU. Thus, the optimal performance can always be achieved at the point where caching and admission resources for one RSU are matched.

We show the impact of total caching capacity in Fig. 4(b). For each point, the performance is the best value that can be achieved by the specific total capacity. When the total capacity grows from 0.56 to 1, the performance improves by 33%. The average service cost (i.e., $(C_C + C_T)$ divided by the data volume of transmitted PF and LF contents) and delay (i.e., $\overline{D}$ divided by the data volume of transmitted LF contents) for the vehicle downloading through V2I are evaluated in Fig. 4(c) and 4(d) respectively, where we compare the numerical results (coop, LP) and the Monte Carlo simulation results (Test). In both subfigures, the numerical results of the proposed algorithm are slightly higher than the lower bound from the LP method. With the increase of caching resources, more files can be precached at MBS and RSUs, hence less data is transmitted through backhaul link, saving both service cost and downloading time. In addition, the Monte Carlo simulation results show similar trends to the numerical results, so we can design the caching scheme according to the numerical analysis.

3) Impact of Weight Parameters: Since the total cost is a weighted sum of the service cost and delivery delay, the weights influence the preference between different files and caching servers. In Fig. 5(a), we compare how the performance of different methods change with delay weight, keeping $W_C = 0.5$.

We use the setting of files and caching capacity as Case 3 in Table IV. Without cooperative caching between MBS and RSUs for LF files, it leads to a higher cost than that of the cooperative caching method. Since the greedy method prefers to cache the PF files at MBS, it has the largest cost with high delay weight. This is because vehicles have to download the low-popularity LF files, that are not cached by both the RSUs and MBS, through the backhaul link. The service cost weight of PF file also affects the performance, as shown in Fig. 5(b). Since the greedy method has a fixed priority of caching, its performance is degraded when weight parameters change. However, the proposed method can adapt to variable weights, which keeps the small gap with performance lower bound obtained by the LP method.

4) Impact of File Parameters: In order to analyze the impact of file popularity on performance, in Fig. 6(a), we plot the performance as a function of parameter s of popularity distribution in (9), using the setting of files and caching capacity as Case 2 in Table IV. We observe that the performance of caching improves as s increases. This is because popularity distribution becomes more skewed towards higher popularity contents, and these contents are given more priority during the resource allocation. If the distribution becomes more skewed, the resources allocated to high popular contents will be utilized more efficiently, which can improve the performance. From Fig. 6(a), we also obtain that the performance achieved by the proposed ACO-based algorithm and the lower bound from the LP method is very close, indicating that the ACO-based algorithm can achieve the near-optimal solution.

To evaluate the scalability of the content placement algorithms, we design the content placement scheme for an increased number of files. On the basis of Case 3 in Table IV, we scale up the number of files and caching capacities, and compare different content placement algorithms in Fig. 6(b). Compared with the LP method, the proposed cooperative algorithm always achieves near-optimal performance, while the performance gap for the noncooperative method is increased with larger file set. This means the noncooperative method suffers performance degradation for large file sets. Without cooperation between the MBS and RSUs, there are less feasible content placement schemes for the noncooperative method, which leads to lower computation complexity than the cooperative method, as shown in Fig. 6(c). For the cooperative algorithm, due to the polynomial relationship between computation complexity and the number of files, its efficiency and scalability are verified.

VIII. Conclusion

In this article, based on a multi-tier H-VNet, we have proposed a cooperative caching scheme to improve content delivery services for connected vehicles. Considering the differential delivery requirements for location-based and popular contents, a cooperative content placement algorithm has been devised to reduce both the transmission delay and the service cost. We have evaluated the convergence speed of the proposed algorithm, and demonstrated that it can effectively improve the overall performance. Besides, the robustness of the algorithm has been validated under various parameter settings. For the future work, we will further investigate the inter-layer cooperation among terrestrial and aerial caching servers to further enhance the system performance in terms of coverage, reliability, and robustness.

Appendix A
Delay Analysis of LF Content Delivery

Since content of $\mathbb{LF}^w$ is requested right after the vehicle enters coverage of RSU W_w, the duration of vehicle staying in each RSU is determined by its average speed and the RSU coverage range, i.e., $T_{R_w} = D_R/E[v]$. According to Fig. 1, the time duration under the MBS can be divided into three segments, denoted by $T_{B_1} = D_{M_1}/E[v], T_{B_2} = D_{M_2}/E[v]$ and $T_{B_3} = D_{M_3}/E[v]$.

