An Information-centric In-network Caching Scheme for 5G-enabled Internet of Connected Vehicles

Cong Wang, Chen Chen, *Senior Member, IEEE*, QingQi Pei, *Senior Member, IEEE*,
Zhiyuan Jiang, *Member, IEEE*, Shugong Xu, *Fellow, IEEE*

Abstract—With the increasing on-board demand for intelligent connected vehicles (ICVs), the fifth-generation (5G) wireless systems are being massively utilized in vehicular networks. As an essential component, content retrieval in the ICV provides a basis for vehicle-to-vehicle or vehicle-to-infrastructure data interaction for many applications. However, content access is still subject to performance degradation due to congested communication channels, diverse requests patterns, and intermittent network connectivity. To mitigate these issues, in-network caching in 5G-enabled ICV has been leveraged to benefit content access by allowing edge nodes to store content for data generators. In this paper, we propose a proactive in-network caching scheme to support various provisions of data sharing in the ICVs by exploring the advantages of information-centric networks (ICN). We first divide each on-board service into several content units. Then, we place these units at the ICV and small cell base stations (SBSs) to reduce the content retrieval delay, further model the proposed system as an integer nonlinear program (INLP) and attain the optimal QoE (Quality of Experience) by placing content units at appropriate cache entities. Finally, we verify the effectiveness and correctness of our proposed model through extensive simulations.

Index Terms—5G, Vehicular Networks, Content Placement, QoE Enhancement.

1 Introduction

A RAPIDLY increasing attention over autonomous driving technology has led to the explosive growth of the research on Intelligent Connected Vehicles (ICVs), i.e., transportation terminals that integrate the Internet of Vehicles (IoV) with intelligent telecommunication systems. ICVs are equipped with advanced onboard units (OBUs), such as sensors, controllers, and actuators, to make vehicles perceive their surroundings, improve driving safety, enhance the driving experience, increase traffic efficiency, and thus achieve autonomous driving [1].

Recently, the incoming fifth-generation (5G) [2] wireless communication systems have been arising as a promising technique to provide users with better quality of services (QoS) for a variety of data transmission applications [3]. Besides ICV, roadside infrastructures and other electronic devices, such as pedestrian handheld equipment, are also endowed with communication capabilities, thus giving rise to the prosper of 5G-Vehicle to Everything (5G-V2X). Accompanying with the progressive 5G-V2X technology and advanced autonomous driving development, passengers or drivers in ICVs are pursuing more diversified in-vehicle applications with better quality of experience (QoE) during the journey. Many wireless multimedia traffic, such as 3D digital landscape maps, breaking news, popular video or advertisement, traffic congestion, or emergency warning, are frequently requested, leading to a surge in data volume emanating. Therefore, the continually downloaded data witnesses a great information transmission redundancy [4] and adds a heavy burden to backhaul links besides experiencing a long content retrieval delay. As a result, the growing mobile data traffic flow in vehicular networks may lead to communication link congestion and further degrade information transmission performance [5]. To address these issues, in-network caching [6], i.e., storing the required content at the edge devices, has been proposed to provide significant gains in the context of content redundancy.

In the 5G framework, small cell base stations (SBSs) are usually densely deployed with cache capacity, enhanced communication, and resource allocation capabilities [7]. That lays a solid foundation for leveraging in-network caching to tackle the problems faced when transmitting data in the ICVs. The users in a cache-enabled vehicular network can get the required services from its cache storage or nearby nodes, instead of communicating with a remote content server [8], [9]. That can significantly reduce the delay when accessing contents as well as the backhaul burden [10]. On a closer inspection, a recently emerged promising communication network architecture, i.e., information-centric network (ICN), provides a context-aware network substrate that uses the content name as the identifier for

This work was supported by the National Key Research and Development Program of China (2019YFE0196600), the National Natural Science Foundation of China (62072360, 62001357, 61672131, 61901367), the key research and development plan of Shaanxi province(2021ZDLGY02-09, 2020JQ-844), the key laboratory of embedded system and service computing (Tongji University)(ESSCKF2019-05), Ministry of Education, Xi'an Science and Technology Plan(20RGZN0005) and the Xi'an Key Laboratory of Mobile Edge Computing and Security (201805052-ZD3CG36).

C. Wang, C. Chen(Corresponding Author) and Q. Pei are with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China (e-mail: cwang_96@stu.xidian.edu.cn; cc2000@mail.xidian.edu.cn; qqpei@mail.xidian.edu.cn.)

Z. Jiang and S. Xu are with the Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China (e-mail: jiangzhiyuan@shu.edu.cn; shugong@shu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3137219, IEEE Transactions on Mobile Computing

multifarious service requests [11]. Subscribers in the ICN can get the desired service by identifying specific content names while publishing service requirements instead of IP addresses. Any node that receives a service request can directly respond to the inquiring node, provided that there is a corresponding content block in the current nodes' storage, which avoids requesting service to the remote content server. By this means, a more effective data transmission process is achieved.

Furthermore, the terminal in vehicular networks, namely ICV, will not remain stationary in most cases, causing intermittent wireless connectivity (may result in packet loss) to affect user experience further. Contrary to the node in other mobile networks, the movement of ICV is often limited by road layout. In the design of caching strategy, we can make full use of this unique mobility feature, place the content units to SBS that ICV is trying to access in the upcoming request along the driving direction. From another perspective, content request uncertainty also poses a significant challenge to cache allocation. Capturing the characteristics of ICV request patterns can usher instructive guidance for cache allocation decisions.

In this paper, we investigate the QoE-optimal content caching in ICVs through cache placement in a 5G-enabled Named Data Network (NDN) framework, which is a realization of the ICN paradigm [12]. In the NDN, the requester will issue an interest packet while requiring the desired files. Also, there is a proprietary data structure, namely Content Store (CS), dedicated to storing data, which caches the content that may be requested, thereby making content placement easy to implement.

The contributions of this paper can be summarized as:

- We divide the complete vehicular data package into several units of uniformed size and sequentially store them in inhomogeneous cache carriers while considering the special mobile characteristics of the ICV. In this way, ICVs can request content from ICV/SBS on a time-sharing basis, and the interest packet can be issued during the units' playing phase, further improving the requesters' QoE.
- We devise the QoE optimization problem based on the V2V, V2B (Vehicle-to-SBS), and V2N (Vehicle-to-Network) transmission delays, besides the playtime or read-out time of files, under cache capacity limitations. In the process of data transmission, we consider the group movement characteristics of ICVs.
- We decouple the QoE-optimal problem into two sub-problems, namely power allocation problem and content placement problem, to acquire the resource allocation scheme that achieves the optimal QoE. In this way, we obtain the optimum cache decision and power distribution of the system.

The rest of this paper is organized as follows. In Section 2, we review the existing studies on content caching in the IoV domain. Then, we introduce the system model in Section 3 and problem formulation in Section 4. These are followed by simulation results of the proposed method in Section 5. Lastly, we provide our conclusions and future research directions in Section 6 to finalize the paper.

2 Related Works

In recent years, several studies have utilized caching within different communication frameworks by designing novel strategies to improve the content delivery performance of systems [9], [13], [14]. In this section, we briefly summarize the current literature on caching schemes for the ICVs and evaluate them from different perspectives.

2.1 Vehicular Caching Scheme

In vehicular caching schemes, vehicles can act as content providers to serve requesters via V2V communications.

Zhang *et al.* [15] explored the interaction behaviors between mobile users and vehicles and proposed an online caching system for content-centric vehicular networks. They investigated the factors impacting caching performance, e.g., the distribution of mobile users and vehicles, the communication mode of heterogeneous nodes, and content popularity, to efficiently perform vehicular caching towards minimizing energy consumption of the whole system.

In [14], the authors designed a cooperative vehicular caching strategy to improve the content acquisition performance by combining the nodes' social attributes and mobility patterns in the vehicular content network. The exhaustive simulations showed that this proposal outperformed DPC [16], LDCC [17], and DAC [18] in terms of different Key Performance Indicators (KPIs).

Our previous work [19] was also dedicated to developing a vehicular caching system called PICS. While making cache decisions, PICS considers several factors affecting in-vehicle caching performance, such as content popularity, vehicle preference, and node selfishness, to alleviate the burden of base stations (BSs).

In the strategies summarized above, the roadside units (RSUs) or BSs were assumed to be the content library, which could hold all files that might be requested. However, the cache space of these entities was not infinite, and the request to the core network was always required in more realistic environments, causing longer content retrieving delay.

2.2 BS/RSU-assisted Caching Scheme

In BS/RSU-assisted caching schemes, BSs/RSUs are used as cache carriers to provide content units for mobile users.

Su *et al.* [20] proposed an edge caching strategy considering vehicle request pattern, limited cache capacity, and RSUs' cooperation to cache the popular contents. Novelly, they also considered the retrieving delay in V2V communications while making RSU cache decisions. Extensive simulations verified the system's effectiveness.

Zhang *et al.* [21] investigated the demand and mobility patterns of autonomous vehicles in the 5G-ICN and developed a proactive caching approach for BSs, RSUs, and routers. In this proposal, videos were divided into multiple chunks, where hierarchical nodes could cache different chunks. However, they did not consider V2V communications.

Song *et al.* [22] developed an RSU caching strategy for short videos in vehicular networks. This proposal takes QoE of short videos as the ultimate objective while regarding the users' requests as a class-based model. The novel part of this

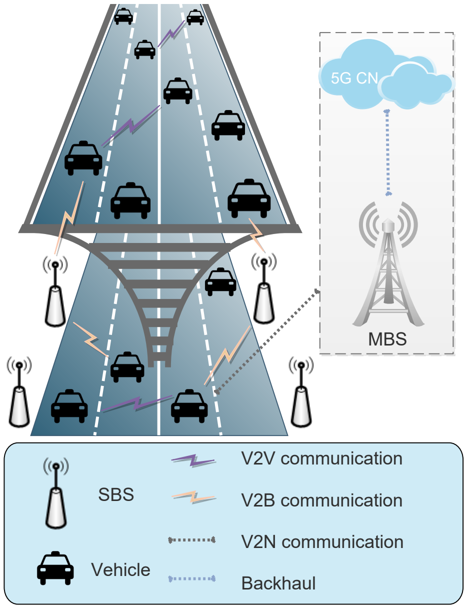


Fig. 1: The proposed system in a 5G-enabled ICV scenario.

