Delay-Minimized Edge Caching in Heterogeneous Vehicular Networks: A Matching-Based Approach

Huaqing Wu, *Graduate Student Member, IEEE*, Jiayin Chen, Wenchao Xu, *Member, IEEE*, Nan Cheng, *Member, IEEE*, Weisen Shi, *Graduate Student Member, IEEE*, Li Wang, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*— To enable ever-increasing vehicular applications, heterogeneous vehicular networks (HetVNets) are recently emerged to provide enhanced and cost-effective wireless network access. Meanwhile, edge caching is imperative to future vehicular content delivery to reduce the delivery delay and alleviate the unprecedented backhaul pressure. This work investigates content caching in HetVNets where Wi-Fi roadside units (RSUs), TV white space (TVWS) stations, and cellular base stations are considered to cache contents and provide content delivery. Particularly, to characterize the intermittent network connection provided by Wi-Fi RSUs and TVWS stations, we establish an on-off model with service interruptions to describe the content delivery process. Content coding then is leveraged to resist the impact of unstable network connections with optimized coding parameters. By jointly considering file characteristics and network conditions, we minimize the average delivery delay by optimizing the content placement, which is formulated as an integer linear programming (ILP) problem. Adopting the idea of student admission model, the ILP problem is then transformed into a many-to-one matching problem and solved by our proposed stable-matching-based caching scheme. Simulation results demonstrate that the proposed scheme can achieve near-optimal performances in terms of delivery delay and offloading ratio with low complexity.**

***Index Terms*— Heterogeneous vehicular networks, edge content caching, stable matching, fountain coding.**

Manuscript received September 17, 2019; revised January 2, 2020 and April 24, 2020; accepted June 9, 2020. Date of publication June 25, 2020; date of current version October 9, 2020. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the National Natural Science Foundation of China (NSFC) under Grant 91638204 and Grant 61871416, in part by the Fundamental Research Funds for the Central Universities under Grant 2018XKJC03, and in part by the Beijing Municipal Natural Science Foundation under Grant L192030. This article was presented in part at the IEEE GLOBECOM 2018. The associate editor coordinating the review of this article and approving it for publication was H. Pishro-Nik. *(Corresponding author: Nan Cheng.)*

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

Wenchao Xu is with the School of Computing and Information Sciences, Caritas Institute of Higher Education, Hong Kong (e-mail: wenchaoxu.ru@gmail.com).

Nan Cheng is with the State Key Laboratory of ISN, Xidian University, Xian 710071, China, and also with the School of Telecommunications Engineering, Xidian University, Xian 710071, China (e-mail: dr.nan.cheng@ieee.org).

Weisen Shi is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada, and also with Huawei Technologies Canada Inc., Ottawa, ON K2K 3J1, Canada (e-mail: w46shi@uwaterloo.ca).

Li Wang is with the Beijing Key Laboratory of Work Safety Intelligent Monitoring, School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: liwang@bupt.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.3003339

I. Introduction

THE Internet of vehicles (IoV), which enables ubiquitous information exchange and content sharing among vehicles, is envisioned as a promising solution to improve road safety and transportation efficiency [2]–[4]. Furthermore, IoV is also expected to support multifarious vehicular infotainment applications, including video/audio streaming, online gaming, and social networks, to improve the experience of both drivers and passengers. The Federal Communications Commission (FCC) has allocated 75 MHz bandwidth at the 5.9 GHz spectrum band for dedicated short-range communications (DSRC) for vehicular communications. However, DSRC mainly focuses on enabling vehicular safety applications [5], i.e., supporting rapid short message exchange, and the limited spectrum resource is insufficient to satisfy the quality of services of the bandwidth-intensive infotainment applications. Regarding the capacity constraint and high cost of data downloading in cellular networks, exploring alternative networks to support infotainment content delivery is imperative to offload cellular traffic and reduce communication cost.

Generally, different radio access technologies have their pros and cons, including transmission delay, throughput, jitter, etc. For instance, Wi-Fi technology can provide cost-effective and high-throughput Internet access for drive-by vehicles [6], but the Wi-Fi service is intermittent for vehicles due to limited coverage of Wi-Fi roadside units (RSUs) and high vehicle mobility. The vacant TV band, which is referred to as TV white space (TVWS) band, is also considered as important alternative spectrum resources to provide network connections with high penetration capabilities, low path loss, and wide coverage [7]. However, the TVWS network access might be interrupted by the primary users to ensure non-interfering spectrum access. Therefore, instead of relying on single radio for vehicular network access, utilizing multiple heterogeneous wireless technologies can make up for the deficiency and exploit their advantages. Thus, the heterogeneous vehicular networks (HetVNet) are deemed to be the future solution for vehicular users [8]. In this work, we consider HetVNet where the cellular networks coexist with Wi-Fi and TVWS access networks to increase the probability of successful content delivery and mitigate the cellular network congestion.

Although the wireless cellular traffic can be offloaded by utilizing HetVNets, the backhaul networks which support all

vehicular Internet data still face substantial pressure. In fact, a large portion of mobile multimedia traffic can be attributed to duplicated downloads of a small fraction of popular files. In vehicular networks (VNs), location-based applications also boost the repetitive download of location-oriented data (e.g., real-time traffic report, high definition maps, and so forth). To better utilize the computing and storage capacities of current network infrastructure and modern vehicles, it is possible to cache popular files closer to the user end. By enabling direct content delivery from the caching-enabled access points (APs), e.g., Wi-Fi RSUs, TVWS stations, and cellular base stations (CBSs), to the vehicles, edge caching on network edge infrastructure can significantly offload backhaul traffic [9]–[11]. Besides, since the repeated transmission on backhaul links can be avoided, the content delivery delay is reduced significantly, which is essential in VNs to facilitate efficient content delivery with rapidly changing network topology.

Motivated by the above, this work focuses on edge caching in HetVNets to serve the vehicular content requests. However, the design of efficient content caching schemes in the highly dynamic HetVNets faces several challenges. Firstly, as vehicles move fast with different velocities and accelerations, the contact duration between the vehicles and the fixed APs is limited and generally insufficient for content delivery. Secondly, caching different files in different types of APs leads to distinct delivery performances. Thus, the characteristics of the content files and access networks should be jointly considered, which further complicates the optimization of the content caching scheme. In addition, given that different types of access networks are generally managed by different operators, it is challenging to design an efficient content caching scheme without requiring inter-operator collaborations.

A. Related Works

As a promising technology dealing with the unprecedented growth of mobile data traffic, edge caching has attracted increasing attentions from both academia and industry. Without loss of generality, we categorize the existing works on edge caching into two parts: *edge caching with single access technology* and *edge caching in heterogeneous networks (HetNets)*.

1) Edge Caching With Single Access Technology: Edge caching in networks with low user mobility is studied in [12], [13], where the content files are cached in user devices. In VNs, caching in vehicles to assist V2V communications is studied in [14]–[16]. In [14], each vehicle makes its caching decision independently to cache popular content by considering different types of applications, the crucial features of data, and a set of key attributes of the VNs. In [15], vehicles communicate in a peer-to-peer manner and determine the content placement in a probabilistic way. Vehicle mobility prediction is studied in [16] to select vehicles with longer sojourn time in hot regions to cache the content files. Edge caching in infrastructures (or APs) is investigated in [2], [17], [18], where the popular contents are cached in homogeneous APs. Optimal content distribution in infrastructures is investigated in [17] by considering the available storage capacity and the link capacity. In [2], content placement in RSUs is optimized by utilizing an auction-based scheme in a two-way street scenario with equidistantly distributed RSUs. Considering that vehicles might also have cached some files, caching in APs is optimized by analyzing the impact of potential V2V content delivery.

2) Edge Caching in HetNets: Motivated by the network performance gain via the cooperation among different access technologies, HetNets have attracted broad interests, including data offloading in Wi-Fi/cellular HetNet [19], TVWS-based HetVNet for vehicular safety message [20], [21], and centralized data routing protocol design in LTE/DSRC HetNet [22]. Edge caching in HetNets has also gained increasing attention recently especially in low-mobility networks, e.g., caching content files in CBSs in a multi-tier cellular HetNet [23], [24] or caching in user devices in device-to-device communications underlying heterogeneous networks [25], [26]. Focusing on the VNs, the high vehicle mobility intensifies the design complexity of the edge caching scheme in HetVNets. In [27], the air-ground integrated vehicular network is investigated where the aerial high-altitude platforms proactively broadcast content files to vehicles while ground RSUs serve the vehicular requests via unicast on demand. In [28], an SDN-based HetVNet is considered to solve the coding-assisted broadcast scheduling problem by incorporating vehicular caching and network coding in scheduling decisions, where the SDN controller has a global view of vehicles, RSUs and CBSs. However, content files are assumed to be cached only in vehicles in [27], [28]. Hierarchical caching in EPC, RSUs, and vehicles is investigated in [29], where the interest topic and content of the vehicles are predicted and then the most popular content files are cached.

Summary: Edge caching in HetVNets is imperative for future vehicular content delivery and has attracted substantial research interests recently. In spite of the aforementioned works, the following problems, which are essential in the highly dynamic vehicular networks to provide enhanced and diversified wireless network access for moving vehicles and reduce delivery delay, are insufficiently studied in existing works: 1) in the time-varying and unreliable VNs, content delivery might encounter service interruptions, which significantly affect the caching performance and further the caching policy. However, this inherent vehicular characteristic has not been considered in any of the exiting works on HetVNet caching; 2) most existing works do not take full advantage of the heterogeneous network resources to boost the caching performance gain. Instead, caching content files simultaneously in multiple types of APs should be considered by taking into account their specific network characteristics, e.g., network coverage, network capacity, AP distribution, etc; and 3) the street layout or vehicle mobility patterns in most existing works are idealized or assumed to follow certain distributions, which is not practical.

B. Main Contributions

In this paper, we investigate content caching in HetVNet APs, i.e., Wi-Fi RSUs, TVWS stations, and CBSs, by taking into account the above-mentioned problems, to provide

enhanced and diversified wireless network access for moving vehicles, effectively offload cellular traffic, and reduce delivery delay. Considering the high vehicle mobility and the intermittent network access provided by Wi-Fi and TVWS transmissions, the volume of data that can be transmitted within one coverage area is limited. Therefore, caching the whole content files, especially large files, in the Wi-Fi RSUs or TVWS stations is inefficient since the complete downloading of one file requires multiple encounters with Wi-Fi RSUs or TVWS stations. To improve the caching capability, resist the impact of intermittent network connection, and enhance storage efficiency, coded caching can be applied to encode files into small pieces. Each AP only needs to cache the encoded pieces with smaller sizes. Recovering the entire file requires downloading a certain number of encoded packets, which may need the cooperation among the APs. Particularly, *Fountain Codes (also known as rateless erasure codes)* [30], [31] are used in this work to encode the files for the following reasons. Firstly, when using a fixed-rate erasure code, a receiver missing a source packet must successfully receive another source packet it has not previously received. This introduces additional overhead when coordinating different APs to serve a moving vehicle. Fountain codes, however, allow receivers to recover the original file by retrieving any subset of encoded packets of size slightly larger than the set of source packets, which is more flexible and reliable with lower communication overhead. Secondly, compared with other codes widely used in distributed data storage systems with $O(K^3)$ complexity, e.g., random linear codes, fountain codes have a superior decoding complexity of $O(K \ln K)$ [32], where K is the number of source packets to be encoded.