For $f \in \mathbb{LF}^1$, its content placement scheme is $[s^f_{R_1}, s^f_{R_2}, s^f_B]$. Without loss of generality, we classify the caching states into four types, based on the caching conditions of two RSUs (i.e. $s^f_{R_w} > 0$ or $s^f_{R_w} = 0, w = 1, 2$). The effective transmission time segments for MBS and RSU can be defined correspondingly. For instance, if $s^f_{R_1} > 0$ and $s^f_{R_2} > 0$, we have $T^f_1 = T_{R_1}, T^f_2 = T^1_B = T_{B_2}, T^f_3 = T_{R_2}, T^f_4 = T^2_B = T_{B_3}$, where T^n_B means the duration of VU accessing to MBS at the n-th time slot. The volume of downloaded data from MBS during T^n_B is denoted by $s^f_{B,n} = \left\lfloor T^n_B / t^f_B \right\rfloor$. For $f \in \mathbb{LF}^2$, since file only needs to be cached at RSU W_2 and MBS, there are two types of caching states (i.e. $s^f_{R_2} > 0$ or $s^f_{R_2} = 0$).

In addition to the connection duration and transmission rate, the number of packets downloaded during the n-th time segment, $S^f_n, n = 1, 2, ..., N^f_t$, is bounded by the content placement scheme, $[s^f_{R_1}, s^f_{R_2}, s^f_B]$. Based on the time segments sequence and the content placement, the average transmission delay of file f, $\overline{D}^f$, can be calculated following the content downloading process shown in Fig. 2. Here, we calculate the $\overline{D}^f$ in the case of caching file f, $f \in \mathbb{LF}^1$, at both RSUs as an example, shown as (21), shown at the bottom of this page.

$$\overline{D}^f = \begin{cases} S_f \cdot t^f_{R_1} & \text{if } s^f_{R_1} \geq S_f \\ T^f_1 + t^f_B \cdot \left(S_f - s^f_{R_1}\right) & \text{else if } s^f_{R_1} + \min\left(s^f_{B,1}, s^f_B\right) \geq S_f \\ T^f_1 + T^f_2 + t^f_{R_2} \cdot \left[S_f - s^f_{R_1} - \min\left(s^f_{B,1}, s^f_B\right)\right] & \text{else if } s^f_{R_1} + \min\left(s^f_{B,1}, s^f_B\right) + s^f_{R_2} \geq S_f \\ T^f_1 + T^f_2 + T^f_3 + & \text{else if } s^f_{R_1} + \min\left(s^f_{B,1} + s^f_{B,2}, s^f_B\right) + s^f_{R_2} \geq S_f \\ t^f_B \cdot \left(S_f - s^f_{R_1} - \min\left(s^f_{B,1}, s^f_B\right) - s^f_{R_2}\right) & \\ T^f_1 + T^f_2 + T^f_3 + t^f_B \cdot \max\left(s^f_B - s^f_{B,1}, 0\right) + & \text{else} \\ t^f_{BL} \cdot \left[S_f - s^f_{R_1} - s^f_{R_2} - \min\left(s^f_{B,1} + s^f_{B,2}, s^f_B\right)\right] & \end{cases} \tag{21}$$

TABLE V
HANDOVER DURATION SEGMENTS

Case	Caching types	Handover segments
1	$s^f_{R_1}=s^f_{R_2}=0$	$H^f_1=(T_{B_1}+T_{R_1}+T_{B_2}+T_{R_2}+T_{B_3})$
2	$s^f_{R_1}=0, s^f_{R_2}>0$	$H^f_1=(T_{B_1}+T_{R_1}+T_{B_2}), H^f_2=T_{R_2}, H^f_3=T_{B_3}$
3	$s^f_{R_1}>0, s^f_{R_2}=0$	$H^f_1=T_{B_1}, H^f_2=T_{R_1}, H^f_3=(T_{B_2}+T_{R_2}+T_{B_3})$
4	$s^f_{R_1}>0, s^f_{R_2}>0$	$H^f_1=T_{B_1}, H^f_2=T_{R_1}, H^f_3=T_{B_2}, H^f_4=T_{R_2}, H^f_5=T_{B_3}$

In order to evaluate delay performance, we calculate the mean of total content download delay for LF files ($\overline{D}$)

$$\overline{D}=\sum_{w=1}^{W}\left[\sum^{f\in\mathbb{LF}^w}\left(N\cdot P_f\cdot P^f_{V2I}\cdot\overline{D}^f\right)\right], \tag{22}$$

where $N\cdot P_f\cdot P^f_{V2I}$ is the average number of VUs fetching file f through V2I connection. Through optimizing $S^f_{R_1}$, $S^f_{R_2}$, and S^f_B, we aim to minimize $\overline{D}$ for LF services.