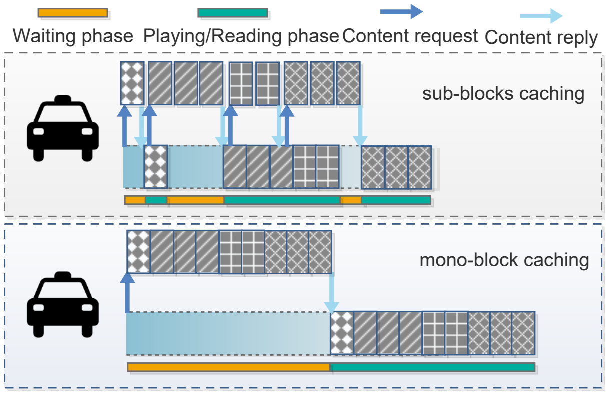


Fig. 2: Comparison of different cache models.

work was the design of a deep reinforcement learning-based solution to achieve the optimal solution of cache allocation.

In general, the cache capacity of each cache entity is finite. The increasing interactive data volume makes the cache space insufficient to store all content [23], therefore, a reasonable content cache selection mechanism becomes indispensable. Furthermore, the dynamic characteristics of the ICV and the diversity of ICV requests make content caching a challenging task. However, since the SBSs are ultra-densely distributed along roadsides and are usually interconnected, cooperative caching becomes easier to implement. That further enables ICVs to be served by multiple SBSs, develops the caching diversity [9], and thus improves the utilization of cache storage. Different from [9], we study inhomogeneous caching and power allocation strategies for the ICVs based on a 5G-enabled system with a particular focus on the communication between entities, i.e., V2V, V2B, and V2N.

3 System Model

This section describes the system model envisioned for cache management in 5G-enabled ICV networks as well as the heterogeneous communication model for different entities. For the convenience of reading, we list the symbols used in this paper in Table. 1.

3.1 Cache Management Architecture for the 5G-enabled ICV Network

Let us consider a network architecture as depicted in Fig. 1 that includes a MBS and connects to the 5G core network (CN) or remote content server, with K SBSs (densely deployed along the road), i.e., $\mathbf{B} = \{B_1, B_2, ..., B_k, ...B_K\}$, and N ICVs (moving on a road region), i.e., $\mathbf{U} = \{U_1, U_2, ..., U_n, ...U_N\}$. Suppose that the ICVs are built with antennas to send or receive data, and SBSs are equipped with antenna arrays to provide ubiquitous communication coverage. Both SBSs and ICVs have storage space to cache several videos, audio files, or text pieces.

A video/audio/text content can be encoded into several sub-units, which can be stored in different ICVs and SBSs. Suppose that each ICV caches several beginning content units of a service with a specific probability, and the SBSs cooperatively cache content units in their storage. When requiring a content service (e.g., a video), as depicted in the upper side of Fig. 2, user U_n ($U_n \in \mathbf{U}$) will first try to look up the beginning content that is stored in its OBU or request it from its neighbor. During the content playing/read-out phase, U_n will then ask for the subsequent segments (in this paper, we use *unit* and *segement* alternately, they represent the same meaning) from the SBSs. Similarly, U_n will play the newly-received segments simultaneously, requesting the following pieces. Compared to mono-block caching (as depicted in the lower side of Fig. 2), i.e., caching the complete file in one node, sub-block caching can shorten the waiting phase. If U_n cannot fetch the entire content service through its OBU or the SBSs, a content access process is required by the MBS. As such, users who can fetch content from the original content servers will experience longer latency, resulting in QoE degradation [24]. The specific time calculation details will be described in Section 3.3.

3.2 File Request Features and File Transmission Delay

3.2.1 File Request Features

Assume that the content library, i.e., files set that might be requested, contains M files $\mathbf{F} = \{F_1, F_2, ..., F_m, ..., F_M\}$. However, the expected file preferences of different ICVs are usually different. For example, the ICVs in a region near complex traffic zones tend to request traffic condition-related services, while the terminals near a supermarket may ask for discount information. To deal with the regionality, we first divide the road section into different regions $R = \{R_1, R_2, ..., R_Y\}$ and then partition the file library set $\mathbf{F}$ into several subsets, i.e., $\mathbf{F} = \mathbf{R}_1 \cup \mathbf{R}_2 \cup ... \cup \mathbf{R}_y \cup ... \cup \mathbf{R}_Y$, where subset $\mathbf{R}_y \subseteq \mathbf{F}$ contains the files that the ICVs in R_y trying to access. For each region, we assume that content acquisition is directional, i.e., ICVs tend to show solicitude for the relevant information ahead. Therefore, we plan to store content units in the SBS sequentially according to ICVs' driving direction.

TABLE 1: Notation table.

Notation	Meaning	Remarks
B_k	$k-$th SBS $(k = 1, 2, ..., K)$	**B**= $\{B_1, B_2, ..., B_k, ...B_K\}$
U_n	$n-$th ICV $(n = 1, 2, ..., N)$	**U** = $\{U_1, U_2, ..., U_n, ...U_N\}$
F_m	$m-$th file $(m = 1, 2, ..., M)$	**F** = $\{F_1, F_2, ..., F_m, ...F_M\}$
$\mathbf{R}_y$	Files that the ICVs in region R_y trying to access	$\mathbf{F} = \mathbf{R}_1 \cup \mathbf{R}_2 \cup ... \cup \mathbf{R}_y \cup ... \cup \mathbf{R}_Y$
$R_{n,V2B_k}$	Content transmission rate between U_n and B_k	Equal to $W_{n,k} \log_2(1 + \text{SINR}^B_{n,k})$
$R_{n,V2V_p}$	Content transmission rate between U_n and U_p	Equal to $W_{V2V} \log_2(1 + \text{SINR}^V_{n,p})$
$P^{U/m}_{n,k}(U^m_{n,k})$	Propagation delay (distance) of upload link between U_n and B_k	$P^{U/m}_{n,k} = U^m_{n,k}/c$
$P^{D/m}_{n,k}(D^m_{n,k})$	Propagation delay (distance) of download between U_n and B_k	$\mathcal{P}^{X/D/m}_{n,k}$ and $D^{mX}_{n,k}$ indicate items in case X
l_m	The number of content units of F_m that might be cached in the ICV	N.A.
p_m	Cache probability of the FlS	N.A.
α^m_k	The number of content units that cached in B_k	$\mathbf{a}_m = \{\alpha^m_1, \alpha^m_2, ..., \alpha^m_K\}$
A	The size of each content unit	N.A.
G_X	Probability of different cases in Stage 1	G_X indicates the probability of case X
$m(k)$	The size of content units served by B_k	$k \geq 2$
$m_X(1)$	The size of content units served by B_1 in case X	$m(0)$ indicates the units' size served by ICV
β	The duration that each content unit can be played	Video, audio, and text have different β values
$T^m_{n,k}$	The content transmission delay between U_n and B_k	$T^{X/m}_{n,k}$ indicates the delay in case X
T^m_{n,U_p}	The content transmission delay between U_n and U_p	N.A.
$T^{R-S}_{n,k}$	The request transmission delay between U_n and B_k	N.A.
T^{R-V}_{n,U_p}	The request transmission delay between U_n and U_p	N.A.
$\mathcal{T}^m_{n,k}$	The delay of U_n to retrieve content from B_k	$\mathcal{T}^{X/m}_{n,k}$ indicates the delay in case X
$\mathbb{T}^m_{n,k}$	The time when U_n received the units from B_k	$\mathbb{T}^{X/m}_{n,k}$ indicates the time in case X
$\mathcal{D}^m_{n,k}$	The play duration time of units that accessed from B_k	$\mathcal{D}^{X/m}_{n,k}$ indicates the duration time in case X
$\mathbb{D}^m_{n,k}$	The play end time of units that accessed from B_k	$\mathbb{D}^{X/m}_{n,k}$ indicates the time in case X
$\mathcal{W}^m_{n,k}$	The waiting time in stage k	$\mathcal{W}^{X/m}_{n,k}$ indicates the time in case X
C_v	The maximum V2V communication distance	N.A.
$q_{R_y(m)}$	The request probability of file m in region R$_y$	N.A.
$\mathcal{C}_k$	Maximum cache capacity of SBS B_k	N.A.
P_j	Transmission power allocated to SBS B_j	N.A.

To model content popularity, we arrange the data that might be requested in region R_y in a popularity-descending order. We divide the files in $\mathbf{R}_y$ into three categories, namely entertainment information, advertising information, and traffic information. These classes cover different file types, including video $\mathbf{R}^v_y$, audio $\mathbf{R}^a_y$ and text $\mathbf{R}^t_y$ ($\mathbf{R}_y$=$\mathbf{R}^v_y$∪$\mathbf{R}^a_y$∪$\mathbf{R}^t_y$). In this paper, we focus more on video and audio files, since the size of text information is relatively small; therefore, they do not need to be divided into smaller content units. Undoubtedly, text files can also use the caching scheme proposed in this paper, but the performance improvement is not as obvious as videos or audios. We assume that the request probabilities of different file types are different. Let Q_v, Q_a, and Q_t denote the request probability of video, audio, and text ($Q_v + Q_a + Q_t = 1$), respectively. To deal with the differentiated request patterns in different regions, we assume that the request probability of each file in each category is similar for the ICVs in a certain region, and the request probability of file F_m in $\mathbf{R}_y$ can be modeled as Zipf distribution function [25]. Taking video files as an example, the probability of a video file F^v_m being requested is:

$$q_{F^v_y(m)} = \frac{(r^v_m)^{-\rho_v}}{\sum_{i=1}^{|\mathbf{R}^v_y|}(r^v_i)^{-\rho_v}}, \tag{1}$$

where $q_{F^v_y(m)}$ is the popularity of the r^v_m-th file in video class, and ρ_v is the parameter of Zipf distribution for video files. In this way, we can calculate the request probability of a video file as $q_{R_y(m)} = q_{F^v_y(m)} \cdot Q_v$. The request probability of other content categories can be calculated in the same way. In the subsequent derivation process, we take file F_m as a representative to finish the deduction.

3.2.2 File Access Delay

There are two components considered in file access delay: transmission delay and propagation delay.