To minimize the content delivery delay for vehicles, we propose a matching-based caching scheme in HetVNets. By modeling the intermittent Wi-Fi/TVWS network connections as on-off service processes, the delivery delay is analyzed by applying partial repeat-after-interruption (PRAI) transmission mode. Based on the delay analysis, the caching placement problem, which is formulated as an integer linear programming (ILP) problem, is further transformed into a many-to-one preference-based matching problem between the content files and the HetVNet APs. More specifically, by designing the preference lists of content files and HetVNet APs based on file popularity, vehicle mobility, and APs' storage capacities, a student admission (SA) matching-based caching scheme is proposed, which is further solved by leveraging the Gale-Shapley (GS) algorithm [33] to obtain a stable matching. Simulation results show that the proposed scheme can effectively reduce the delivery delay and offload the cellular traffic.

The main contributions of this paper have three-folds:

1) By leveraging the interplay between file characteristics and network conditions, the problem of edge caching in multiple types of APs in HetVNets is investigated. Particularly, the dynamics of the content files (e.g., file size and file popularity) and the network connection (e.g., network capacity, AP distribution, and vehicle mobility pattern) are jointly considered in this work. The joint consideration facilitates efficient content caching schemes in the heterogeneous APs to achieve the minimal average delivery delay.
2) Taking into account the inherent time-varying and unreliable characteristics of VNs, we model the intermittent network connections to Wi-Fi RSUs and TVWS stations as on-off service processes with generally distributed on-periods and off-periods. Furthermore, coded caching is leveraged to resist the impact of unstable network connection. The coding parameters are optimized to adapt to the characteristics of different access networks. Then, by applying PRAI transmission mode, the proposed coded caching scheme can achieve a good balance between delivery delay and offloading performance (i.e., the volume of data downloaded without going through backhaul links).
3) The problem of content caching in HetVNets with service interruption is formulated as a many-to-one matching problem and solved by our proposed stable-matching-based caching algorithm. The construction of the two-sided preference lists is multi-objective, considering both the delivery delay and offloading performances. With the carefully designed preferences for content files and the APs, our matching-based caching scheme achieves a good performance with low time complexity.
4) We carry out extensive experimental results and provide insightful views on the suitability of various caching schemes in different HetVNet scenarios, by comparing multiple performance metrics including delivery delay, offloading ratio, cache hit rate, etc.

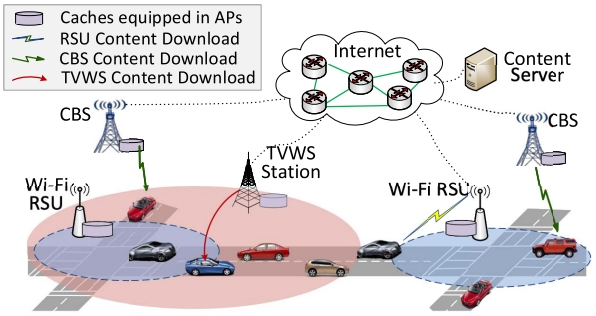
The remainder of this paper is organized as follows. In Section II, the system model and content delivery scenario in HetVNet are described and the problem formulation is given. In Section III, the detailed derivation of the average content downloading delay from HetVNet APs is analyzed, and the matching-based content placement optimization scheme is described in Section IV. Simulation results are carried out in Section V to demonstrate the performance of the proposed scheme. At last, we conclude the paper and direct our future work in Section VI.

II. System Scenario and Problem Formulation

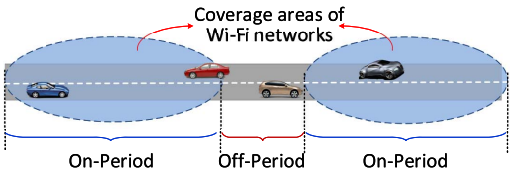
A. Scenario Description and Assumptions

Vehicular users in this work are assumed to be equipped with three radio interfaces for cellular, Wi-Fi, and TVWS communications.[1] Notice that, in addition to Wi-Fi and TVWS based access technologies, there exist many other techniques [34]. Although only Wi-Fi, TVWS, and cellular networks are considered in this work, our methodology and

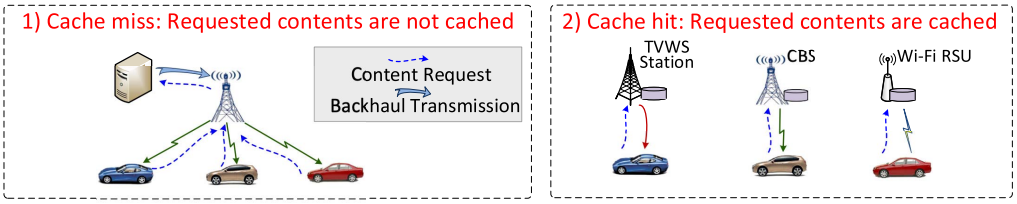
[1]Compared with the cellular and Wi-Fi radio interfaces which have been widely adopted, TVWS technology has not been widely implemented. However, many standards and research works have been done to facilitate vehicular communications in the TVWS band. Furthermore, there exist many industrial organizations (e.g., Carlson RuralConnect, Adaptrum, and 6 Harmonics) that provide devices and systems for TVWS Internet connectivity. Therefore, it can be expected that, just like Wi-Fi, the widespread implementation of TVWS technology will also become a reality in the near future.



(a) Communication scenario with coexistence of cellular, Wi-Fi, and TVWS access networks



(b) An example of on-off service processes of Wi-Fi networks



(c) Content delivery cases: cache miss and cache hit

Fig. 1. Caching-based content delivery scenario in HetVNet.

the proposed algorithm are applicable to HetVNets scenarios with more access techniques.

This work studies content caching in HetVNet APs, i.e., Wi-Fi RSUs, TVWS stations, and CBSs, and the communication scenario is depicted in Fig. 1(a). Considering urban and sub-urban scenarios where CBSs are densely deployed, we assume that the CBSs can provide seamless network connection for vehicles at any time. Content delivery from Wi-Fi RSUs or TVWS stations is available only when vehicles travel through the corresponding coverage areas, as shown in Fig. 1(b). The intermittent connections to Wi-Fi/TVWS networks are modeled as on-off processes, which will be introduced in detail in Section II-B. When a vehicle generates content requests, there are two possible cases as shown in Fig. 1(c): *1) cache miss:* if the requested files are not cached in the APs, the CBSs can fetch them from the content server via backhaul links and then deliver to the vehicle; and *2) cache hit:* the requested files are cached and the vehicle can download data from the APs based on the caching location.

In addition to the CBSs covering the entire target area, there also exist N_T TVWS stations and N_W Wi-Fi RSUs in the scenario. Notations used in this paper are summarized in Table I. The TVWS and Wi-Fi coverage radii are denoted by r_T and r_W, respectively. The bandwidth of a Wi-Fi RSU is shared by all vehicles associated to it, i.e., the average transmission data rate of one vehicle equals to $\overline{R}_W = R^a_W/N^a_W$, where R^a_W is the overall achievable aggregate rate and N^a_W is the average number of vehicles associated with one RSU. Likewise, the bandwidths of TVWS stations and CBSs are shared by vehicles associated to the same AP. The average TVWS and CBS transmission data rates are denoted by $\overline{R}_T$ and $\overline{R}_C$, respectively.

Notice that file popularity distribution has a great impact on the caching performance, including hit ratio, delivery delay, and cellular traffic offloading ratio. Let $\mathcal{F} = \{f_1, \cdots, f_M\}$ be the set of M files and $\mathbf{z}_f = [z_1, \cdots, z_M]$ be the vector representing the sizes of the M content files. All the files in set $\mathcal{F}$ are sorted by descending order based on the file popularity.[2] Thus, f_m is the m-th popular file with request probability p^m_{req}. Considering the fact that the popularity distribution of the network content items (e.g., YouTube videos) approximately follows Zipf's law, we model the file popularity by the Zipf distribution in this work: for a content file which ranks m, the probability that it is requested by vehicles is

$$p^m_{\text{req}} = \frac{1}{m^\xi} \Big/ \left(\sum\nolimits_{k=1}^{M} \frac{1}{k^\xi} \right), \tag{1}$$

where the exponent ξ ($\xi \geq 0$) controls the relative popularity of the files, i.e., a larger ξ means that the first few popular files account for the majority of requests.

B. On-Off Service Model

Recall that Wi-Fi RSUs and TVWS stations provide intermittent network access for drive-thru vehicles. To avoid harmful interference to licensed users, TVWS band can only be accessed by unlicensed users when it is not occupied by incumbent users.[3] Intermittent Wi-Fi and TVWS network services are modeled as on-off processes in this work. For Wi-Fi

[2]File popularity can be estimated based on the historical requests and predicted as studied in many existing works (e.g., [11]). Popularity prediction is beyond the scope of this paper.

[3]To conform to this rule, a database-assisted TVWS network architecture is used according to 802.11af, where a master TVWS device (TVWSD) (i.e., TVWS station in this work) can communicate with the geolocation database to obtain a list of available TVWS channels, while slave TVWSDs (i.e., vehicles in this work) can only access to TVWS channels under the control of the master TVWSD. The TVWSDs need to update the TVWS information subjecting to regulatory constraint, e.g., every 60 seconds according to 802.11af.

TABLE I
SUMMARY OF NOTATIONS

N_T, N_W	Number of Wi-Fi RSUs and TVWS stations, respectively.
r_T, r_W	Coverage radius of Wi-Fi RSUs and TVWS stations, respectively.
$\overline{R}_T, \overline{R}_W, \overline{R}_C$	Average TVWS, Wi-Fi, and CBS transmission rates, respectively.
k_m, α_m	The number of source (or encoded) packets and size of each packet for file f_m, respectively.
K_m	Number of encoded packets required to recover file f_m.
p^m_{req}	Probability that file f_m is requested by vehicles.
p^W_{suc}	Probability that vehicles can successfully download at least one encoded packet from Wi-Fi RSUs.
$p^W_{\max}$	Probability that vehicles can download enough encoded packets from RSUs without wasting time.
$p^W_{\text{on}}(x), p^W_{\text{off}}(x)$	Pmfs of the time length of the on-periods and off-periods for Wi-Fi content downloading.
a^W_m, a^T_m, a^C_m	Indicators showing the caching of file f_m in Wi-Fi RSUs, TVWS stations, and CBSs, respectively.
n^W_m, n^T_m	Number of encoded packets of f_m cached in each Wi-Fi RSU and TVWS station, respectively.
$\overline{D}^W_m, \overline{D}^T_m, \overline{D}^C_m, \overline{D}^B_m$	Average download delay of f_m from Wi-Fi, TVWS, CBS, and backhaul delivery, respectively.
C_T, C_W, C_C	Storage capacities of the Wi-Fi RSUs, TVWS stations, and CBSs, respectively.
$\mu^W_{\text{on}}, \mu^W_{\text{off}}$	Average time length in slots for on- and off-periods for Wi-Fi transmission.
σ	Probability that an arbitrary slot is an on-slot.
δ	Probability that on-period continues after an on-slot.
$\mathcal{P}_{\text{files}}(f_m,I)$, $\mathcal{P}_I(I,f_m)$	File f_m's preference over the APs and APs' preference over content files.

transmissions, as shown in Fig. 1(b), *on-periods* correspond to the time duration when Wi-Fi access is available and *off-periods* appear when the vehicle is not covered by Wi-Fi RSUs. For TVWS transmissions, the off-periods also include the duration when TVWS channels are occupied by incumbent users and not available for secondary access. In this work, we consider a discrete-time system to divide on-periods and off-periods into constant length intervals called slots. Taking Wi-Fi transmission as an example, we define the length of one slot as the time required to transmit one bit of data by Wi-Fi RSUs: $l = 1/\overline{R}_W$. Thus, an on-period with time length of T^W_{on} has T^W_{on}/l slots.