APPENDIX B
DELAY ANALYSIS OF PF CONTENT DELIVERY

We assume that the number of vehicles driving into the coverage follows Poisson process and PF contents are requested according to content popularity. Thus, the locations where vehicles raise the request and start the transmission are uniformly distributed within MBS coverage. We consider the file download latency requirement for file f as D^f_R, and evaluate the average data volume that can be downloaded during D^f_R as S^f_D.

The content placement scheme for file f, $f\in\mathbb{PF}$, is $[s^f_{R_1}, s^f_{R_2}, s^f_B]$. Without loss of generality, we classify the caching states into four types, the possible handover locations for vehicles can be demonstrated by a duration sequence for handover segments in Table V, the duration of each segment is defined as H^f_n. Different from LF, since the PF transmission may start at any location, the time segment is not fixed for each case. Thus, we calculate the average T^f_n for each case, considering latency constraint, $\sum_{n=1}^{N^f_t} T^f_n = D^f_R$, where N^f_t is the number of segments that the vehicle can go through. To guarantee successful file transmission, we should make sure S^f_D for all possible combinations of time segments are larger than S_f, so we need to calculate the minimum value of S^f_D. Based on the handover duration sequences in Table V and the content placement, the minimum downloaded data volume for file f within the delay constraint can be calculated for each case.

To evaluate the minimum downloaded data volume within D^f_R, we need to compare transmission rates between the MBS and RSU $W_w (w=1,2)$ to determine the worst case. To compare with RSU W_w, the average MBS transmission delay (t^f_{M,W_w}) of each packet can be calculated as the weighted average of t^f_B and t^f_{BL}

$$t^f_{M,W_w}=\frac{s^f_B\cdot t^f_B+\max\left(s^f_{R_w}-s^f_B,0\right)\cdot t^f_{BL}}{s^f_B+\max\left(s^f_{R_w}-s^f_B,0\right)}. \tag{23}$$

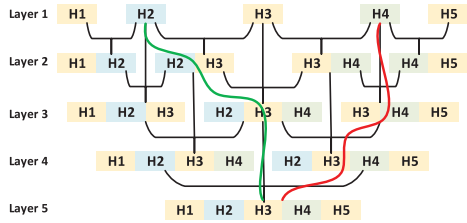


Fig. 7. Illustration of possible combinations of handover segments.

The minimum S^f_D corresponds to the download data volume of the worst case, which is defined as vehicle downloading from edge caching server with the lowest transmission rate, following by higher ones. We build a tree diagram to obtain the possible combinations of handover segments, with a given D^f_R. In Fig. 7, we set a child node as the extension of its parent nodes. Then, each path from a layer 1 node to the last layer node is a possible segment combining process to achieve a given D^f_R. The tree diagram with 5 layers illustrates the handover for Case 4. Since there are 3 segments, (H^f_1, H^f_2, H^f_3), for Case 2 and Case 3, a 3-layer tree diagram can be built accordingly. For instance, the green line in Fig. 7 represents a possible combination of handover segments for Case 4, when the ascending order of transmission time for one packet follows $W_1 > W_2 >$ MBS. The order indicates the worst case is downloading from RSU W_1, so the path starts from H^f_2. If $D^f_R > H^f_2$, the vehicle can drive out of the coverage of W_1 during the service, so it can extend to layer 2, i.e., $(H^f_2 - H^f_3)$. Since the transmission from RSU W_2 is more time consuming than MBS, the extension directs to H^f_4, i.e., $(H^f_2 - H^f_4)$ of layer 3. If $D^f_R > H^f_2 + H^f_3 + H^f_4$, we consider the combination of all handover segments, i.e., $(H^f_1 - H^f_5)$ of layer 5. We compare D^f_R with the length of each node in one path from layer 1 to the last layer, the first node longer than D^f_R is one possible combination for transmission segments, which determines the specific transmission pattern. All possible combinations can be found by searching paths in the diagram. Next, we need to calculate the downloaded data volume of all possible transmission patterns and find the required $[s^f_{R_1}, s^f_{R_2}, s^f_B]$ for successful transmission.