Transmission Delay: If requester U_n is served by SBS B_k, the content transmission rate can be calculated by:

$$R_{n,V2B_k} = W_{n,k} \log_2(1 + \text{SINR}^B_{n,k}), \tag{2}$$

where $W_{n,k}$ is the available bandwidth between U_n and B_k, and SINR$^B_{n,k}$ denotes the received Signal to Interference plus Noise Ratio (SINR) between U_n and B_k [26], i.e.:

$$\text{SINR}^B_{n,k} = \frac{|\mathbf{h}_{n,k}\mathbf{w}_{n,k}|^2}{\sum_{i \neq m}^{N} |\mathbf{h}_{i,k}\mathbf{w}_{i,k}|^2 + N_0}, \tag{3}$$

where $\mathbf{h}_{n,k}$ is the channel vector of B_k to U_n with $1 \times N_T$, N_T is the number of transmitting antennas in the SBS, $\mathbf{w}_{n,k} = c\sqrt{P_k}\mathbf{h}^{\dagger}_{n,k}$ is the precoding vector from B_k to

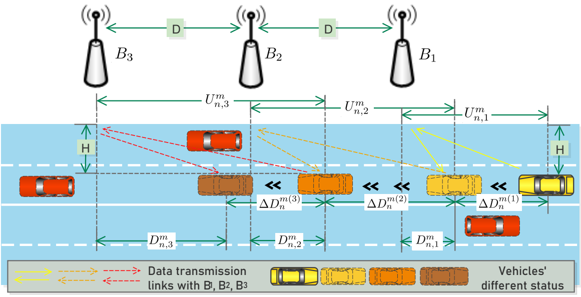


Fig. 3: Calculation of transmission distance.

U_n, N_0 is the noise power, $\mathbf{h}^{\dagger}_{n,k}$ is the k-th column of $\mathbf{H}^{\dagger}_n$ ($\mathbf{H}^{\dagger}_n = \mathbf{H}^H_n(\mathbf{H}_n\mathbf{H}^H_n)^{-1}$), and $\mathbf{H}_n = [\mathbf{h}^T_{n,1}, \mathbf{h}^T_{n,2}, ..., \mathbf{h}^T_{n,k}]$. Substituting $\mathbf{w}_{n,k}$ into (3), we get:

$$\mathrm{SINR}^B_{n,k} = \frac{|c\sqrt{P_k}\mathbf{h}_{n,k}\mathbf{h}^{\dagger}_{n,k}|^2}{\sum^N_{i\neq n}|c\sqrt{P_k}\mathbf{h}_{i,k}\mathbf{h}^{\dagger}_{i,k}|^2 + N_0}. \tag{4}$$

Similarly, the explicit file transmission rate can be calculated by substituting (4) into (2).

Due to the dynamic nature of vehicles, the retransmission times must be taken into account. We define the transmission delay between U_n and B_k concerning file F_m of a specific served size as $T^m_{n,k} = [r_t \cdot m(k)] \cdot m(k)/R_{n,V2B_k}$, where $r_t \cdot m(k)$ is the expected retransmission time, and $m(k)$ is the size of units to be transmitted. In general, retransmission time is related to the signal-to-noise ratio (SNR), bit error rate, and packet length. However, for simplicity, we assume that the number of retransmissions is positively related to the length of the transmitted content. Also, there will be a transmission delay $T^{R-S}_{n,k} = r_t \cdot s^2_r/R_{n,V2B_k}$ when the request is issued, where s_r is the size of the interest packet.

If requester U_n is served by another ICV, U_p ($U_p \in \mathbf{U}$), the achievable data transmission rate for radio access will be:

$$R_{n,V2V_p} = W_{V2V}\log_2(1 + \mathrm{SINR}^V_{n,p}), \tag{5}$$

where W_{V2V} is the bandwidth allocated to V2V communication, $\mathrm{SINR}^V_{n,p} = \frac{P_T d^{-\alpha}_{n,p}}{\sigma^2+I}$ denotes the SINR between U_n and U_p, where P_T is the transmission power of the associated content provider U_p, α is the path loss exponent, $d_{n,p}$ is the transmission distance between U_n and U_p, σ^2 is the addictive noise power, and I is the intercell interference. Similar to V2B communication delay, we can define the transmission delay between U_n and U_p as $T^m_{n,U_p} = r_t \cdot m^2(0)/R_{n,V2V_p}$, where $m(0)$ is the size of content segments that are cached in ICVs. Similarly, the transmission delay of the interest packet between vehicles is $T^{R-V}_{n,U_p} = r_t \cdot s^2_r/R_{n,V2V_p}$.

Propagation Delay: The propagation delay is proportional to the distance between the requester and the provider. In general, requester U_n will send an interest packet to the SBS after issuing requests to other ICVs for the desired content unit. As shown in Fig. 3, the propagation distance of the uplink transmission between U_n and B_1 is $U^m_{n,1}$ (omitting the longitudinal distance H). Denoting D as the SBSs' deployment interval, the uplink propagation distance between U_n and B_k can be expressed as:

$$U^m_{n,k} = U^m_{n,1} + (k-1)D - \sum^{k-1}_{j=1}\Delta D^{m(j)}_n, \tag{6}$$

where $\Delta D^{m(j)}_n = \mathcal{T}^m_{n,j} \cdot \bar{v}_n$, denoting the distance that U_n moves during the j-th content request process ($\mathcal{T}^m_{n,j}$) and $\bar{v}_n$ being the mean velocity in this process. The propagation delay of this upload phase is $\mathcal{P}^{U/m}_{n,k} = U^m_{n,k}/c$, where c is the propagation rate of the electromagnetic wave. Upon receiving the interest packet, B_1 will serve U_n with the required content segment. The propagation distance of each content segment, i.e., download link, can be summarized as:

$$D^m_{n,k} = U^m_{n,1} + (k-1)D - \sum^{k}_{j=1}\Delta D^{m(j)}_n. \tag{7}$$

Similarly, the propagation delay of this download phase can be calculated as $\mathcal{P}^{D/m}_{n,k} = D^m_{n,k}/c$. In particular, for special case X (X = A,B, and C, which will be specified in Section 3.3), $\mathcal{P}^{X/D/m}_{n,k} = D^{m_X}_{n,k}/c$, and $D^{m_X}_{n,k} = U^m_{n,1} + (k-1)D - \sum^k_{j=1}\Delta D^{m_X(j)}_n$.

Considering that data acquisition in V2V often requires close-range communications, the propagation delay is relatively small, which is far less than the transmission delay. Therefore, we neglect this item in V2V communications. Furthermore, the movement of ICVs during the V2V request is also ignored. It is worth noting that, in this proposal, we set the initial time t_0 as the moment when U_n sends a request to the first SBS, B_1.

3.3 The Request Model

A consumer in a content-oriented network will issue an interest packet identified by a unique name while asking for specific content. When an intermediate ICV/SBS receives such a packet and finds some fragments in its storage that are corresponding to the content requested, it will send back the content duplicate through the incoming interface.

Supposing that a complete file can be encoded into several sub-segments, and each ICV caches the FlS (First l_m content Segments) of file F_m with probability p_m, SBS B_1 caches the beginning $\alpha^m_1(\alpha^m_1 \geq l_m)$ content segments, SBS B_2 caches the subsequent α^m_2 segments, and so on, SBS B_k stores the α^m_k units that follow the previous α^m_{k-1} units. Denoting A as the size of each segment, the size of the content units $m(k)$ that served by an ICV ($k = 0$) or $B_k(k \geq 1)$ is supposed to be:

$$m(k) = \begin{cases} A \cdot l_m & i = 0, \\ A \cdot \alpha^m_k & i = 1, 2, 3, ..., K. \end{cases} \tag{8}$$

When service is demanded, requester U_n first tries to request the FlS via V2V homogeneous transmission, then asks for the subsequent segment from the SBSs. If the size of the accessed content, i.e., $\sum^{k=K}_{k=1} m(k)$, is less than the file size s_m, requester U_n will fetch the remaining segments from the MBS as shown in Fig. 4. In this subsection, we divide the content retrieval process into several stages and describe them in detail.

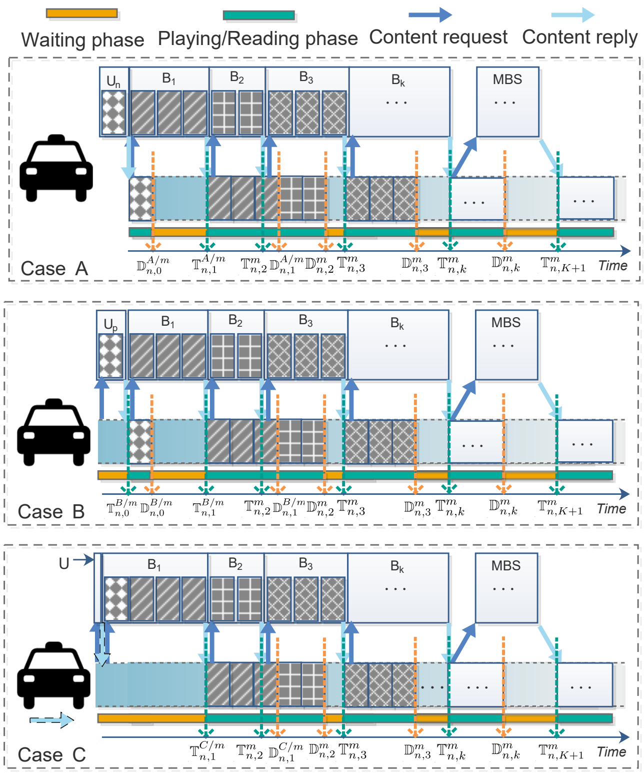


Fig. 4: Content request process in different cases.

3.3.1 Stage 1

Stage 1 indicates the content access process of the beginning α_1^m segments, where α_1^m is the number of content units stored in the first SBS B_1. This stage includes two steps: i) data acquisition via probabilistic V2V communication, ii) content access between B_1 and the requester.

In general, there are three different cases in *Stage 1*:

Case A: The FlS of F_m is cached in the local storage of U_n, i.e., $U_n \in \mathbf{C}_m$, where $\mathbf{C}_m$ denotes the collection of ICVs that cached the FlS of file F_m. The probability of Case A occurring is $G_A = p_m$. The content request process in *Stage 1* follows the steps below:

Step 1: Requester U_n retrieves the FlS of F_m from the local cache, the waiting time of which is $\mathcal{W}_{n,0}^{A/m} = 0$.