Let $p^W_{on}(x)$ and $p^W_{off}(x)$ be the probability mass functions (pmfs) of the duration of the on-periods and off-periods for the Wi-Fi transmission. Generally, the distributions of the on-off periods are affected by the characteristics of the RSUs (e.g., the deployment density and the coverage radius) and the mobility patterns of the vehicles. The distributions of the on- and off-periods can be obtained by observing the vehicle mobility traces in certain area, which leads to various distributions in different areas. Alternatively, the distributions can also be assumed to follow geometrical distributions for analysis simplicity. In this work, the on- and off-periods are assumed to be generally distributed, and the scheme proposed in this work can be applied to the cases with any known distributions for the on- and off-periods.

C. Fountain Coding

In this work, rateless fountain codes are used to encode files cached in TVWS stations and Wi-Fi RSUs due to their good computational efficiency and high flexibility and reliability. In the following, LT (Luby Transform) codes [35], the first proposed fountain codes, are briefly introduced and used in our subsequent discussions and performance evaluation.

When applying LT coding, a source file f_m is divided into k_m source packets $s_1, s_2, \cdots, s_{k_m}$, each of which has a size of $\alpha_m = \frac{z_m}{k_m}$, where z_m is the total size of f_m. Each encoded packet is obtained from the bitwise exclusive-or (XOR) of d randomly and independently chosen source packets, where d is drawn from a degree probability distribution $\Omega(d)$ with $1 \leq d \leq k_m$. In other words, with d obtained from $\Omega(d)$, a vector $(v_1, v_2, \cdots, v_{k_m})$ is generated randomly satisfying $v_i \in \{0,1\}$ for $i = 1, 2, \cdots, k_m$ and $\sum_{i=1}^{i=k_m} v_i = d$. The encoded packet is $\sum_{i=1}^{i=k_m} v_i s_i$ (bitwise sum modulo 2). From any set of K_m encoded packets (K_m is slightly larger than k_m, which will be explained at the end of this subsection), source file f_m can be decoded with success probability $1-\epsilon$, where ϵ is the decoding failure probability when receiving K_m encoded packets.

Following [35], the degree distribution $\Omega(d)$ follows the *Robust Soliton Distribution*. Let $R \equiv c\sqrt{k_m}\ln(\frac{k_m}{\epsilon})$ for some suitable constant $c > 0$ and $0 < \epsilon \leq 1$. Define

$$\rho(d) = \begin{cases} 1/k_m & \text{for } d=1 \\ 1/[d(d-1)] & \text{for } d=2, \cdots, k_m \end{cases},$$

$$\phi(d) = \begin{cases} R/(k_m d) & \text{for } d=1, \cdots, \frac{k_m}{R} - 1 \\ R\ln(R/\epsilon)/k_m & \text{for } d=\frac{k_m}{R} \\ 0 & \text{for } d > \frac{k_m}{R} \end{cases},$$

$$\beta_m = \sum\nolimits_{d=1}^{d=k_m} [\rho(d) + \phi(d)]. \tag{2}$$

Then we have $\Omega(d) = [\phi(d) + \rho(d)]/\beta_m$ for $d = 1, \cdots, k_m$. To ensure that the source file can be decoded with success probability no smaller than $1-\epsilon$, at least $K_m = k_m\beta_m$ encoded packets should be downloaded. Since $\sum_d \rho(d) = 1$, β_m is always larger than 1. Therefore, the improvement of caching reliability and storage efficiency in RSUs and TVWS stations are achieved at the expense of total storage space. Considering that data downloading from CBSs is always available, content files cached in the CBSs are stored without coding to avoid unnecessary extra storage occupancy and delivery delay.

D. Problem Formulation

In this work, content caching in HetVNet APs is investigated to minimize the average content delivery delay. For files with various popularities and data sizes, caching them in different types of APs with diverse coverage ranges, transmission data rates, and availabilities leads to distinct content delivery performances. Therefore, different files are suitable to be cached in different types of HetVNet APs, i.e., the network priority varies for different files. In this work, we jointly consider the file characteristics and network conditions to facilitate efficient content caching schemes in heterogeneous APs to minimize the average delivery delay.

Let a_m^W, a_m^T, and a_m^C indicate the caching of file f_m in Wi-Fi RSUs, TVWS stations, and CBSs, respectively, where

$$a_m^W = \begin{cases} 1, & \text{file } f_m \text{is cached in Wi-Fi RSUs} \\ 0, & \text{Otherwise} \end{cases},$$

$$a_m^T = \begin{cases} 1, & \text{file } f_m \text{is cached in TVWS stations} \\ 0, & \text{Otherwise} \end{cases},$$

$$a_m^C = \begin{cases} 1, & \text{file } f_m \text{is cached in CBSs} \\ 0, & \text{Otherwise} \end{cases}.$$

Notice that, one content file can only be cached in one type of APs in this work for the following reasons: 1) when adopting encoded caching, if a vehicle downloads encoded packets of a file from multiple access networks, the HetVNet APs need to negotiate and keep the same coding parameters to ensure successful content decoding. However, different types of access networks are managed by different service operators, which are competitors in the market and do not generally cooperate and coordinate the caching scheme; and 2) by constraining $a_m^W + a_m^T + a_m^C \leq 1$, the storage efficiency can be improved and more content files can be cached in the HetVNet APs to serve more vehicular requests, which facilitates the overall content delivery delay minimization.

The ideal case is that all the files are cached in the APs to avoid extra backhaul delays, which however is impractical due to limited storage capacities of the APs. Therefore, what kind of content files should be selected for caching and where they should be cached need to be carefully designed to reach an overall low delay. Uncached files can be downloaded from CBSs through backhaul links without coding. Therefore, focusing on the minimization of the average delivery latency for all the files in the library, we formulate our objective function as

$$\min_{\mathbf{A}_T, \mathbf{A}_W, \mathbf{A}_C} \sum\nolimits_{m=1}^{M} p_{req}^m \Big(a_m^W \overline{D}_m^W + a_m^T \overline{D}_m^T + a_m^C \overline{D}_m^C + \left(1 - a_m^T - a_m^W - a_m^C\right) \overline{D}_m^B \Big) \tag{3}$$

$$s.t. \sum\nolimits_{m=1}^{M} a_m^T n_m^T \alpha_m \leq C_T, \tag{3a}$$

$$\sum\nolimits_{m=1}^{M} a_m^W n_m^W \alpha_m \leq C_W, \tag{3b}$$

$$\sum\nolimits_{m=1}^{M} a_m^C z_m \leq C_C, \tag{3c}$$

$$a_m^W + a_m^T + a_m^C \leq 1, \quad \forall m = 1, \cdots, M, \tag{3d}$$

$$a_m^W, \quad a_m^T, a_m^C \in \{0, 1\}, \tag{3e}$$

where $\mathbf{A}_W = \left[a_1^W, \cdots, a_m^W, \cdots, a_M^W\right]$, $\mathbf{A}_T = \left[a_1^T, \cdots, a_m^T, \cdots, a_M^T\right]$, and $\mathbf{A}_C = \left[a_1^C, \cdots, a_m^C, \cdots, a_M^C\right]$. C_W, C_T, and C_C denote the storage capacities of the Wi-Fi RSUs, TVWS stations, and CBSs, respectively. n_m^T and n_m^W are the numbers of encoded packets of file f_m cached in each TVWS station and Wi-Fi RSU, respectively. $\overline{D}_m^W, \overline{D}_m^T, \overline{D}_m^C$, and $\overline{D}_m^B$ represent the average delays of downloading file f_m from the Wi-Fi, TVWS, CBS, and backhaul transmissions, respectively. Therefore, constraint (3a) indicates that the total size of files cached in the HetVNet APs cannot exceed the corresponding maximum storage capacities. Constraint (3b) shows that one content file can be cached in at most one type of APs.

III. Average Delivery Delay Analysis in HetVNet

To design a caching policy with minimized average overall content delivery delay, the delay performances of different delivery options (i.e., Wi-Fi, TVWS, CBS, and backhaul transmissions) should be analyzed. Firstly, for files encoded and cached in Wi-Fi RSUs and TVWS stations, the coding parameters are optimized based on the file characteristics and network conditions. Then the PRAI transmission mode is used to deliver the encoded packets and the corresponding average delivery delay is analyzed. For files not cached in Wi-Fi RSUs or TVWS stations, the delays of the CBS and backhaul transmission will also be given.

A. Determination of Coding Parameters

Determined by the AP deployment and vehicle mobility patterns, the distributions of the on- and off-periods for Wi-Fi and TVWS transmissions are spatially and temporally variant. For instance, the on-periods in urban scenarios generally sustain longer than in rural areas due to lower vehicle velocity and denser deployment of the APs. Targeting only on the urban scenarios, the distributions of the on-off periods vary in rush hours and in off-peak hours by virtue of different vehicle densities and velocities. The information of these distributions, however, can be gathered by monitoring vehicle mobility traces over a certain area. Generally, the distributions of the on-off periods might change over a day, but regularity can be observed for the same time period (e.g., rush hours) in different days. Without loss of generality, this work assumes that the characteristics of the on-off service processes are known in consequence of previous observations. In the following, content download from Wi-Fi RSUs is taken as an example to illustrate the impact of the on-off model on the determination of coding parameters.