A. Case 1

If $s^f_B > 0$, a vehicle firstly downloads content from caching server at MBS, then from the remote server if necessary. Otherwise, the vehicle directly downloads from the remote server. If $\left\lfloor D^f_R/t^f_B \right\rfloor < S_f$, file f cannot be successfully downloaded for

this content placement case regardless of the value of s_B^f. Otherwise, file f may be successfully downloaded, which depends on s_B^f. Since $s_B^f \le S_f$, we have $D_R^f \ge s_B^f \cdot t_B^f$ as the necessary condition for successful transmissions.

$$S_D^f = s_B^f + \left\lfloor \left(D_R^f - s_B^f \cdot t_B^f \right) / t_{BL}^f \right\rfloor. \tag{24}$$

In order to guarantee successful transmission, we let $S_D^f \ge S_f$, then we can obtain the required s_B^f.

$$s_B^f \ge \left(S_f + 1 - \frac{D_R^f}{t_{BL}^f} \right) \cdot \frac{t_{BL}^f}{t_{BL}^f - t_B^f} \tag{25}$$

where $D_R^f \ge S_f \cdot t_B^f$.

B. Case 2 and Case 3

For Case 2, vehicles can download content from caching server at the MBS or RSU W_2, then from the remote server if necessary. If $t_{R_2}^f \ge t_{M,W_2}^f$, the worst case is that vehicles download from W_2 as much time as possible, so the segments combination path is $(H_2^f \rightarrow (H_1^f + H_2^f + H_3^f))$. Different ranges of D_R^f lead to different transmission patterns. For each pattern, the S_D^f can be estimated, then the required $s_{R_2}^f$ and s_B^f can be obtained.

1) *Pattern-1 – W_2*: if $D_R^f \le H_2^f$

$$\begin{cases} S_D^f = \left\lfloor D_R^f / t_{R_2}^f \right\rfloor \\ s_{R_2}^f \ge S_f \cdot \frac{H_2^f}{D_R^f}, s_B^f \ge 0 \end{cases} \tag{26}$$

2) *Pattern-2 – W_2* & MBS: if $H_2^f \le D_R^f \le s_B^f \cdot t_B^f + H_2^f$

$$\begin{cases} S_D^f = s_{R_2}^f + \left\lfloor \left(D_R^f - H_2^f \right) / t_B^f \right\rfloor \\ s_{R_2}^f \ge S_f + 1 - \frac{D_R^f - H_2^f}{t_B^f}, s_B^f \ge \frac{D_R^f - H_2^f}{t_B^f} \end{cases} \tag{27}$$

3) *Pattern-3 – W_2* & MBS & backhaul: if $D_R^f > s_B^f \cdot t_B^f + H_2^f$

$$\begin{cases} S_D^f = s_{R_2}^f + s_B^f + \left\lfloor \left(D_R^f - H_2^f - s_B^f \cdot t_B^f \right) / t_{BL}^f \right\rfloor \\ s_{R_2}^f + s_B^f \cdot \frac{t_{BL}^f - t_B^f}{t_{BL}^f} \ge S_f + 1 - \frac{D_R^f - H_2^f}{t_B^f} \end{cases} \tag{28}$$

If $t_{M,W_2}^f \ge t_{R_2}^f$, the worst case is that vehicles download from the MBS as much time as possible, so the segments combination path is $\max(H_1^f, H_3^f) \rightarrow \max(H_1^f, H_3^f) + H_2^f \rightarrow (H_1^f + H_2^f + H_3^f)$. The analysis of the required $s_{R_2}^f$ and s_B^f is similar to the case of $t_{R_2}^f \ge t_{M,W_2}^f$.

With a given D_R^f, we first find the possible transmission patterns for the worst case considering two possible relationships of t_{M,W_2}^f and $t_{R_2}^f$. Then, we get the required $s_{R_2}^f$ and s_B^f. For Case 3, the estimation of S_D^f and analysis of the required $s_{R_1}^f$ and s_B^f are similar to *Case 2*.