Step 2: During the content playing phase of the FlS, the requester tries to access the subsequent units of B_1.

In Case A, the number of content units that is served by SBS B_1 is $m_A(1) = m(1) - m(0)$, and the delay to retrieve content from B_1 is:

$$\mathcal{T}_{n,1}^{A/m} = T_{n,1}^{R-S} + \mathcal{P}_{n,1}^{U/m} + T_{n,1}^{A/m} + \mathcal{P}_{n,1}^{A/D/m}, \qquad (9)$$

where $T_{n,1}^{R-S}$, $\mathcal{P}_{n,1}^{U/m}$ and $T_{n,1}^{A/m}$, $\mathcal{P}_{n,1}^{A/D/m}$ denote the transmission and propagation delays of the interest packet and the data packets ($T_{n,1}^{A/m} = r_t \cdot m_A^2(1)/R_{n,V2B_1}$), respectively. To facilitate the following representation, we define $T_{n,1}^{A/m} \triangleq f[m_A(1)]$.

The play duration time of the FlS of F_m is $\mathcal{D}_{n,0}^{A/m} = m(0) \cdot \beta \triangleq g[m(0)]$, where β is the duration that each unit can be played. The waiting time during this process is:

$$\mathcal{W}_{n,1}^{A/m} = \max\{\mathbb{T}_{n,1}^{A/m} - \mathbb{D}_{n,0}^{A/m}, 0\} \triangleq \max\{w_{n,1}^{A/m}, 0\}, \qquad (10)$$

where $\mathbb{T}_{n,1}^{A/m}$ is the time when the content units in SBS B_1 are transmitted successfully, and $\mathbb{D}_{n,0}^{A/m}$ is the play end-time of the previous content units. In this sub-case, i.e., Case A of *Stage 1*, $\mathbb{T}_{n,1}^{A/m} = \mathcal{T}_{n,1}^{A/m}$ and $\mathbb{D}_{n,0}^{A/m} = \mathcal{D}_{n,0}^{A/m}$.

Case B: The FlS of F_m is not cached in U_n but cached in other ICVs that are located in U_n's V2V range, i.e., $\exists U_p \in \mathbf{C}_m \backslash \{U_n\} : d_{p,n} \leq C_v$, where C_v is the maximum V2V communication range. The probability of Case B is $G_B = 1 - G_A - (1-p_m)^{\mathbb{E}[N_n]+1}$, where $\mathbb{E}[N_n]$ denotes the expected number of ICVs that are located in the communication range of U_n, which is subject to the actual distribution of the ICV group.

Step 1: In this case, the FlS of F_m is served by the neighbor of U_n. The waiting time in this step is equal to the transmission delay, i.e., $\mathcal{W}_{n,0}^{B/m} = \mathbb{T}_{n,0}^{B/m} = T_{n,U_p}^{R-V} + T_{n,U_p}^{m}$, where T_{n,U_p}^{R-V} and T_{n,U_p}^{m} denote the transmission delay of interest and data packets, respectively.

Step 2: Similar to Case A, during the content playing phase of the FlS, the requester tries to access the subsequent units of B_1.

In Case B, the size of content units served by SBS B_1 is $m_B(1) = m(1) - m(0)$, and the delay to retrieve content from B_1 is:

$$\mathcal{T}_{n,1}^{B/m} = T_{n,1}^{R-S} + \mathcal{P}_{n,1}^{U/m} + T_{n,1}^{B/m} + \mathcal{P}_{n,1}^{B/D/m}. \qquad (11)$$

The specific meanings of these items are the same as those of Case A. The play duration time of the FlS is $\mathcal{D}_{n,0}^{B/m} = g[m(0)]$, and the waiting time during this process is:

$$\mathcal{W}_{n,1}^{B/m} = \max\{\mathbb{T}_{n,1}^{B/m} - \mathbb{D}_{n,0}^{B/m}, 0\} \triangleq \max\{w_{n,1}^{B/m}, 0\}. \qquad (12)$$

In this sub-case, $\mathbb{T}_{n,1}^{B/m} = \mathcal{T}_{n,1}^{B/m}$ and $\mathbb{D}_{n,0}^{B/m} = \mathcal{D}_{n,0}^{B/m}$.

Case C: None of the requesters and their neighbors cached the FlS of F_m, i.e., $U_n \notin \mathbf{C}_m$ & $\nexists U_p \in \mathbf{C}_m \backslash \{U_n\} : d_{p,n} \leq C_v$. The probability of this case is $G_C = (1 - p_m)^{\mathbb{E}[N_n]+1}$.

Step 1: In this case, the FlS of F_m cannot be accessed via the V2V communication. The waiting time of this step is $\mathcal{W}_{n,0}^{C/m} = \text{RTT}$, where RTT is the round-trip time of the U_n's request.

Step 2: The requester tries to access the subsequent units of B_1.

In Case C, the number of content units of F_m that is served by SBS B_1 is $m_C(1) = m(1)$, and the delay to retrieve content from B_1 is:

$$\mathcal{T}_{n,1}^{C/m} = T_{n,1}^{R-S} + \mathcal{P}_{n,1}^{U/m} + T_{n,1}^{C/m} + \mathcal{P}_{n,1}^{C/D/m}, \qquad (13)$$

where $T_{n,1}^{C/m} = f[m(1)]$, and the specific meanings of the remaining items are the same as those of Case A. Since the requester cannot get the desired content from the ICV, the play duration time of the previous sequence is $\mathcal{D}_{n,0}^{C/m} = 0$. Thus, the waiting time in this case is:

$$\mathcal{W}_{n,1}^{C/m} = \max\{\mathbb{T}_{n,1}^{C/m} - \mathbb{D}_{n,0}^{C/m}, 0\} \triangleq \max\{w_{n,1}^{C/m}, 0\}, \qquad (14)$$

where $\mathbb{T}_{n,1}^{C/m}$ is the time when content units in SBS B_1 is transmitted successfully, and $\mathbb{D}_{n,0}^{C/m}$ is the play end-time of the previous content unit. In this sub-case, $\mathbb{T}_{n,1}^{C/m} = \mathcal{T}_{n,1}^{C/m}$, $\mathbb{D}_{n,0}^{C/m} = \mathcal{D}_{n,0}^{C/m}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3137219, IEEE Transactions on Mobile Computing

Combining the cases explained above, we can calculate the expected waiting time of the FIS as:

$$\begin{aligned}\mathcal{W}^m_{n,0} &= G_A \mathcal{W}^{A/m}_{n,0} + G_B \mathcal{W}^{B/m}_{n,0} + G_C \mathcal{W}^{C/m}_{n,0} \\ &= \frac{G_B r_t[s_r^2 + m^2(0)]}{R_{n,V2V_p}} + G_c \mathcal{W}^{C/m}_{n,0}.\end{aligned} \tag{15}$$

The waiting time of the first content segments is denoted as $\mathcal{W}^m_{n,1} = \max\{\mathbb{T}^m_{n,1} - \mathbb{D}^m_{n,0}, 0\} \triangleq \max\{w^m_{n,1}, 0\}$. Also, we have $\mathbb{T}^m_{n,1} = \mathcal{T}^m_{n,1}$ and $\mathbb{D}^m_{n,0} = \mathcal{D}^m_{n,0}$. Substituting the specific calculation of transmission time in different cases into $\mathcal{T}^m_{n,1}$, i.e., the expected time of the requester to receive content units from B_1, we get:

$$\mathcal{T}^m_{n,1} = G_A \mathcal{T}^{A/m}_{n,1} + G_B \mathcal{T}^{B/m}_{n,1} + G_C \mathcal{T}^{C/m}_{n,1}. \tag{16}$$

Similarly, the expected play duration time of the FIS can be calculated as:

$$\begin{aligned}\mathcal{D}^m_{n,0} &= G_A \mathcal{D}^{A/m}_{n,0} + G_B \mathcal{D}^{B/m}_{n,0} + G_C \mathcal{D}^{C/m}_{n,0} \\ &= (G_A + G_B) \cdot m(0) \cdot \beta.\end{aligned} \tag{17}$$

3.3.2 *Stage 2-K*

Stage k refers to the content access process between requester U_n and provider $B_k (k = 2, 3, ..., K)$.

During the content playing phase of the units retrieved from the previous SBSs, U_n will request the subsequent segments. The size of the segments that ICV U_n can get from B_k is $m(k), k = 2, 3, ..., K$, and the delay to retrieve content is:

$$\mathcal{T}^m_{n,k} = T^{R-S}_{n,k} + \mathcal{P}^{U/m}_{n,k} + T^m_{n,k} + \mathcal{P}^{D/m}_{n,k}, \tag{18}$$

where $T^{R-S}_{n,k}, \mathcal{P}^{U/m}_{n,k}$ and $T^m_{n,k}, \mathcal{P}^{D/m}_{n,k}$ denote the transmission and propagation delays of the interest packet and the content units that are stored in B_k ($T^m_{n,k} = f[m(k)]$), respectively. The waiting time in Stage k is

$$\mathcal{W}^m_{n,k} = \max\{\mathbb{T}^m_{n,k} - \mathbb{D}^m_{n,k-1}, 0\} \triangleq \max\{w^m_{n,k}, 0\}, \tag{19}$$

where $\mathbb{T}^m_{n,k}$ is the time when the content units in SBS B_k are transmitted successfully, i.e.:

$$\mathbb{T}^m_{n,k} = \mathcal{T}^m_{n,1} + \mathcal{T}^m_{n,2} + ... + \mathcal{T}^m_{n,k}, \tag{20}$$

and the play end-time of the content units accessed from B_{k-1} is:

$$\mathbb{D}^m_{n,k-1} = \sum_{i=0}^{k-1} \mathcal{D}^m_{n,i} + \sum_{i=1}^{k-1} \mathcal{W}^m_{n,i}, \tag{21}$$

where $\mathcal{D}^m_{n,k} = g[m(k)]$ denotes the play duration time of the content units retrieved from B_k.