Basically, the time length of the on-periods dominates the volume of data that can be transmitted within one coverage area. In our coding-based caching scheme, the size of one encoded packet is determined based on the distribution of on-periods to ensure that most vehicles can successfully download at least one packet when traveling through the RSUs' coverage areas. To ensure that vehicles driving through a Wi-Fi coverage area have a probability of at least p_{suc}^W to

Algorithm 1: Determination of Coding Parameters in Wi-Fi Transmission

$\mathcal{F}$: Set of all content files. α_m: Size of one encoded packet.
z_m: Size of file f_m. k_m^W: Number of source packets.
K_m^W: Number of encoded packets required to recover file f_m.
n_m^W: Number of encoded packets cached in each Wi-Fi RSU.

begin
  for $f_m \in \mathcal{F}$ **do**
    Calculate the coding parameters α_m^W and k_m^W based on Eqs. (4)$\sim$(5).
    if $k_m^W = 1$ *and* $\alpha_m^W = z_m$ **then**
      File f_m has small file size and can be cached in Wi-Fi RSUs without coding.
      $n_m^W = 1, K_m^W = 1$.
    else
      Obtain the value of n_m^W based on Eq. (6) and calculate the value of K_m^W based on analysis in Section II-C. Let $n_m^W = \min\{n_m^W, K_m^W\}$.
    end
  end
  Output: α_m^W, k_m^W, n_m^W, and K_m^W for any $f_m \in \mathcal{F}$.
end

successfully download at least one packet, we determine the coding parameters k_m^W and α_m^W by

$$\Pr(T_{on}^W \geq \frac{\alpha_m^W}{\overline{R}_W}) \geq p_{suc}^W \Rightarrow \sum_{T_{on}^W = \alpha_m^W/\overline{R}_W}^{\infty} p_{on}^W(T_{on}^W) \geq p_{suc}^W, \quad (4)$$

where T_{on}^W is the length of the on-period.

With any known distribution of the on-periods, we can obtain the upper bound for the value of α_m^W from (4), which is denoted by $\alpha_m^{\max}$. Considering that the encoding and decoding complexity of LT codes increase with the value of k_m^W [32], we choose the smallest possible value of k_m^W (largest possible value of α_m^W) as follows.

$$k_m^W = \lceil z_m/\alpha_m^{\max} \rceil, \quad \alpha_m^W = z_m/k_m^W. \quad (5)$$

Since vehicles spend different amount of time within different coverage areas, the volume of data downloaded by the vehicles within each RSU varies from one another. Given that fountain codes can generate unlimited number of encoded packets for each file, the number of packets cached in each RSU should be carefully designed. On one hand, a small number of cached packets gives rise to large delivery delays for vehicles spending long time in the coverage area, since they have to wait after downloading all the cached packets in the RSU. On the other hand, caching storage is wasted if too many packets are cached in each RSU while the vehicles can never download so much data within one coverage area.

To achieve a good trade-off between delivery delay and storage efficiency, the number of encoded packets cached in each RSU can be determined based on service requirements. For instance, to ensure that $p_{\max}^W \times 100\%$ of the vehicles can download enough number of packets for f_m within one RSU without wasting time, each RSU should cache at least n_m^W packets:

$$\Pr(T_{on}^W \leq \frac{n_m^W \alpha_m^W}{\overline{R}_W}) \geq p_{\max}^W \Rightarrow \sum_{0}^{\frac{n_m^W \alpha_m^W}{\overline{R}_W}} p_{on}^W(T_{on}^W) \geq p_{\max}^W. \quad (6)$$

With known pmf for on-periods, we can easily obtain the values of α_m^W, k_m^W, K_m^W, and n_m^W. Note that small files (with $k_m^W = 1$ and $\alpha_m^W = z_m$) can be cached without coding to avoid extra storage occupancy and delivery delay. In addition, when the value of n_m^W calculated from Eq. (6) is larger than K_m, then $n_m^W = K_m^W$ since K_m^W encoded packets are sufficient to recover f_m. The detailed procedure of determining the coding parameters is given in **Algorithm 1**. Similar analysis can be applied to coded caching parameter design in TVWS stations.

B. Effective Service Time

Let us define the terms *service time* and *effective service time (EST)*. Service time of a packet is the time required for transmission without interruption. By defining the length of one slot as the time required to transmit one bit by Wi-Fi RSUs, the service time of f_m in Wi-Fi transmission is equal to z_m slots.[4] On the other hand, the EST of delivering content file f_m is defined as the time period between the slot when the file request is generated and the end of the slot when transmission of the K_m-th packet is finished. For the Wi-Fi and TVWS content delivery, the EST includes the periods when content downloading is available and the time duration when the service is interrupted. In the following, the EST of content delivery is analyzed by taking the Wi-Fi content downloading as an example.

Considering that content files are encoded using LT codes and then cached in Wi-Fi RSUs, continuous-after-interruption (CAI) transmission mode is not suitable since the encoded data packets cached in the RSUs are different. Thus, PRAI transmission mode is adopted in this work. When delivering an encoded packet, if it is not finished before service interruption, this packet will be dropped and a new encoded packet of the same content file needs to be transmitted when the vehicle encounters another available Wi-Fi coverage in PRAI.

Taking the illustration in Fig. 2 as an example, a vehicle requests file f_m with size z_m at time t_0 and content delivery starts as soon as there is available network access at time t_1. Therefore, if the vehicle is not covered by a Wi-Fi RSU when generating the file request, it has to wait for $t_1 - t_0$ to get served; otherwise, $t_1 = t_0$. After successfully downloading the first packet of f_m, the transmission of the second packet is interrupted when the vehicle leaves the Wi-Fi coverage area. When entering the coverage area of another Wi-Fi RSU, the second packet (not necessarily the same as the unfinished one) needs to be re-transmitted. Thus, the EST of the second packet is $t_5 - t_2$. Correspondingly, the EST of file f_m is $t_7 - t_0$,

[4]Similarly, for TVWS transmission, we can define the length of one slot as the time required to transmit one bit by TVWS station, and the service time of file f_m in TVWS transmission is equal to z_m slots.

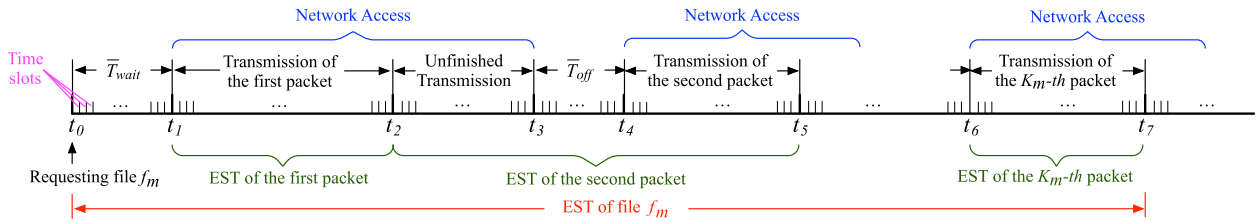


Fig. 2. Effective service time illustration for PRAI transmission mode.

which includes the waiting period $t_0 \sim t_1$ and the ESTs of K_m packets.

C. Average Delay of Wi-Fi and TVWS Delivery

Let μ_{on}^W and μ_{off}^W denote the average time length in slots for on- and off-periods. The probability that an arbitrary slot is an on-slot, denoted by σ, can be expressed as

$$\sigma = \mu_{on}^W / \left(\mu_{on}^W + \mu_{off}^W\right). \tag{7}$$

For a randomly generated file request, if the vehicle is out of the Wi-Fi service range, it has to wait for a certain time to get served. Denoting by T_{off}^W the length of the off-period in slots with mean μ_{off}^W, the average waiting time slots can be obtained by:

$$\begin{aligned}\overline{T}_{wait}^W &= E\left\{(1-\sigma)\cdot\sum_{x=1}^{T_{off}^W}\frac{1}{T_{off}^W}x\right\}\\ &= (1-\sigma)\cdot E\left\{\frac{1}{T_{off}^W}\cdot\frac{T_{off}^W(T_{off}^W+1)}{2}\right\}\\ &= \frac{(1-\sigma)}{2}(\mu_{off}^W+1).\end{aligned} \tag{8}$$

Denote by A the event that an on-period continues after an on-slot. Let δ denote the probability that event A happens and T_{on}^W denote the duration of on-periods with mean value μ_{on}^W. We have:

$$\begin{aligned}\delta &= \Pr(A) = \sum_{x=0}^{\infty}\Pr(A|T_{on}^W=x)\times\Pr(T_{on}^W=x)\\ &= \sum_{x=0}^{\infty}\frac{x-1}{x}\times p_{on}^W(x) = 1-E\left\{\frac{1}{T_{on}^W}\right\},\end{aligned} \tag{9}$$

which can be easily calculated with known $p_{on}^W(x)$.

To obtain the EST of the files, we first calculate the EST of one encoded packet. Referring to the repeat-after-interruption mode in [36], let $s_{n,\ell}^W(x)$ denote the probability that the remaining EST of a packet with size n bits equals x slots given that the remaining service time is ℓ slots and that the slot preceding the remaining EST is an on-slot. Thus $s_{n,\ell}^W(x) = 0$ for $x < \ell$ and

$$\begin{aligned}s_{n,\ell}^W(x) &= \delta s_{n,\ell-1}^W(x-1)\\ &\quad+(1-\delta)\sum_{j=1}^{\infty}p_{off}^W(j)s_{n,n-1}^W(x-j-1),\end{aligned} \tag{10}$$

which is obtained based on the on-off state of the first slot of the remaining EST.

Let $S_{n,\ell}^W(z)$ and $P_{off}^W(z)$ be the probability generating functions (pgfs) of $s_{n,\ell}^W(x)$ and $p_{off}^W(x)$, respectively, i.e.,

$$P_{off}^W(z) = \sum_{x=1}^{\infty}p_{off}^W(x)z^x, \quad S_{n,\ell}^W(z) = \sum_{x=1}^{\infty}s_{n,\ell}^W(x)z^x.$$

Thus, we have

$$\begin{aligned}S_{n,\ell}^W(z) &= \delta z S_{n,\ell-1}^W(z)+(1-\delta)zP_{off}^W(z)S_{n,n-1}^W(z),\\ S_{n,n}^W(z) &= \frac{\left(\delta z+(1-\delta)zP_{off}^W(z)\right)(\delta z)^{n-1}(1-\delta z)}{1-\delta z-(1-\delta)zP_{off}^W(z)\left[1-(\delta z)^{n-1}\right]}.\end{aligned} \tag{11}$$

The detailed derivation of (11) can be found in Appendix. Notice that $S_{n,0}^W(z) = 1$ because if there are no more bits to send, the downloading process ends in the current slot with probability 1. Based on the moment-generating property of pgf's, the average EST (in slots) of transmitting a packet with size n bits, denoted by $\overline{T}_n^W$, can be obtained by

$$\begin{aligned}\overline{T}_n^W &= \left.\frac{dS_{n,n}^W(z)}{dz}\right|_{z=1} = \left(\frac{\delta}{1-\delta}+\delta\mu_{off}^W\right)\left(\frac{1}{\delta^n}-1\right)\\ &= \frac{\delta}{1-\delta}\left(1+(1-\delta)\mu_{off}^W\right)\left(\frac{1}{\delta^n}-1\right).\end{aligned} \tag{12}$$

Thus, after waiting for $\overline{T}_{wait}^W$ slots, the following slot is an on-slot which can serve one unit of data. Then, the EST of transmitting the remaining $\alpha_m^W - 1$ units of the packet can be obtained by replacing n by $\alpha_m^W - 1$ in (12). After transmitting the first packet, the remaining $K_m^W - 1$ packets' service is preceded by an on-slot as the last slot of each packet's service is clearly an on-slot. Therefore, each of the remaining $K_m^W - 1$ packets has an EST of $\overline{T}_{\alpha_m}^W$ slots. Thus, the average EST (i.e., delivery delay) of file f_m, denoted by $\overline{D}_m^W$, is expressed as:

$$\overline{D}_m^W = (\overline{T}_{wait}^W + 1 + \overline{T}_{\alpha_m^W-1}^W + (K_m^W - 1)\overline{T}_{\alpha_m^W}^W)\times l. \tag{13}$$

As shown in the equation above, the EST of a content file is determined by the on-off distribution (affected by AP distributions and vehicle mobility patterns), network capacity, and the content file size. The same analysis procedure can be applied when calculating $\overline{D}_m^{TV}$, the average delay of downloading file f_m from TVWS stations, with different distributions of the on-off periods and network capacities.