C. Case 4

Vehicles can download content from caching server at the MBS or RSU W_1, W_2, then from the remote server if necessary. By sorting $t_{R_1}^f$, $t_{R_2}^f$, and t_{M,W_w}^f in ascending order, we get the edge server order for the worst-case path. There are six possible paths for different orders. Two examples of segment combination paths are given as follows, in which *Path-1* (green line) and *Path-2* (red line) are shown in Fig. 7. The corresponding transmission pattern path for each segment combination path can be obtained, we use W_w^W and W_w^P denote vehicle driving through whole or part of RSU W_w.

1) If $W_1 > W_2 >$ MBS:
   $H_2^f \rightarrow (H_2^f - H_3^f) \rightarrow (H_2^f - H_4^f) \rightarrow (H_1^f - H_5^f)$
   Pattern: $W_1^P \rightarrow W_1^W + MBS \rightarrow W_1^W + MBS + W_2^P \rightarrow W_1^W + MBS + W_2^W$;
2) If $W_2 >$ MBS$> W_1$:
   $H_4^f \rightarrow (H_3^f - H_5^f) \rightarrow (H_2^f - H_5^f) \rightarrow (H_1^f - H_5^f)$
   Pattern: $W_2^P \rightarrow W_2^W + MBS \rightarrow W_2^W + MBS + W_1^P \rightarrow W_2^W + MBS + W_1^W$.

Then, we can calculate the required content placement for guarantee of successful transmission.

Different from LF content caching, the objective of PF content placement is to satisfy latency requirements rather than to achieve the lowest download latency. Thus, when design the caching scheme for file $f \in \mathbb{PF}$ with specific latency requirement, the required content placement scheme for it is $[s_{R_1}^f, s_{R_2}^f, s_B^f]$, which can be obtained by the lower bound of content placement for each case.

REFERENCES

[1] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[2] F. Lyu *et al.*, "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, Jun. 2020.

[3] Y. Hui, Z. Su, T. H. Luan, and J. Cai, "A game theoretic scheme for optimal access control in heterogeneous vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4590–4603, Jan. 2020.

[4] J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang, and Y. Zhang, "Smart and resilient EV charging in SDN-Enhanced vehicular edge computing networks," *IEEE J. Select. Areas Commun.*, vol. 38, no. 1, pp. 217–228, Jan. 2020.

[5] Y. Tang, N. Cheng, W. Wu, M. Wang, Y. Dai, and X. Shen, "Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3967–3979, Apr. 2019.

[6] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Content popularity prediction towards location-aware mobile edge caching," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 915–929, Apr. 2019.

[7] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proc. IEEE*, vol. 108, no. 2, pp. 274–291, Feb. 2020.

[8] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu, and Y. Jia, "An edge caching scheme to distribute content in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5346–5356, Jun. 2018.

[9] L. T. Tan, R. Q. Hu, and L. Hanzo, "Twin-timescale artificial intelligence aided mobility-aware edge caching and computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3086–3099, Apr. 2019.

[10] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, and L. Zhu, "A mobility-aware vehicular caching scheme in content centric networks: Model and optimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3100–3112, Apr. 2019.

[11] Y. Zhang, R. Wang, M. S. Hossain, M. F. Alhamid, and M. Guizani, "Heterogeneous information network-based content caching in the Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10 216–10 226, Oct. 2019.

[12] J. Chen *et al.*, "SDATP: An SDN-based traffic-adaptive and service-oriented transmission protocol," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 756–770, Jun. 2020.

[13] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5435–5444, Jun. 2018.

[14] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical VANETs," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1786–1801, Aug. 2018.

[15] L. Wang, H. Wu, Y. Ding, W. Chen, and H. V. Poor, "Hypergraph-based wireless distributed storage optimization for cellular D2D underlays," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2650–2666, Oct. 2016.

[16] D. J. MacKay, "Fountain codes," *IEEE Proc. Commun.*, vol. 152, no. 6, pp. 1062–1068, Dec. 2005.

[17] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791–1805, Aug. 2018.

[18] Z. Qu, B. Ye, B. Tang, S. Guo, S. Lu, and W. Zhuang, "Cooperative caching for multiple bitrate videos in small cell edges," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 288–299, Feb. 2020.

[19] Q. Li, W. Shi, X. Ge, and Z. Niu, "Cooperative edge caching in software-defined hyper-cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2596–2605, Nov. 2017.