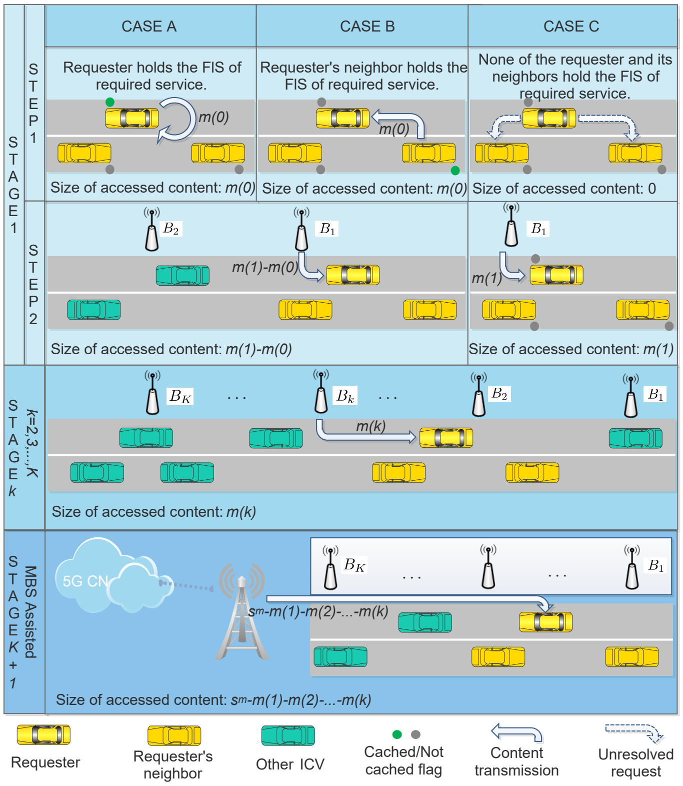
3.3.3 *Stage K+1*

Afterward, if the requester still cannot retrieve the complete file from SBSs, a service request will be sent to the MBS. The size of units that are served by the MBS is $m(K+1) = s_m - \sum_{i=0}^{K} m(i)$, and the waiting time during the current process is:

$$\mathcal{W}^m_{n,K+1} = \max\{\mathbb{T}^m_{n,K+1} - \mathbb{D}^m_{n,K}, 0\} \triangleq \max\{w^m_{n,K+1}, 0\}. \tag{22}$$

The file access time of *Stage K+1* and play end-time of its previous segments can be calculated as:

$$\mathbb{T}^m_{n,K+1} = \mathcal{T}^m_{n,1} + \mathcal{T}^m_{n,2} + ... + \mathcal{T}^m_{n,K+1}, \tag{23}$$



CASE A
CASE B
CASE C
STEP 1
STAGE 1
Requester holds the FIS of required service.
Requester's neighbor holds the FIS of required service.
None of the requester and its neighbors hold the FIS of required service.
m(0)
m(0)
Size of accessed content: m(0)
Size of accessed content: m(0)
Size of accessed content: 0
STEP 2
B2
B1
B1
m(1)-m(0)
m(1)
Size of accessed content: m(1)-m(0)
Size of accessed content: m(1)
STAGE k
k=2,3,...,K
BK
Bk
B2
B1
m(k)
Size of accessed content: m(k)
STAGE K+1
MBS Assisted
5G CN
BK
B1
sm-m(1)-m(2)-...-m(k)
Size of accessed content: sm-m(1)-m(2)-...-m(k)
Requester
Requester's neighbor
Other ICV
Cached/Not cached flag
Content transmission
Unresolved request

Fig. 5: Content access details in different stages.

$$\mathbb{D}^m_{n,K} = \sum_{i=0}^{K} \mathcal{D}^m_{n,i} + \sum_{i=1}^{K} \mathcal{W}^m_{n,i}, \tag{24}$$

where the required content access delay in *Stage K+1* is $\mathcal{T}^m_{n,K+1} = D_{bh} \cdot m(K+1)$, the backhaul latency is D_{bh}, and $\mathcal{D}^m_{n,K} = g[m(K)]$. Fig. 5 shows the content access details in different stages.

4 Problem Formulation and Solution

In this section, we first present the problem formulation of the QoE maximization. Then, using the optimization objective and the decision variables presented, we customize a solution for the formulated problem.

4.1 Problem Formulation

The QoE of a content request is defined as follows:

$$\text{QoE}_m = \mathcal{R}\Big(\sum_{i=1}^{K+1} \int_{\mathbb{D}^m_{n,i-1}}^{\max\{\mathbb{D}^m_{n,i-1}, \mathbb{T}^m_{n,i}\}} Q(t)\text{d}t\Big). \tag{25}$$

Since the impact of the stuck time during information playout on the QoE is different, we consider $Q(t)$ as a satisfaction function concerning time t to evaluate the QoE changes in different intervals. In this proposal, we take $Q(t) = 1$ as an example to analyze the cache performance and assume that $\mathcal{R}(\cdot)$ is the QoE function, which is inversely proportional to the waiting time. Thus, the QoE optimization task can be regarded as a waiting time minimization problem, i.e., $\sum_{i=1}^{K+1} \int_{\mathbb{D}^m_{n,i-1}}^{\max\{\mathbb{D}^m_{n,i-1}, \mathbb{T}^m_{n,i}\}} \text{d}t$, which is equal to $\sum_{i=0}^{K+1} \mathcal{W}^m_{n,i}$.

Proposition 1: The waiting time of a complete content retrieval process is the sum of access delay minus play duration time in each stage, i.e.:

$$\mathcal{W}_n^m = \sum_{j=0}^{L+1} \mathcal{T}_{n,j}^m - \sum_{j=0}^{L} \mathcal{D}_{n,j}^m, \tag{26}$$

where L is the largest stage serial number that satisfies $\mathbb{T}_{n,L+1}^m > \mathbb{D}_{n,j}^m$.

Proof: See Appendix A.

We can rewrite the optimization problem concerning the cache allocation vector, i.e., $\mathbf{a}_m = \{\alpha_1^m, \alpha_2^m, ..., \alpha_K^m\}$, and power allocation vector, i.e., $\mathbf{P}$=$\{P_1, P_2, ..., P_K\}$. Thus, the optimal waiting time $\mathcal{W}_{\text{opt}}^m$ in region R_y can be calculated as:

$$\begin{aligned}
\mathcal{W}_{\text{opt}}^m =& \min_{\mathbf{a}_m, \mathbf{P}} q_{R_y(m)} \mathcal{W}_n^m \\
&\mathcal{C}1: A \cdot \alpha_k^m \leq \mathcal{C}_k, \\
&\mathcal{C}2: A \cdot \sum_{k=1}^{K} \alpha_k^m \leq s_m, \\
&\mathcal{C}3: \sum_{j=1}^{K} P_j \leq P_{\max}, \\
&\mathcal{C}4: \alpha_k^m \in \{0, 1, 2, ..., s_m/A\}, \forall m \in \mathbf{R}_y, \\
&\mathcal{C}5: P_k > 0, k = 1, 2, ..., K,
\end{aligned} \tag{27}$$

where $\mathcal{C}1$ and $\mathcal{C}2$ indicate that the cache allocation is subject to storage capacity and content size limitation, $\mathcal{C}3$ states that the transmission power assigned to file F_m in $\mathbf{R}_y$ is restricted to a maximum of $P_{\max}$, $\mathcal{C}_k$ is the cache capacity of B_k, $\mathcal{C}4$ is the integer constraint on cache allocation, and $\mathcal{C}5$ is the nonnegative constraint on power allocation.

4.2 Problem Simplification

The problem defined in (27) is a mixed-integer nonlinear program (MINLP) [27]. Thus, to simplify the calculation, we decouple (27) into two sub-problems, namely the power and cache allocation sub-problems. The optimality of these sub-problems is equivalent to the previous optimization problem, and to solve which we take $L = K$ as a special case.

Proposition 2: The waiting time is equivalent to the following expression with respect to $\mathbf{a}_m$ and $\mathbf{P}$:

$$\begin{aligned}
\mathcal{W}_n^m =& \sum_{i=1}^{K} \frac{r_t s_r^2}{\log_2(1 + \frac{P_i}{I+N_0})} + \frac{2KU_{n,1}^m}{c} + \sum_{i=1}^{K} \frac{2(i-1)D}{c} \\
&+ \sum_{i=1}^{K} \frac{r_t\{1 - [1 + 2(K-i)] \cdot \frac{\bar{v}_n}{c}\} \cdot \mathcal{S}(i)}{\log_2(1 + \frac{P_i}{I+N_0})} + \mathcal{W}_{n,0}^m \\
&+ D_{bh}[s_m - \sum_{i=1}^{K}(A \cdot \alpha_i^m)] - \sum_{i=1}^{K}(A \cdot \alpha_i^m \cdot \beta),
\end{aligned} \tag{28}$$

where

$$\mathcal{S}(i) = \begin{cases} (G_A + G_B)[A(\alpha_1^m - l_m)]^2 + G_C(A \cdot \alpha_1^m)^2, & i = 1 \\ (A \cdot \alpha_i^m)^2, & 2 \leq i \leq K. \end{cases} \tag{29}$$

Proof: See Appendix B.

4.2.1 Power allocation

For a given content placement decision $\{\alpha_1^m, \alpha_2^m, ..., \alpha_k^m\}$, the power allocation sub-problem can be formulated as:

$$\begin{aligned}
\min_{\mathbf{P}} \mathcal{W}_n^m =& \sum_{i=1}^{K} \frac{r_t s_r^2 + r_t\{1 - [1 + 2(K-i)] \cdot \frac{\bar{v}_n}{c}\} \cdot \mathcal{S}(i)}{\log_2(1 + \frac{P_i}{I+N_0})} \\
&- (D_{bh} + \beta) \sum_{i=1}^{K}(A \cdot \alpha_i^m) + \mathbb{C},
\end{aligned} \tag{30}$$

$$s.t. \quad \sum_{i=1}^{K} P_i \leq P_{\max}, \tag{30a}$$

$$P_i > 0, i = 1, 2, ..., K, \tag{30b}$$

where $\mathbb{C} = \mathcal{W}_{n,0}^{B/m} + D_{bh}s_m + \frac{2KU_{n,1}^m}{c} + \sum_{i=1}^{K} \frac{2(i-1)D}{c}$ contains items that are independent of cache and power allocation vectors, i.e., $\mathbf{a}_m$ and $\mathbf{P}$.