D. Delivery Delay of Cellular Downloading

Recall that CBSs can provide seamless coverage for driving vehicles and uncoded caching scheme is applied for CBS caching. When a file is not completely delivered within the range of one CBS, a CAI transmission mode can be applied

when a vehicle travels through multiple CBSs. With uncoded caching and CAI transmission mode, no re-transmission is required and the average delay of downloading a file f_m from CBSs can be expressed as $\overline{D}_m^C = z_m/\overline{R}_C$. Note that the requests for content files not cached in CBSs should be served by the CBS through the backhaul links. Without loss of generality, CBSs are connected to the core network with wired backhaul links. Referring to [37], one hop can be assumed for the wired backhauls, and the average delay for transmitting file f_m with size z_m through wired backhaul links is:

$$\overline{D}_m^B = \overline{D}_m^C + \left(\left(1 + 1.28\frac{\lambda_b}{\lambda_g}\right)\kappa\right) \cdot (a + bz_m), \quad (14)$$

where λ_b and λ_g are the densities of the CBSs and the gateways, respectively. a, b, and κ are constants that reflect the processing capability of the nodesz[5]

IV. Matching-Based Content Caching Scheme

In this section, the content placement is optimized to minimize the overall content delivery delay based on the optimized coding parameters and the average delivery delay analysis given in Section III. Note that (3) is an ILP problem, and the optimal solution can be obtained by using ILP algorithms, e.g., branch and bound (B&B) algorithm. Considering the high time complexity of the B&B algorithm, a more efficient way to solve this problem is needed.

Construct a weighted bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{I}, \mathcal{E})$, where $\mathcal{V}$ is the set of content files, $\mathcal{I}$ is the set of HetVNet APs, and $\mathcal{E}$ is the set of edges connecting vertices in $\mathcal{V}$ and $\mathcal{I}$. Each edge has a weight which can be defined related to the content delivery delay as analyzed in Section III. Therefore, the content placement problem is actually a weighted bipartite b-matching problem, i.e., to find a subgraph $\mathcal{A} \subset \mathcal{G}$ to minimize the overall delay such that each vertex in $\mathcal{A}$ has at most b edges. Specifically, each file has $b = 1$, but the values of b for the APs are unknown for two reasons: 1) HetVNet APs have different storage capacities; and 2) caching different files in the same AP requires different storage sizes. Therefore, b-matching algorithms are not suitable for the content placement optimization. In this work, we adopt the idea of the SA model and apply GS-based stable matching algorithm to allocate content files to the HetVNet APs.

A. Complexity Analysis

As stated above, the optimal solution to the caching placement problem can be found by applying B&B algorithm. Although the B&B algorithm is guaranteed to find the optimal solution, its worst-case time complexity is as high as that of brute-force exhaustive search. In this work, considering that each file has four possible states (i.e., cached in Wi-Fi RSUs, TVWS stations, CBSs, or uncached), the worst-case complexity of the B&B algorithm is $\mathcal{O}(4^M)$, where M is the total number of content files. Although the practical searching times is not as large as 4^M in most cases, and the average complexity of the B&B algorithm can be reduced to be polynomial under some conditions, there is no complexity guarantee and the effectiveness of B&B is still limited by the potential exponential growth of the execution time as a function of problem size.

SA model solves the matching between students and colleges which have limited quotas, based on the two-sided preferences. By adopting the SA model in this work, we can map the content files to be the students and the APs to be the colleges. The content placement problem in this work can then be formulated as a many-to-one matching problem between the files and the HetVNet APs. That is, one file can only be cached in one type of APs, while one type of APs can cache multiple files up to its quota (i.e., storage capacity). Then, the GS algorithm can be leveraged to solve this matching problem with much lower time complexity, which is $\mathcal{O}(4\times M)$.

B. Preference Lists

The matching between files and the caching APs is processed based on the two-sided preferences. The construction of the preference lists can significantly affect the matching results and further the caching performance gain. Basically, the two-sided preference lists should be defined highly related to, but not always exactly the same as our optimization objective. In this work, a multi-objective construction of the preference lists is considered, by using two different metrics when designing the preference lists for content files and the APs.

Considering that our optimization objective is to minimize the overall delivery delay for all files, the preferences of content files over the HetVNet APs can be measured by the average delivery delay. Specifically, file f_m's preference over the APs is expressed as

$$\mathcal{P}_{files}(f_m, I) = \overline{D}_m^I, \quad (15)$$

where I refers to different ways to download file f_m, i.e., Wi-Fi RSUs, TVWS stations, and CBS transmissions. In other words, $\mathcal{P}_{files}(f_m, \text{Wi-Fi}) = \overline{D}_m^W$, $\mathcal{P}_{files}(f_m, \text{TVWS}) = \overline{D}_m^T$, and $\mathcal{P}_{files}(f_m, \text{CBS}) = \overline{D}_m^C$. Basically, it is preferred that a content file is cached in the type of APs leading to the lowest delivery delay. Thus, by sorting the elements in $\mathcal{P}_{files}(f_m, I)$ in ascending order, the first type of APs in f_m's preference list is the one that leads to the minimum delay.

When designing the preference lists for the APs, however, we do not prioritize the content files based on the delivery delay. Since the content delivery delay is largely dependent on file size, the delay oriented preference would let all APs prefer to cache files of smaller size rather than higher popularity. To address this issue, in this work, we define a new metric to rank the files based on file popularity and the volume of data that can be offloaded from the backhaul traffic. In other words, APs prefer to cache files with higher probability to be downloaded from them and with larger requested data size. By caching this kind of files, the APs can leverage their storage capacities more efficiently and offload more traffic with lower delivery latency. Thus, the APs ' preferences over file f_m are measured by

$$\mathcal{P}_I(I, f_m) = p_{\text{req}}^m \cdot \alpha_m^I \cdot K_m^I, \quad (16)$$

[5]The exact values of the above parameters can be obtained through fitting using real measurements, which goes beyond the scope of our paper.

Algorithm 2: Matching-Based Caching Optimization Algorithm

$\mathcal{F}$: Set of all the content files. z_m: Size of content file f_m. $\mathcal{F}_u$: Set of unmatched content files.
C_T, C_W, C_C: Storage capacities of the Wi-Fi RSUs, TVWS stations, and CBSs, respectively.
a_m^W, a_m^T, a_m^C: Indicators showing the caching of file f_m in RSUs, TVWS stations, and CBS, respectively.
$\mathcal{P}_{files}(f_m, I)$: Preference lists of content files.
$\mathcal{P}_I(I, f_m)$: Preference lists of HetVNet APs.

begin
  Initialize $\mathcal{F}_u = \mathcal{F}$.
  repeat
    for $f_m \in \mathcal{F}_u$ **do**
      Propose to the first type of APs I in its preference list $\mathcal{P}_{files}(f_m, I)$.
      Set $a_m^I = 1$ ($a_m^I \in \{a_m^W, a_m^T, a_m^C\}$) and remove I from $\mathcal{P}_{files}(f_m, I)$.
    end
    for $I \in$ *Wi-Fi RSUs, TVWS stations, CBSs* **do**
      $S_{I,req}^m = z_m$ if I is CBS; otherwise, $S_{I,req}^m = \alpha_m^I \cdot n_m^I$.
      if $\sum_{m\in\mathcal{F}}(a_m^I \cdot S_{I,req}^m) \le C_I$ **then**
        I keeps all the proposed content files and removes accepted files from $\mathcal{F}_u$.
      else
        I keeps the most preferred files under storage capacity constraint and rejects the rest; Remove these accepted files from $\mathcal{F}_u$.
        For the rejected files, set $a_m^I = 0$ and add them into $\mathcal{F}_u$.
      end
      $C_{I,remain} = C_I - \sum_{m\in\mathcal{F}}(a_m^I \cdot S_{I,req}^m)$
    end
  until $\mathcal{F}_u = \emptyset$ *or* $C_{I,remain} \le \min_{f_m\in\mathcal{F}_u} S_{I,req}^m, \forall I$;
  Output: a_m^W, a_m^T, and a_m^C for any $f_m \in \mathcal{F}$.
end

where the definition of I is the same as in (15). Thus, by sorting the elements in $\mathcal{P}_I(I, f_m)$ in descending order, the first file in each type of APs' preference list is file f_m leading to the maximum average offloading data size.

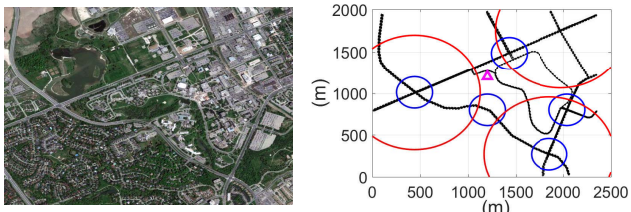
C. Matching-Based Content Placement Policy

In this subsection, we illustrate the caching placement scheme in HetVNet with on-off service model. The details are summarized in **Algorithm 2**. Firstly, for every content file f_m, the average delay performance is analyzed for all the HetVNet APs, based on which the preference lists are constructed as discussed in (15) and (16). After that, the GS algorithm is exploited to solve the SA-based many-to-one matching problem between the content files and the APs. The matching process can be described as follows:

Step 1: Each content file proposes to its current most favorite caching APs and then removes this type of APs from its preference list.

Algorithm 3: Overall Process of the Proposed Matching-Based Caching Scheme

begin
  Step 1: Determine the coding parameters according to **Algorithm 1**.
  Step 2: Analyze the content delivery delay of the coded caching scheme in HetVNets with service interruption based on **Sections III-C and III-D**.
  Step 3: Construct multi-objective two-sided preference lists based on **Section IV-B**.
  Step 4: Optimize content placement according to the matching-based algorithm in **Algorithm 2**.
end



(a) Map (b) AP deployment

Fig. 3. Simulation settings.

Step 2: Each type of APs check all the received proposals from the files, including both the new proposals and those accepted in former iterations, and then accept the most preferred files within the storage capacity constraint and reject the rest.