[20] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1382–1393, May 2017.

[21] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. Shen, "Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach," *IEEE Netw.*, vol. 31, no. 5, pp. 14–20, Sep. 2017.

[22] T. H. Luan, X. Shen, and F. Bai, "Integrity-oriented content transmission in highway vehicular ad hoc networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2562–2570.

[23] H. Zhou *et al.*, "Chaincluster: Engineering a cooperative content distribution framework for highway vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2644–2657, Dec. 2014.

[24] 3GPP, "Vehicle-to-everything (V2X) services based on LTE; user equipment (UE) radio transmission and reception," ETSI, 3GPP, Sophia Antipolis, France, Tech. Rep. TR 36.786 V14.0.0, Mar. 2017.

[25] W. Xu, W. Shi, F. Lyu, H. Zhou, N. Cheng, and X. Shen, "Throughput analysis of vehicular internet access via roadside WiFi hotspot," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3980–3991, Apr. 2019.

[26] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE standard 802.11, Jan. 1999.

[27] D. Ciullo, V. Martina, M. Garetto, and E. Leonardi, "How much can large-scale video-on-demand benefit from users' cooperation?" *IEEE/ACM Trans. Netw.*, vol. 23, no. 6, pp. 1846–1861, Dec. 2014.

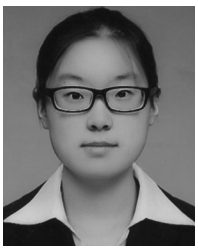
[28] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artif. Life*, vol. 5, no. 2, pp. 137–172, 1999.

[29] J. Branke, J. Branke, K. Deb, K. Miettinen, and R. Slowiński, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, vol. 5252, Berlin, Germany: Springer Science & Business Media, 2008.

[30] D. Alanis *et al.*, "A quantum-search-aided dynamic programming framework for pareto optimal routing in wireless multihop networks," *IEEE Trans. Commun.*, vol. 66, no. 8, pp. 3485–3500, Aug. 2018.

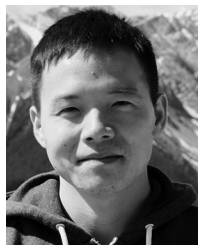
[31] M. Mavrovouniotis, F. M. Müller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1743–1756, Jul. 2017.

[32] H. Pirkul, "A heuristic solution procedure for the multiconstraint zero-one knapsack problem," Naval Res. Logistics, vol. 34, no. 2, pp. 161–172, Apr. 1987.



Jiayin Chen received the B.E. and M.S. degrees from the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China, in 2014 and 2016, respectively. She is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her research interests are in the area of vehicular networks and machine learning, with current focus on intelligent transport system and big data.

Huaqing Wu (Student Member, IEEE) received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively. She is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her current research interests include vehicular networks with emphasis on edge caching, resource allocation, and space-air-ground integrated networks.



Peng Yang (Member, IEEE) received the B.E. degree in communication engineering and the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013 and 2018, respectively. He was with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, as a Visiting Ph.D. Student from September 2015 to September 2017, and a Postdoctoral Fellow from September 2018 to December 2019. Since January 2020, he has been a Faculty Member with the School of Electronic Information and Communications, HUST. His current research interests focus on next-generation networking, mobile edge computing, video streaming and analytics.

Feng Lyu (Member, IEEE) received the B.S. degree in software engineering from Central South University, Changsha, China, in 2013 and the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently a Professor with the School of Computer Science and Engineering, Central South University. During respective September 2018–December 2019 and October 2016. October 2017, he was a Postdoctoral Fellow and was a Visiting Ph.D. Student with BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include vehicular networks, beyond 5G networks, big data measurement and application design, and could/edge computing. He is a member of the IEEE Computer Society, Communication Society, and Vehicular Technology Society.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests focus on network resource management, wireless network security, social networks, 5G and beyond, and vehicular ad hoc and sensor networks. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. Dr. Shen received the R.A. Fessenden Award in 2019 from IEEE, Canada, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo, and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16, the IEEE Infocom'14, the IEEE VTC'10 Fall, the IEEE Globecom'07, the Symposia Chair for the IEEE ICC'10, the Tutorial Chair for the IEEE VTC'11 Spring, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He was the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL and the Vice President on Publications of the IEEE Communications Society.