Lemma 1. The optimal problem can be organized as:

$$\begin{aligned}
\min_{\mathbf{R}} \mathcal{W}_n^m =& \sum_{i=1}^{K} \frac{r_t s_r^2 + r_t\{1 - [1 + 2(K-i)] \cdot \frac{\bar{v}_n}{c}\} \cdot \mathcal{S}(i)}{\mathcal{R}_i} \\
&- (D_{bh} + \beta) \sum_{i=1}^{K}(A \cdot \alpha_i^m) + \mathbb{C},
\end{aligned} \tag{31}$$

$$s.t. \quad \sum_{i=1}^{K} \mathcal{R}_i \leq \mathcal{R}_{\max}, \tag{31a}$$

$$\mathcal{R}_i > 0, k = 1, 2, ..., K, \tag{31b}$$

where $\mathbf{R} = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_K\}$, $\mathcal{R}_i = \log_2(1 + \frac{P_i}{I+N_0})$, and $\mathcal{R}_{\max} = K \log_2(\frac{|c\mathbf{h}_i\mathbf{h}_i^\dagger|^2 P_{\max} + KI + KN_0}{KI + KN_0})$.

Proof. See Appendix C.

Proposition 3: For a fixed cache decision vector, the optimal power assignment decision is $P_i = (I + N_0)(2^{\mathcal{R}_i} - 1)$, where $\mathcal{R}_i$ is:

$$\mathcal{R}_i = \frac{\mathcal{R}_{\max}\sqrt{r_t s_r^2 + r_t\{1 - [1 + 2(K-i)] \cdot \frac{\bar{v}_n}{c}\} \cdot \mathcal{S}(i)}}{\sum_{j=1}^{K} \sqrt{r_t s_r^2 + r_t\{1 - [1 + 2(K-j)] \cdot \frac{\bar{v}_n}{c}\} \cdot \mathcal{S}(j)}} \tag{32}$$

Proof. See Appendix D.

4.2.2 Content placement

By substituting **Proposition 3** into the waiting time minimization problem defined in (28), the following optimal cache allocation problem can be formulated:

$$\min_{\{\mathbf{a}_m\}} \mathcal{W}_n^m = \mathbf{S}^2/\mathcal{R}_{\max} - (D_{bh} + \beta) \sum_{i=1}^{K}(A \cdot \alpha_i^m) + \mathbb{C} \tag{33}$$

$$\text{s.t.} \quad \alpha_k^m \leq \mathcal{C}_k, \tag{33a}$$

$$\sum_{k=1}^{K} \alpha_k^m \leq s_m, \tag{33b}$$

$$\alpha_k^m \in \{0, 1, 2, ..., s_m/A\}, \tag{33c}$$

where $\mathbf{S} = \sum_{i=1}^{K} \sqrt{r_t s_r^2 + r_t\{1 - [1 + 2(K-i)] \cdot \frac{\bar{v}_n}{c}\} \cdot \mathcal{S}(i)}$.

Algorithm 1 The BRANCH & BOUND ALGORITHM

```
Input: Objective function $f(x)$, constraints $h_i(x)$, and integer requirments $IRs$
Output: Optimal solution $\mathbf{X}^*$ and optima Obj
System_Initilization:
sys.upp=sys.maxsize,sys.Q=[],Q$_1$=[],Q$_2$=[],opt_val=None,
opt_sol=None, cur_val=None, cur_sol=None.
function B&B($f(x)$,$h_i(x)$,$IRs$)
    if LP$_0$=LP($f(x), gi(x)$) is feasible then
        [cur_soln, cur_val]=LP$_0$
        Push [cur_sol, cur_val, gi(x)] to sys.Q.
    else  Return -1
    end if
    while sys.Q is not empty do
        [cur_sol, cur_val, gi(x)]= sys.Q.get()
        if cur_sol satisfies all IRs then
            if opt_val = None or opt_val > cur_val then
                [opt_val,opt_sol]=[cur_val,cur_sol]
                Return opt_val, opt_sol
            end if
        else
            Select a cur_sol.x$_j$=$b_j$ that do not satisfy IRs
            Let C$_1$ : $x_j \leq [b_j]$ C$_2$ : $x_j \geq [bj] + 1$
            if LP($f(x), (gi(x)$,C$_1$)) is feasible then
                [cur_sol,cur_val]=LP($f(x), (gi(x)$,C$_1$))
                Push [cur_sol,cur_val,($g_i(x)$,C$_1$)] to Q$_1$.
            else Push [sys.upp, None, ($g_i(x)$,C$_1$)] to Q$_1$.
            end if
            if LP($f(x), (gi(x)$,C$_2$)) is feasible then
                [cur_sol,cur_val]=LP($f(x), (gi(x)$,C$_2$))
                Push [cur_sol,cur_val,($g_i(x)$,C$_2$)] to Q$_2$.
            else Push [sys.upp, None, ($g_i(x)$,C$_2$)] to Q$_2$.
            end if
            if Q$_1$[0]<Q$_2$[0] then
                Push Q$_1$ to sys.Q
            else
                Push Q$_2$ to sys.Q
            end if
        end if
    end while
end function
```

4.3 The Algorithm

The problem defined in (33) is an integer nonlinear program (INLP). Although an INLP can be directly solved by Branch & Bound (B&B) algorithm [28], the time complexity of the B&B will sharply increase with the increasing SBS quantity. Furthermore, it requires higher time and space complexity due to the variable dimension and the computational complexity of the derivation. Therefore, in this paper, we adopt a modified Extended Cutting Plane method, named αECP [27] to solve Problem (33).

We first substitute the objectives and constraints into αECP as:

$$\min_{\mathbf{a}_m,\mu\in L} \mu \tag{34}$$

$$\text{s.t.} \quad \mathcal{W}_n^m(\mathbf{a}_m) - \mathcal{W}_n^m(\mathbf{a}_m^{o-l}) \leq 0, \tag{34a}$$

$$D_j[\mathcal{W}_n^m(\mathbf{a}_m)] - \mu \leq 0, j = o-l, ..., o, \tag{34b}$$

TABLE 2: The calculation complexity of the B&B algorithm.

SBS number	Iterations	Elapsed time (s)
1	10	0.065
2	26	0.233
3	57	0.598
4	120	1.317
5	247	2.295

where μ is an additional variable, L is the linear constraint of (33), and $D_j[\mathcal{W}_n^m(\mathbf{a}_m)] = \mathcal{W}_n^m(\mathbf{a}_m^j) - \nabla\mathcal{W}_n^m(\mathbf{a}_m^j)^T(\mathbf{a}_m - \mathbf{a}_m^j)$.

First, let $o = l = 0$ and select an initial point $\mathbf{a}_m^0$. Then, solve the integer linear optimization problem (ILP) P_0, i.e., the objective function given in (34), s.t. (34a) (34b). Using the B&B provoded in Algorithm 1, we can get the optimal solution of P_0, which is denoted as $\mathbf{a}_m^o$.

Second, if $\mathcal{W}_n^m(\mathbf{a}_m^{o+1}) < \mathcal{W}_n^m(\mathbf{a}_m^o)$, replace the objective with $D_{o+1}[\mathcal{W}_n^m(\mathbf{a}_m)]$, and let $o = o+1$. If $\mathcal{W}_n^m(\mathbf{a}_m^{o+1}) = \mathcal{W}_n^m(\mathbf{a}_m^o)$, add $D_{o+1}[\mathcal{W}_n^m(\mathbf{a}_m)]$ to the objective and let $o = o+1, l = l+1$.

Third, if $|\mathcal{W}_n^m(\mathbf{a}_m^{o-l}) - \mu| < \epsilon_\mu$, terminate the algorithm and take $\mathbf{a}_m^* = \mathbf{a}_m^{o+1}$. Else, solve the current problem and perform the second part.

A more detailed procedure can be found in [27]. In our experiments, we consider ϵ_μ as a positive tolerance. The results are not optimal sometimes, but they are usually close to the global optimal solution. Also, ϵ_μ can act as a balancer between calculation accuracy and complexity. The variations between iteration/calculation time and the SBS quantity are shown in Table 2, which are obtained using a laptop with Intel(R) Core(TM) i5-8265U CPU@1.6GHz. Note that Table 2 only shows the iterations for a single calculation, and different initial point selections might affect the actual calculation.

5 Simulation Results

5.1 Waiting time

The purpose of enabling content cache in inhomogenous ICVs is to reduce the waiting time during service access. In this part, we evaluate the performance of the proposed caching strategy in terms of users' waiting time with respect to various parameters. The content size ranges from 40 to 55 units, where the size of each segment is fixed to 1, the default ICV/SBS cache size is set to [2-6]/[6-8] units, the length of the FlS ranges from 1 to 6 units, and the onboard cache probability is set to [0.1,0.25).

Fig. 6 illustrates the influence of SBS quantity and backhaul delay on the user's waiting time. For a fixed D_{bh}, the waiting time of content requests decreases with the increasing SBS quantity. The more SBSs deployed, the more contents available, which avoids remote content fetching procedures through the V2N. Consequently, the burden of the MBS would be alleviated, causing shorter $\mathcal{W}_n^m$. As depicted, smaller D_{bh} can offer a shorter waiting time to the user. Once the number reaches a certain level (in this setting, K=7), SBSs are capable of caching the whole content. In this case, with the continuous increase in the SBS quantity, the decrease rate of the waiting time will slow down. Because when the cache space is large enough, the requester will not

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3137219, IEEE Transactions on Mobile Computing

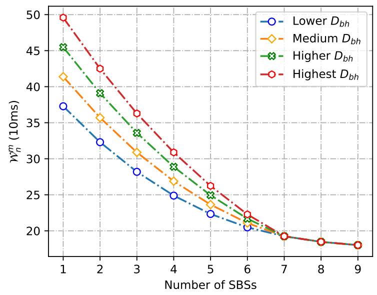


Fig. 6: Waiting time with different number of SBSs and D_{bh}.