Step 3: For all the rejected files, go to **Step 1**. The matching process terminates when all the files are successfully cached or all the APs' storage capacities are occupied.

As a summary, **Algorithm 3** shows the overall process of our proposed matching-based content caching optimization scheme with on-off service model.

V. Performance Evaluation

A. Simulation Setting

We conduct simulations based on the real scenario of University of Waterloo campus. The campus map is shown in Fig. 3a and the main roads are drawn in Fig. 3b. Without loss of generality, we assume that vehicles in the target region can always access to the CBS, which is marked as the pink triangle in Fig. 3b. Blue circles in Fig. 3b represent the coverage areas of five Wi-Fi RSUs, and the red circles are the coverage areas of three TVWS stations. To simulate vehicle traffic, we use VISSIM simulation tool to generate the traffic of 200 vehicles in the campus scenario. The size of the content files is within the range of [0 MB, 1000 MB]. The file popularity follows Zipf distribution with exponent $\xi = 0.7$. The default values of main simulation parameters are listed in Table II unless otherwise specified.

Recall that we assume known distributions of the on-off periods in this work. Considering that the time length of the on-off periods might not follow any well-known distributions

TABLE II
SIMULATION PARAMETERS

$[r_W, r_T]$: Coverage radii of Wi-Fi RSUs and TVWS stations	[150, 600] m
$[R_W^a, R_T^a, R_C^a]$: Aggregate rates of a Wi-Fi RSU, TVWS station, and CBS	[65, 54, 128] Mbps
c and ϵ: Constant in Robust Soliton Distribution and decoding failure probability	0.1 and 0.05
p_{suc}^W and p_{suc}^T: Probabilities that vehicles can download at least one packet from Wi-Fi RSUs and TVWS stations	0.99
$p_{\max}^W$ and $p_{\max}^T$: Probability that vehicles can download enough packets from Wi-Fi RSUs and TVWS stations without waiting	0.9
$[C_W, C_T, C_C]$: Storage capacities of a Wi-Fi RSU, a TVWS station, and a CBS	[10, 10, 20] GB

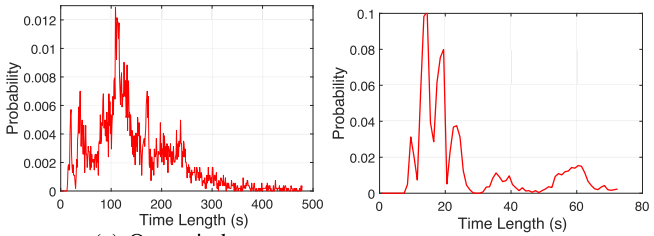
(e.g., exponential distribution and normal distribution) in practice and hence a general distribution is assumed for the on-off service models. In our simulation, based on the mobility traces generated by VISSIM and the deployment of the APs, we can easily obtain the time vehicles spend in each kind of APs, thus distributions of the on-off periods can be acquired. For instance, our simulation uses Matlab to process the vehicle trace files, and the distributions of the on-off service periods for TVWS transmission are shown in Fig. 4. Note that the time length of the on- and off-periods is affected by factors including, but not limited to the distance between the TVWS stations, the road layout, and the average vehicle velocity. Therefore, the distributions vary in different target regions with various AP deployments. Simulations in this work leverage the on-off TVWS service time distributions in Fig. 4, and the on-off Wi-Fi service time distribution can be obtained in a similar way. However, the caching scheme proposed in this work can be applied to scenarios with any other well-known distributions for the on-off periods.

To evaluate the delay performance, we monitor all vehicles in the target region, which generate content requests based on the file popularity distribution. Then a vehicle is randomly chosen at a random time instant and its data downloading performance is observed. All the following simulation results are averaged over 1000 trials.

B. Tradeoff Between Delay Performance and Complexity

Fig. 5 shows the impact of the number of content files on the achievable delay performance[6] and complexity of the algorithms. In addition to the B&B and our matching-based algorithm, one evolutionary algorithm, the particle swarm optimization (PSO) algorithm, is also included in the performance comparison to further illustrate the effectiveness and efficiency of our proposed scheme. Intuitively, with more content files in the network, the delivery delay per unit data and the time complexities of all the three algorithms increase. As shown in Fig. 5a, the delivery delay increases because more files need to be fetched from backhaul transmission due to limited storage capacities. The delay performance of the B&B algorithm outperforms that of the GS matching-based algorithm, while the performance gap is insignificant. On the other hand, comparing to the B&B algorithm which has exponentially increased complexity over the network size, the GS-based matching algorithm is significantly less time-consuming, which can be seen in Fig. 5b, especially when the system scales. The PSO algorithm, as shown in the figure, achieves larger delivery delay with longer simulation time when compared with the proposed matching-based algorithm. The delay performance of the PSO algorithm can be further improved, while the corresponding simulation time will also increase significantly. Therefore, the proposed algorithm is a favorable choice to reduce the time complexity with modest delay performance loss, especially in complex or heterogeneous networks with a large number of files.

[6]Given that overall content delivery delay is dominated by file sizes, average delay per unit data (sec/bit), which is the ratio of the overall delay defined in (3) over the total size of the requested data, is adopted in the simulation to better illustrate the content delivery delay performance.



(a) On-periods (b) Off-periods

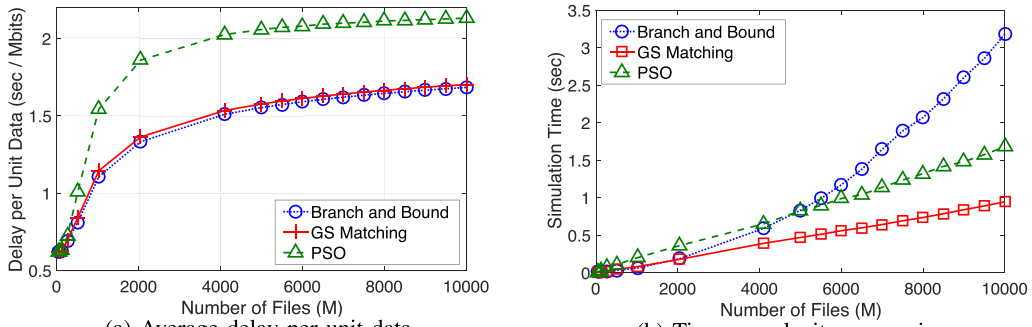
Fig. 4. Distributions of on-off periods for TVWS transmission.

C. Impact of File Size

Fig. 6 shows the impact of file size on the average delay performance of downloading files from Wi-Fi RSUs, TVWS stations, CBS, or backhaul transmission. Since the average transmission rates of CBS and backhaul delivery are mainly dominated by the number of vehicles sharing the spectrum and the deployment of the CBSs, the average delays of these two kinds of transmissions keep unchanged with increasing file size, as shown in Fig. 6. The average delays per unit data for Wi-Fi and TVWS transmissions decline with larger file size, and the former has better delay performance than the latter. For small-size files, Wi-Fi and TVWS transmissions have poor delay performance. The reason is that, when compared to the average waiting time, the required service time for small files plays a minor part in the EST in our on-off service models, therefore leading to a large delay per unit data. When file size grows, an increasing part of the EST comes from the service time rather than the time wasted in waiting for service. Finally, the average delay per unit data converges to a constant value which is determined by the average Wi-Fi/TVWS transmission rate and the ratio of the average time length of on-periods over that of the off-periods. Therefore, it is advisable that small files (e.g., texts or pictures) are cached in the CBS without coding, while large files such as movies and high definition maps should be encoded into packets and cached in Wi-Fi RSUs or TVWS stations to achieve performance improvement.

D. Coded Caching vs. Uncoded Caching

Recall that content files are encoded and cached in the Wi-Fi RSUs and TVWS stations. If the transmission of an



(a) Average delay per unit data (b) Time complexity comparison

Fig. 5. Delay and complexity performance comparison between B&B, PSO, and GS matching algorithms.

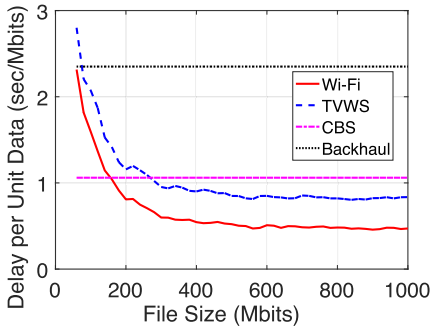


Fig. 6. Average delay per unit data vs. file size.

encoded packet is interrupted when vehicles move out of the coverage area, a PRAI transmission mode is adopted as explained in Section III-B. In contrast, uncoded caching scheme indicates that files are cached entirely in the APs and the CAI transmission mode can be applied when the vehicle travels through multiple APs. With uncoded caching and CAI transmission mode, one may naively believe that the delivery delay can be reduced since no re-transmission is required. However, uncoded caching leads to low storage efficiency especially for large files, which further affects the overall delay and offloading performance. Therefore, in this subsection, we compare the performances of the coded caching and uncoded caching schemes to dispel any wishful thinking.

Similar to the analysis in Section III-C, the EST of transmitting a file with size z_m by using the CAI transmission mode is analyzed as follows (taking Wi-Fi transmission as an example). Given that the slot preceding the effective service time is an on-slot, the probability that the EST of a packet of length n bits equals ℓ slots is:

$$s_n^{\text{W,CAI}}(\ell) = \delta s_{n-1}^{\text{W,CAI}}(\ell-1) + (1-\delta)\sum_{j=1}^{\infty} p_{off}^{W}(j) s_{n-1}^{\text{W,CAI}}(\ell-j-1), \quad (17)$$

The corresponding pgf of $s_n^{\text{W,CAI}}(\ell)$ is:

$$S_n^{CAI}(z) = \left[\delta z + (1-\delta) z P_{off}^{W}(z)\right] S_{n-1}^{CAI}(z) \Rightarrow S_n^{CAI}(z) = \left[\delta z + (1-\delta) z P_{off}^{W}(z)\right]^n \quad (18)$$

Therefore, the EST of transmitting a file with size z_m in CAI mode is:

$$\overline{D}_m^{\text{W, CAI}} = \left(\overline{T}_{wait}^{\text{W, CAI}} + 1 + \left.\frac{dS_{\alpha_m-1}^{CAI}(z)}{dz}\right|_{z=1}\right) \times l = \left(\frac{(1-\sigma)\mu_{off}^{W}}{2} + 1 + (z_m - 1)\left[1 + (1-\delta)\mu_{off}^{W}\right]\right) \times l. \quad (19)$$

In Fig. 7, we compare the delay and offloading performances of coded and uncoded content caching schemes. As shown in Fig. 7a, the average delay per unit data for both caching schemes increases with more content files, because more files need to be retrieved by backhaul transmissions due to limited storage capacities of the HetVNet APs. It is worth noting that when the number of files is large enough ($\geq$ 50), coded caching scheme outperforms the uncoded scheme in terms of overall average delay, since the former can cache more content files in the APs due to higher storage efficiency.

Define offloading ratio as the ratio of the data volume downloaded without going through backhaul links over the overall requested data volume. As shown in Fig. 7b, the offloading ratio performances of the coded and uncoded caching schemes are identical when there are less than twenty files, since the HetVNet APs can successfully cache all the files. With increasing number of files, a smaller portion of files can be cached in the APs for uncoded caching scheme, leading to higher probability of backhaul downloading and lower offloading ratio. On the other hand, despite the decline of offloading ratio, the coded caching scheme has significant advantage in terms of offloading performance when compared with the uncoded scheme. Stemming from the above observations, uncoded caching is preferred in scenarios with small number of content files, while coded caching scheme is more suitable when network scales to achieve better delay and offloading performances.

E. Single-Access-Based Caching vs. Multi-Access-Based Caching

In the proposed caching scheme, one file is allowed to be cached in only one type of APs to improve caching storage efficiency without requiring the cooperation among

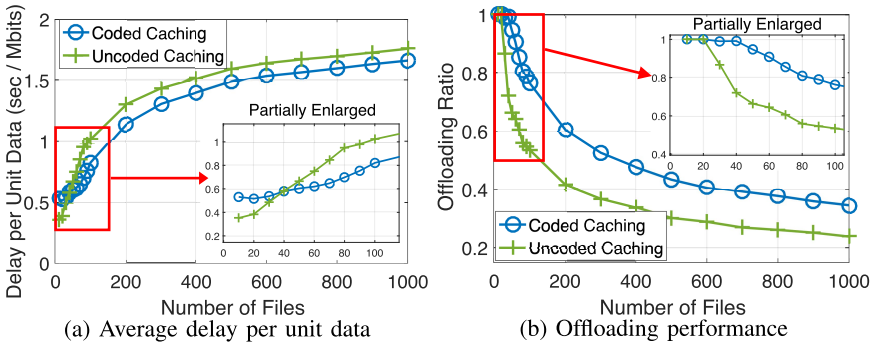


Fig. 7. Delay and offloading performance comparison for coded caching and uncoded caching schemes.

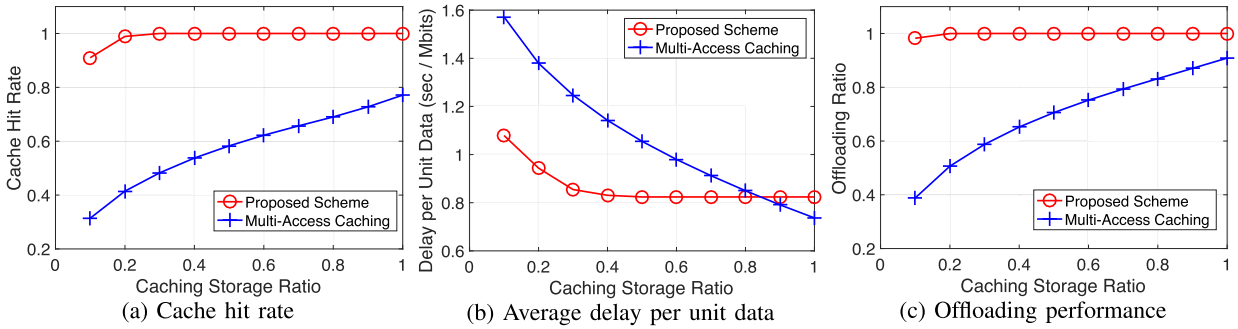


Fig. 8. Cache hit rate, delay, and offloading performance comparison between the proposed scheme and multi-access-based caching scheme.

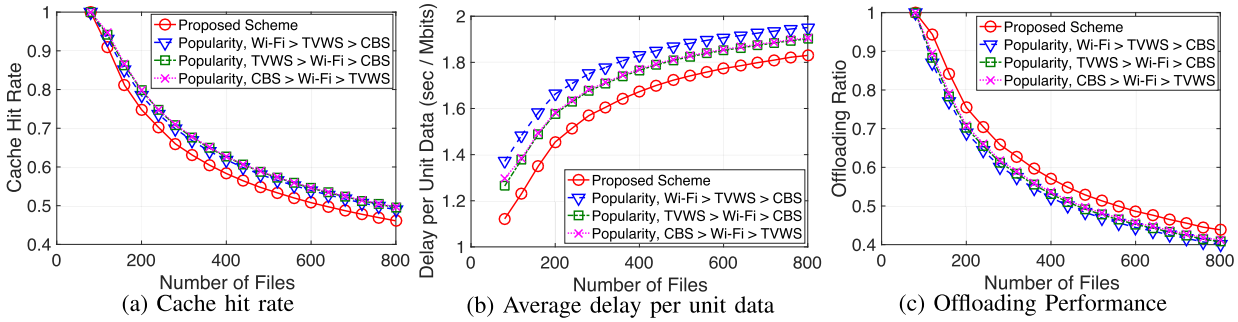


Fig. 9. Cache hit rate, delay, and offloading performance comparison between the proposed scheme and popularity-based caching schemes.

different access networks. Intuitively, for the cached files, the delivery delay can be further reduced if they are stored in all the APs such that the vehicles can get served within any access network coverage. However, whether the overall delay performance can be improved remains unknown. In this subsection, we compare the proposed caching scheme with the case where the files are encoded and cached in all the APs, named as the "Multi-Access Caching", to reveal insights on the suitability of these two types of caching schemes in different scenarios. Notice that, in "Multi-Access Caching" scheme, the files are encoded with the same coding parameters (i.e., in **Algorithm 1**, let $\alpha_m = \min\{\alpha_m^W, \alpha_m^T\}$ and $k_m = z_m/\alpha_m$, then n_m^I and K_m^I (I refers to Wi-Fi RSUs, TVWS stations, or CBSs) can be calculated accordingly.).

In Fig. 8, we provide the performance comparison between the proposed scheme and the "Multi-Access Caching" scheme with different caching storage ratio.[7] With increasing storage capacities of the HetVNet APs, both caching schemes achieve better cache hit rate, delivery delay, and offloading performances. As shown in Figs. 8(a) and 8(c), the proposed scheme can effectively cache files in different APs and has a high cache hit rate and offloading ratio. On the other hand, when adopting the "Multi-Access Caching" scheme, much fewer content files are cached in the HetVNet APs and offloaded from the backhaul transmission since each cached file occupies substantial caching storage. Although the cached files can be downloaded faster in the "Multi-Access Caching" scheme, the overall delivery delay performance is unsatisfactory due to the substantial backhaul transmission when the caching storage capacity is limited. Nevertheless, the "Multi-Access Caching"

[7]All the APs are assumed to have the same storage capacity, and the caching storage ratio is the ratio of the storage capacity of one AP over the total size of all the content files.

scheme outperforms the proposed scheme in terms of overall delivery delay when the storage capacity is large enough, e.g., when the cache size of each AP is no less than 0.9 of the total size of all the files as shown in Fig. 8b. To summarize, the "Multi-Access Caching" scheme is a favourable choice when the storage capacity is sufficiently large, while the proposed scheme works well in general cases with limited caching resources.

F. Performance Comparison With Popularity-Based Caching Schemes

In this subsection, we compare our proposed caching scheme with the popularity-based caching schemes. In particular, the popularity-based schemes prioritize and cache the files based only on file popularity. As shown in Fig. 9, we use 'Popularity' to denote the popularity-based caching schemes. In addition, 'Wi-Fi $>$ TVWS $>$ CBS' denotes that content files are cached in the Wi-Fi RSUs with the highest priority, i.e., the most popular content files are first cached in Wi-Fi RSUs until reaching the caching capacity, then in the TVWS stations, and the CBSs have the lowest priority. 'TVWS $>$ Wi-Fi $>$ CBS' and 'CBS $>$ Wi-Fi $>$ TVWS' are defined in a similar way. In addition to the average delay and offloading performance, the cache hit rate[8] is also considered to further compare the performance of the proposed algorithm and the popularity-based algorithms.

As shown in Fig. 9, with a small number of content files, all the files can be cached in the APs, which means that the cache hit rate and offloading ratio are both equal to 1 for all the algorithms. When the number of files increases, more files need to be retrieved from backhaul links, thereby leading to lower cache hit rate, longer delivery delay, and lower offloading ratio for all the algorithms. As shown in Fig. 9a, the popularity-based schemes have higher cache hit rate than the proposed scheme since the former ones only cache the most popular files. On the other hand, in addition to the file popularity, the proposed scheme also takes file size, vehicle mobility, and network characteristics into consideration to wisely cache different types of files in the APs. Therefore, the proposed scheme presents better delay and offloading performances as shown in Figs. 9b and 9c.

VI. Conclusion

In this paper, we have investigated content caching in HetVNet APs to provide enhanced and diversified wireless network access for moving vehicles and reduce delivery delay, by considering the impact of factors including file popularity, vehicle mobility, network service interruption, and storage capacities of the APs. Specifically, we have proposed a matching-based scheme with multi-objective two-sided preference lists to optimize the content placement problem. Simulation results have validated the effectiveness of the proposed content caching scheme, which can further provide an insight into the optimization of content sharing in different network conditions. This work provides a theoretical basis for future studies related to content caching in heterogeneous networks, e.g., the emerging space-air-ground integrated network (SAGIN), which is a specific HetNet involving satellites, UAVs, and ground devices. For the future work, we will further investigate the design of cooperative caching schemes where content files are delivered with cooperation among different types of APs in various types of HetVNets to further improve the delivery performance.

[8]The cache hit rate is defined as the ratio of the number of cache hit in all the APs' caches to the overall number of vehicular content requests.

Appendix

According to Eq. (13), the pgf of $s^W_{n,\ell}(x)$ is

$$\begin{aligned}
&S^W_{n,\ell}(z)\\
&= \sum_{x=1}^{\infty} \delta s^W_{n,\ell-1}(x-1)z^x\\
&\quad + \sum_{x=1}^{\infty}(1-\delta)\sum_{j=1}^{\infty} p^W_{off}(j)s^W_{n,n-1}(x-j-1)z^x\\
&= \delta z \sum_{x=0}^{\infty} s^W_{n,\ell-1}(x)z^x\\
&\quad + (1-\delta)z\sum_{x=1}^{\infty}\sum_{j=1}^{\infty} p^W_{off}(j)z^j s^W_{n,n-1}(x-j-1)z^{x-j-1}\\
&= \delta z S^W_{n,\ell-1}(z) + (1-\delta)z P^W_{off}(z) S^W_{n,n-1}(z).
\end{aligned}$$

For notational simplicity, let $\zeta = (1-\delta)zP^W_{off}(z)$. Thus we have $S^W_{n,\ell}(z) = \delta z S^W_{n,\ell-1}(z) + \zeta S^W_{n,n-1}(z)$.