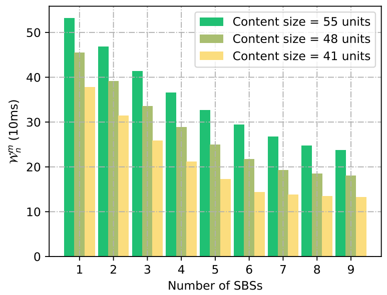


Fig. 7: Waiting time with different number of SBSs and content size.

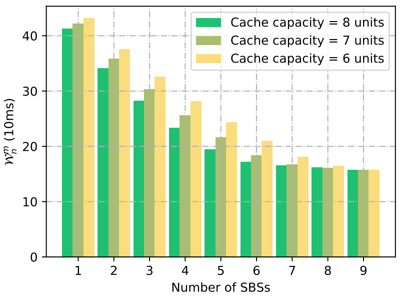


Fig. 8: Waiting time with different number of SBSs and cache capacity.

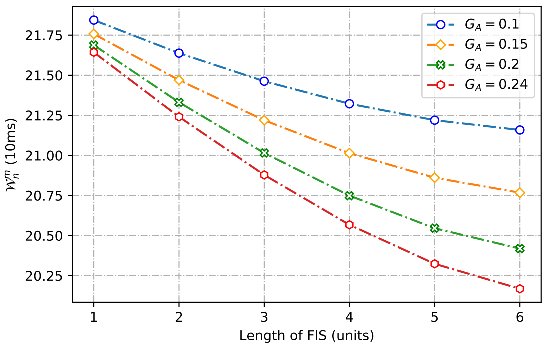


Fig. 9: Waiting time for different FlS lengths and OBU cache probability.

get content units from MBS anymore, and thus the difference in D_{bh} will not affect the waiting time. As the number of SBS increases, the available cache space becomes larger and larger. In this case, the content distribution method can be adjusted to reduce the waiting time; however, this downward trend will eventually saturate.

Fig. 7 and Fig. 8 reveal that the waiting time witnessed a decreasing trend for the increasing SBS quantity, concerning different content sizes and cache capacity. The waiting time first decreases and then tends to be stable. Moreover, as the file size increases, the waiting time gets longer. A larger content size will push the stable inflection point to a larger SBS quantity. As shown in Fig. 7, for the file with 41 units, the waiting time reaches the inflection point when the number of SBS reaches 6, and with the increase of the SBS quantity, the decline rate of the waiting time starts to slow down. However, the inflection point is $K = 7$ for the file with 48 units. As can be seen, the waiting time will increase as the content size increases. Fig. 8 shows the waiting time under different numbers of SBSs and SBSs' cache capacity. With the increasing SBS cache space, the waiting time of content requests decreases gradually. In summary, when the number of SBSs reaches a certain level, the waiting time will remain unchanged. Moreover, the stationary point will come earlier as the storage capability of the SBS increases. These phenomena are attributed to the decreasing content size retrieved via V2N communication.

Fig. 9 evaluates the waiting time's variation during successful file access under different OBU cache probability and FlS lengths. As shown, a considerable decrease in the waiting time occurs as the length of FlS increases. The larger the cache probability is, the shorter the waiting time. In pace with a continuous increment in the OBU cache probability, it is clear that the decrease rate of the waiting time will slow down. This phenomenon is attributed to the fact that the node will obtain the FlS from a local or other OBU cache with greater probability when the onboard cache probability increases, further cut down the total waiting time.

5.2 QoE Evaluation

In general, users' QoE is related to the actual waiting time and the size of the accessed services. For instance, there is a QoE difference between receiving 2 Gigabyte (2GB) content within 3 seconds and receiving 4GB file within 3 seconds. The QoE in this paper is therefore defined as:

$$QoE_m = \mathcal{R}(\mathcal{W}_n^m) = \frac{1}{e^{\mathcal{W}_n^m/s_m - \mathrm{Exp}_m}}, \qquad (35)$$

where Exp_m is the expected waiting time of content F_m.

Fig. 10 shows the variation of the QoE for different SBS quantities and backhaul latency. For the increasing SBS quantity, the overall tendency of the QoE is inversely proportional to the waiting time. However, by comparing Fig. 6 and Fig. 10, it can be found that in the wake of SBS quantity increases, waiting time showed a downward trend, while the QoE showed the opposite. Similar to Fig. 6, when the number of SBSs is large enough, the content request can be satisfied without connecting to the MBS. Therefore, when the amount of SBSs is more than 6, the four curves will overlap.

In Fig. 11 and Fig. 12, we evaluate the impact of content size and cache capacity on the QoE for a fixed D_{bh}. Similar to the previous analysis, the QoE shows the opposite trend of the waiting time. As expected, a smaller content size usually leads to a better QoE since the length of Exp_m increases with the increasing content size, and the delay of unit content acquisition is relatively smaller when the SBS has enough cache capacity. The larger cache capacity can also improve the user's QoE, thereby enabling the turning point in the rate of QoE change to be reached earlier in Fig. 11 and Fig. 12.

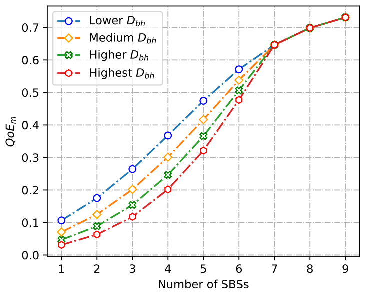


Fig. 10: QoE for different numbers of SBSs and D_{bh}.

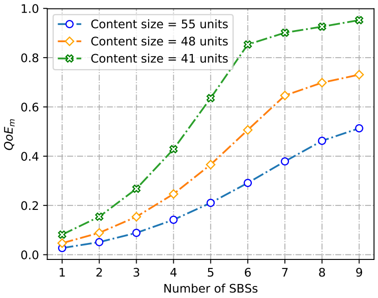


Fig. 11: QoE for different numbers of SBSs and content sizes.

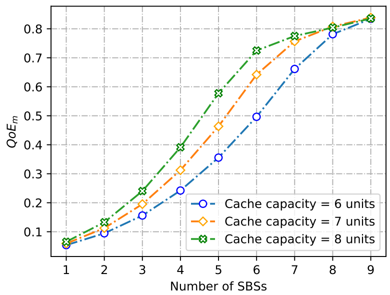


Fig. 12: QoE for different numbers of SBSs and cache capacities.

5.3 Comparison of Different Cache Schemes

Here, we evaluate the performance of our information-centric in-network caching scheme (ICIC) in terms of waiting time, QoE, and cache space utilization. To show the superiority of the ICIC, we compare it with two mainstream caching schemes, i.e., distribute content segments in on-board cache entities and allocate content units in the SBSs.

On-board cache scheme: In onboard cache schemes, such as [15] and [19], content units are mainly placed in the ICV, and user's requests are satisfied through V2V communications. We first compare the ICIC with an onboard cache scheme (labeled as the TR-ICV) in this subsection. Considering that the cache space of vehicles is usually smaller than that of the SBS, we assume that vehicles can effectively collaborate in the TR-ICV, and different parts of a complete content can be distributed to various ICV nodes.

Moreover, the performance of the onboard cache strategy is subject to vehicle mobility. Therefore, we adopt a reference point group mobility (RPGM) model [29] defining ICV movement. Specially, we focus on the randomness of vehicle mobility on lateral movement, considering that the direction of the random motion vector in the RPGM is lateral. In addition, we assume that the velocity of the checkpoint in the RPGM is fixed, and the ICVs' movement feature can be seized.

Fig. 13 shows the content access performance for different onboard cache probabilities and node densities. In this simulation, the cache capacity of each ICV (referred to as normalized caching capability) is set to about 10% of all content. Besides, the cache probability is set to the range of 0.2-1.0, and the vehicles' density is set to [3,8]/100m. Afterward, we compare the ICIC and the TR-ICV in terms of waiting time and data transmission. Fig. 13 shows that the ICIC outperforms the TR-ICV when the onboard cache probability is relatively low. For the TR-ICV, the total waiting time shows an approximately linear decrease trend for the increasing onboard cache probability, since the TR-ICV has several content units that are accessed through the MBS with a limited onboard cache capacity. Moreover, the increase in node density enables the requester to get content from its neighbor with greater probability, thus reducing the waiting time. For the ICIC, as expected, the fluctuation of $\mathcal{W}_n^m$ is relatively smaller than that of the TR-ICV. The reason for that is when the content cannot be accessed from the vehicle, the requester will fetch it from the SBS instead of the MBS. The polyline below is the time required for the content to be transmitted through the ICV and the SBS. In this experiment setting (consistent with the actual situation), the storing capability of the SBS is far greater than the ICVs' cache capacity. As such, the transmission time of the SBS is greater than that of the ICV. Since the required service time of the MBS in the TR-ICV is much longer than that of the SBS in the ICIC, the total waiting time of the ICIC is smaller than that of the TR-ICV.

Fig. 14 zooms in on the ICV and SBS transmission time polyline in Fig. 13. The transmission time of the onboard cache scheme shows an increasing trend, which becomes

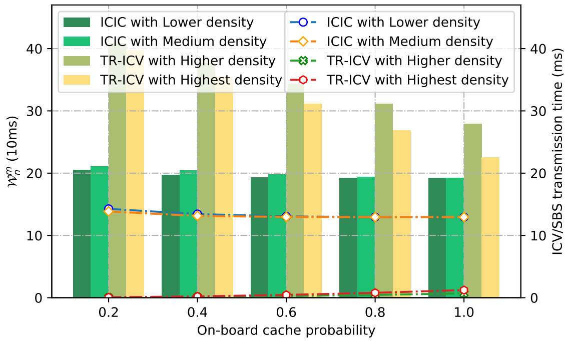


Fig. 13: Waiting and short-range communication times for different onboard cache probabilities.