By substituting different values for ℓ, we have:

- $\boldsymbol{\ell = n}$ **:** $S^W_{n,n}(z) = \delta z S^W_{n,n-1}(z) + \zeta S^W_{n,n-1}(z)$
  $\Rightarrow \quad S^W_{n,n}(z) = (\delta z + \zeta)\, S^W_{n,n-1}(z)$;
- $\boldsymbol{\ell = n-1}$ **:** $S^W_{n,n-1}(z) = \delta z S^W_{n,n-2}(z) + \zeta S^W_{n,n-1}(z)$
  $\Rightarrow S^W_{n,n-1}(z) = \frac{\delta z}{1-\zeta} S^W_{n,n-2}(z)$;
- $\boldsymbol{\ell = n-2}$ **:** $S^W_{n,n-2}(z) = \delta z S^W_{n,n-3}(z) + \zeta S^W_{n,n-1}(z)$
  $\Rightarrow S^W_{n,n-1}(z) = \frac{(\delta z)^2}{1-\zeta-\zeta\delta z} S^W_{n,n-3}(z)$;
- $\boldsymbol{\ell = n-3}$ **:** $S^W_{n,n-3}(z) = \delta z S^W_{n,n-4}(z) + \zeta S^W_{n,n-1}(z)$
  $\Rightarrow S^W_{n,n-1}(z) = \frac{(\delta z)^3}{1-\zeta-\zeta\delta z-\zeta(\delta z)^2} S^W_{n,n-4}(z)$;
  $\cdots$

By deductive proof, we have

$$\begin{aligned}
S^W_{n,n-1}(z) &= \frac{(\delta z)^{n-1}}{1-\zeta\sum_{j=0}^{n-2}(\delta z)^j} S^W_{n,0}(z)\\
&= \frac{(\delta z)^{n-1}(1-\delta z)}{1-\delta z-\zeta[1-(\delta z)^{n-1}]},\\
S^W_{n,n}(z) &= (\delta z+\zeta)\,\frac{(\delta z)^{n-1}(1-\delta z)}{1-\delta z-\zeta[1-(\delta z)^{n-1}]}\\
&= \frac{\left(\delta z + (1-\delta)zP^W_{off}(z)\right)(\delta z)^{n-1}(1-\delta z)}{1-\delta z-(1-\delta)zP^W_{off}(z)\left[1-(\delta z)^{n-1}\right]}.
\end{aligned}$$

References

[1] H. Wu, W. Xu, J. Chen, L. Wang, and X. Shen, "Matching-based content caching in heterogeneous vehicular networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.

[2] Z. Hu, Z. Zheng, T. Wang, L. Song, and X. Li, "Roadside unit caching: Auction-based storage allocation for multiple content providers," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6321–6334, Oct. 2017.

[3] Y. Wang, L. Huang, T. Gu, H. Wei, K. Xing, and J. Zhang, "Data-driven traffic flow analysis for vehicular communications," in *Proc. IEEE Conf. Comput. Commun.*, Toronto, ON, Canada, Apr/May 2014, pp. 1977–1985.
[4] Y. Cao *et al.*, "A trajectory-driven opportunistic routing protocol for VCPS," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 6, pp. 2628–2642, Dec. 2018.
[5] 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.
[6] 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.
[7] T. Jiang, Z. Wang, L. Zhang, D. Qu, and Y.-C. Liang, "Efficient spectrum utilization on TV band for cognitive radio based high speed vehicle network," *IEEE Trans. Wireless Commun.*, vol. 13, no. 10, pp. 5319–5329, Oct. 2014.
[8] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2377–2396, 4th Quart., 2015.
[9] T. H. Luan, L. X. Cai, J. Chen, X. S. Shen, and F. Bai, "Engineering a distributed infrastructure for large-scale cost-effective content dissemination over urban vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 3, pp. 1419–1435, Mar. 2014.
[10] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Scaling behavior for Device-to-Device communications with distributed caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4286–4298, Jul. 2014.
[11] 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.
[12] W. Chen and H. V. Poor, "Caching with time domain buffer sharing," *IEEE Trans. Commun.*, vol. 67, no. 4, pp. 2730–2745, Apr. 2019.
[13] L. Wang, H. Wu, Z. Han, P. Zhang, and H. V. Poor, "Multi-hop cooperative caching in social IoT using matching theory," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2127–2145, Apr. 2018.
[14] D. D. Van, Q. Ai, Q. Liu, and D.-T. Huynh, "Efficient caching strategy in content-centric networking for vehicular ad-hoc network applications," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 703–711, Sep. 2018.
[15] N. Kumar and J.-H. Lee, "Peer-to-Peer cooperative caching for data dissemination in urban vehicular communications," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1136–1144, Dec. 2014.
[16] 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.
[17] G. Mauri, M. Gerla, F. Bruno, M. Cesana, and G. Verticale, "Optimal content prefetching in NDN Vehicle-to-Infrastructure scenario," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2513–2525, Mar. 2017.
[18] 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.
[19] H. Park, Y. Jin, J. Yoon, and Y. Yi, "On the economic effects of user-oriented delayed Wi-Fi offloading," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2684–2697, Apr. 2016.
[20] J.-H. Lim, K. Naito, J.-H. Yun, and M. Gerla, "Reliable safety message dissemination in NLOS intersections using TV white spectrum," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 169–182, Jan. 2018.
[21] H. Zhou *et al.*, "TV white space enabled connected vehicle networks: Challenges and solutions," *IEEE Netw.*, vol. 31, no. 3, pp. 6–13, May 2017.
[22] 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.
[23] S. Rezvani, N. Mokari, M. R. Javan, and E. A. Jorswieck, "Fairness and transmission-aware caching and delivery policies in OFDMA-based HetNets," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 331–346, Feb. 2020.
[24] X. Li, X. Wang, K. Li, Z. Han, and V. C. M. Leung, "Collaborative multi-tier caching in heterogeneous networks: Modeling, analysis, and design," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6926–6939, Oct. 2017.
[25] 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.
[26] G. Shan and Q. Zhu, "Sociality and mobility-based caching strategy for Device-to-Device communications underlying heterogeneous networks," *IEEE Access*, vol. 7, pp. 53777–53791, 2019.
[27] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, and X. Shen, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114–2127, Sep. 2018.
[28] K. Liu, L. Feng, P. Dai, V. C. S. Lee, S. H. Son, and J. Cao, "Coding-assisted broadcast scheduling via memetic computing in SDN-based vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2420–2431, Aug. 2018.
[29] 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. 10216–10226, Oct. 2019.
[30] C. Stefanovic, D. Vukobratovic, F. Chiti, L. Niccolai, V. Crnojevic, and R. Fantacci, "Urban Infrastructure-to-Vehicle traffic data dissemination using UEP rateless codes," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 94–102, Jan. 2011.
[31] L. Wang, H. Yang, X. Qi, J. Xu, and K. Wu, "ICast: Fine-grained wireless video streaming over Internet of intelligent vehicles," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 111–123, Feb. 2019.
[32] Y. Lin, B. Liang, and B. Li, "Data persistence in large-scale sensor networks with decentralized fountain codes," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Barcelona, Spain, May 2007, pp. 1658–1666.
[33] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, Jan. 1962.
[34] O. Kaiwartya *et al.*, "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.
[35] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.
[36] D. Fiems, B. Steyaert, and H. Bruneel, "Discrete-time queues with generally distributed service times and renewal-type server interruptions," *Perform. Eval.*, vol. 55, nos. 3–4, pp. 277–298, Feb. 2004.
[37] G. Zhang, T. Q. S. Quek, M. Kountouris, A. Huang, and H. Shan, "Fundamentals of heterogeneous backhaul design—Analysis and optimization," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 876–889, Feb. 2016.

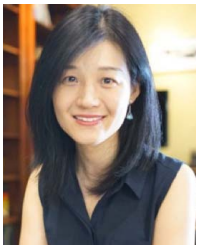
Huaqing Wu (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University 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.

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 pursuing 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.

Wenchao Xu (Member, IEEE) received the B.E. and M.E. degrees from Zhejiang University, Hangzhou, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2018. He is currently an Assistant Professor with the School of Computing and Information Sciences, Caritas Institute of Higher Education, Hong Kong. In 2011, he joined Alcatel Lucent Shanghai Bell Company Limited, where he was a Software Engineer of telecom virtualization. His interests include wireless communications with emphasis on resource allocation, network modeling, and AI applications.

Nan Cheng (Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, in 2016. He worked as a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2019. He is currently a Professor with the State Key Laboratory of ISN and with the School of Telecommunication Engineering, Xidian University, Shaanxi, China. His current research focuses on B5G/6G, space-air-ground integrated network, big data in vehicular networks, and self-driving system. His research interests also include performance analysis, MAC, opportunistic communication, and application of AI for vehicular networks.

Weisen Shi (Graduate Student Member, IEEE) received the B.S. degree from Tianjin University, China, in 2013, the M.S. degree from the Beijing University of Posts and Telecommunications, China, in 2016, and the Ph.D. degree from the University of Waterloo, Canada, in 2020. He is currently working with the Huawei Ottawa Research and Development Center, Canada. His interests include drone communication and networking, space-air-ground integrated networks, and vehicular networks.



Li Wang (Senior Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2009. She held visiting positions with the School of Electrical and Computer Engineering, Georgia Tech, Atlanta, GA, USA, from December 2013 to January 2015, and with the Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden, from August 2015 to November 2015 and July 2018 to August 2018. She is currently a Full Professor with the School of Electronic Engineering, BUPT, where she also heads the High Performance Computing and Networking Laboratory. She is also a member of the Key Laboratory of the Universal Wireless Communications, Ministry of Education, China, and the Associate Dean of the School of Software Engineering, BUPT. She has authored or coauthored almost 50 journal articles and two books. Her current research interests include wireless communications, distributed networking and storage, vehicular communications, social networks, and edge AI. She was a recipient of the 2013 Beijing Young Elite Faculty for Higher Education Award, best paper awards from several IEEE conferences, e.g., the IEEE ICCC 2017, the IEEE GLOBECOM 2018, the IEEE WCSP 2019, and so forth. She was also a recipient of the Beijing Technology Rising Star Award in 2018 and was selected as a Distinguished Young Investigator by the China Academy of Engineering in 2018. She also serves on the Editorial Board for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE ACCESS, *Computer Networks*, and *China Communications*. She was the Symposium Chair of IEEE ICC 2019 on Cognitive Radio and Networks Symposium and the Tutorial Chair of the IEEE VTC 2019-fall. She also chairs the Special Interest Group (SIG) on Social Behavior Driven Cognitive Radio Networks for the IEEE Technical Committee on Cognitive Networks. She has served on TPC of multiple IEEE conferences, including the IEEE Infocom, Globecom, International Conference on Communications, the IEEE Wireless Communications and Networking Conference, and the IEEE Vehicular Technology Conference in recent years.

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 focuses on network resource management, wireless network security, the Internet of Things, 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. He received the R. A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) presents in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society, and the Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). 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 has 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, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the elected IEEE Communications Society Vice President of Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of the IEEE Fellow Selection Committee. He was/is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, the IEEE NETWORK, *IET Communications*, and *Peer-to-Peer Networking and Applications*.