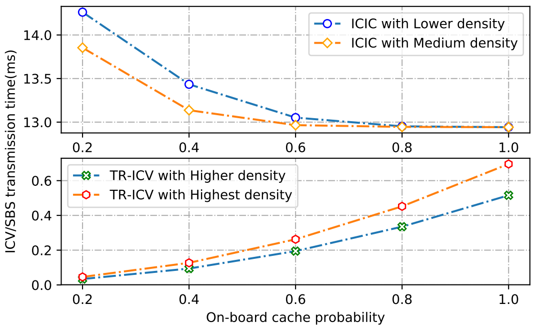


Fig. 14: Short-range communication time for different onboard cache probabilities.

more obvious with the increasing cache probability. Furthermore, the required ICV transmission time in the highest density scenario is generally greater than in the higher situation. Because when the vehicle density is high, the proportion of content obtained through V2V will be greater, leading to a longer V2V transmission time. Conversely, a higher onboard cache probability will cut down the V2V plus V2B communication time in the ICIC. That is because the total amount of content obtained by short-distance communication remains unchanged, while the transmission time of the FlS becomes shorter due to the higher onboard cache probability. It should be noted that the TR-ICV usually performs satisfactorily for a higher vehicle density, whereas the performance of the ICIC is not sensitive to the changes in density. Therefore, the higher and highest ICV densities are selected in the TR-ICV, while the ICIC adopts the lower and medium densities, to magnify the difference between these two schemes.

The SBS/RSU-assisted caching scheme: In the SBS/RSU-assisted caching schemes, such as [20] and [21], content units are placed in SBSs or RSUs that are distributed on the roadside, and the various ICV requests are satisfied through V2B communications. Therefore, we will compare the performance difference in the ICIC, i.e., sub-blocks caching vs. mono-block caching, which is labeled as TR-SBS.

Fig. 15 presents the cache utilization of the two caching schemes for different file sizes. In this experiment, we set the amount of SBSs to 5 and the cache capability of each SBS to 10 units. Besides, the average size of the content units is set to the range of [1,7] units. As seen, the TR-SBS shows an approximately downward trend. In general, the TR-SBS places the complete content block in one SBS. When the remaining cache space of the current SBS is not adequate, the whole content will be placed in the next SBS, which leads to a waste of cache resources. Thus, the utilization of cache space will decrease with the increase of average content size. In this simulation, we also assume that the content size follows a normal distribution with different mean sizes. Due to the uncertainty of content size distribution, there may be random fluctuations in utilization at several points, but it does not affect the overall trend. As we can also observe, the ICIC achieves a major increase and reaches a relatively steady level, gradually outperforming the TR-SBS. In this experiment, we did not set the size of the content to an integer, i.e., s_m may not be divided by A with no remainder. In the ICIC, those fractional parts will be treated as 1. Hence, when the content is small, the utilization rate of the ICIC will be lower than that of the TR-SBS. However, as the average content size increases, these fractional parts account for smaller and smaller proportions, which helps the ICIC to achieve better performance.

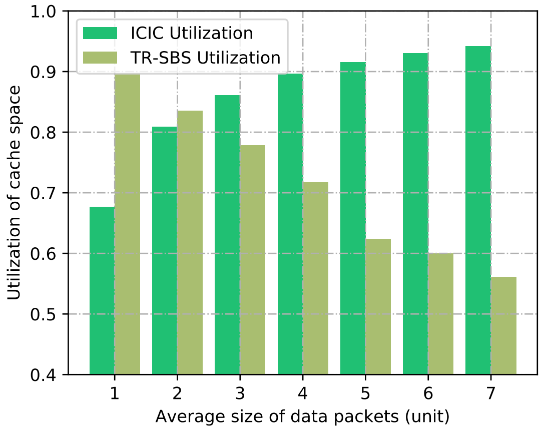


Fig. 15: Storage utilization for different average sizes of data packets.

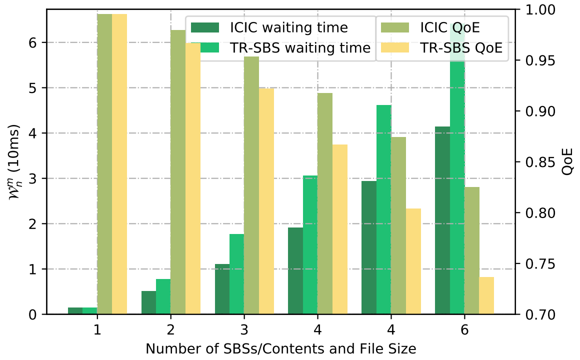


Fig. 16: Waiting time and QoE for different number SBSs/contents and file sizes.

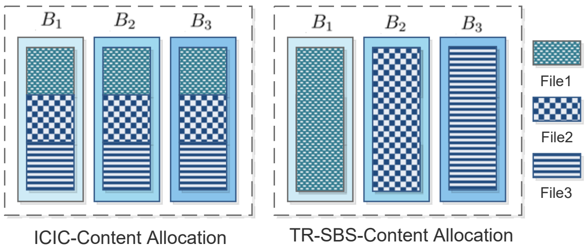


Fig. 17: Different cache allocations of the ICIC and the TR-SBS.

Fig. 16 analyzes the impact of SBS/file quantities and file sizes on the waiting time and the QoE. In this experiment, we set the number of SBSs, content number, and file size as [1,6] while keeping other parameters constant. Fig. 17 shows the cache allocation difference between the ICIC and the RT-SBS (taking K=M=S_m=3 as an example). The ICIC breaks up the content and distributes it to different SBSs, while the TR-SBS places the entire content in one SBS. This special parameter setting method can ensure that the cache utilization in both methods is kept at 1. We can observe that the required waiting time increases with the scale of the network, while the QoE shows a downward trend. When the dynamic parameter is set to 1, the ICIC and the TR-SBS perform similarly. However, when the dynamic parameter is greater than 1, the ICIC outperforms the TR-SBS, and the gap between the two increases with the growth of parameters.

6 Conclusions

This paper introduces a content placement strategy for the ICVs to improve the users' QoE. By integrating the mobility characteristics, heterogeneous/homogeneous transmission features, and content popularity patterns of vehicles, the proposed method can help cache entities make wiser cache decisions and thus achieve better content retrieval performance.

In our future studies, we will design an incentive strategy to deal with more complicated service requests in the V2X communication framework. This strategy will encourage different network participants, including vehicles, pedestrians, or infrastructures, to energetically serve others via homogeneous or heterogeneous communications without accessing the core network.

References

[1] X. Wang, Z. Ning, X. Hu, L. Wang, L. Guo, B. Hu, and X. Wu, “Future communications and energy management in the internet of vehicles: Toward intelligent energy-harvesting,” *IEEE Wireless Communications*, vol. 26, no. 6, pp. 87–93, 2019.

[2] H. Ullah, N. G. Nair, A. Moore, C. Nugent, P. Muschamp, and M. Cuevas, “5G communication: An overview of vehicle-to-everything, drones, and healthcare use-cases,” *IEEE Access*, vol. 7, pp. 37251–37268, 2019.

[3] J. Tang, B. Shim, and T. Q. Quek, “Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 881–895, 2019.

[4] A. Asheralieva and D. Niyato, “Game theory and lyapunov optimization for cloud-based content delivery networks with device-to-device and uav-enabled caching,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10094–10110, 2019.

[5] S. Park, S. Oh, Y. Nam, J. Bang, and E. Lee, “Mobility-aware distributed proactive caching in content-centric vehicular networks,” *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 175–180, Paris, France, 2019.

[6] T. Deng, P. Fan, and D. Yuan, “Optimizing retention-aware caching in vehicular networks,” *IEEE Transactions on Communications*, vol. 67, no. 9, pp. 6139–6152, 2019.

[7] A. Kammoun, M.-S. Alouini, *et al.*, “Elevation beamforming with full dimension MIMO architectures in 5G systems: A tutorial,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3238–3273, 2019.

[8] Meng, Naeem, Ali, Y. Zikria, and S. W. Kim, “DCS: Distributed caching strategy at the edge of vehicular sensor networks in information-centric networking,” *Sensors*, vol. 19, pp. 4407–4425, 2019.

[9] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, “Cooperative edge caching in user-centric clustered mobile networks,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1791–1805, 2017.

[10] Y. Cao, C. Long, T. Jiang, and S. Mao, “Share communication and computation resources on mobile devices: A social awareness perspective,” *IEEE Wireless Communications*, vol. 23, no. 4, pp. 52–59, 2016.

[11] S. Mastorakis, A. Mtibaa, J. Lee, and S. Misra, “ICedge: When edge computing meets information-centric networking,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4203–4217, 2020.

[12] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, “Caching in vehicular named data networking: Architecture, schemes and future directions,” *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2378–2407, 2020.

[13] J. Liao, K.-K. Wong, M. R. Khandaker, and Z. Zheng, “Optimizing cache placement for heterogeneous small cell networks,” *IEEE Communications Letters*, vol. 21, no. 1, pp. 120–123, 2016.

[14] L. Yao, Y. Wang, X. Wang, and G. WU, “Cooperative caching in vehicular content centric network based on social attributes and mobility,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 391–402, 2021.

[15] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, and L. Zhu, “A mobility-aware vehicular caching scheme in content centric networks: Model and optimization,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3100–3112, 2019.

[16] G. Deng, L. Wang, F. Li, and R. Li, “Distributed probabilistic caching strategy in VANETs through named data networking,” in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 314–319, 2016.

[17] E. Chan, Y. Wang, W. Li, and S. Lu, “Movement prediction based cooperative caching for location dependent information service in mobile ad hoc networks,” *Journal of Supercomputing*, vol. 59, no. 1, pp. 297–322.

[18] X. Zhuo, Q. Li, G. Cao, Y. Dai, and T. L. Porta, “Social-based cooperative caching in dtns: A contact duration aware approach,” in *IEEE 8th International Conference on Mobile Adhoc and Sensor Systems, Valencia, Spain*, 2011.

[19] C. Wang, C. Chen, Q. Pei, N. Lv, and H. Song, “Popularity incentive caching for vehicular named data networking,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020, doi:10.1109/TITS.2020.3038924.

[20] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu, and Y. Jia, “An edge caching scheme to distribute content in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5346–5356, 2018.

[21] Z. Zhang, C.-H. Lung, M. St-Hilaire, and I. Lambadaris, “Smart proactive caching: Empower the video delivery for autonomous vehicles in icn-based networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7955–7965, 2020.

[22] C. Song, W. Xu, T. Wu, S. Yu, P. Zeng, and N. Zhang, “QoE-driven edge caching in vehicle networks based on deep reinforcement learning,” *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2021, doi: 10.1109/TVT.2021.3077072.

[23] J. A. Khan and Y. Ghamri-Doudane, “ROVERS: Incentive-based recruitment of connected vehicles for urban big data collection,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5281–5294, 2019.

[24] Z. Su, Q. Xu, F. Hou, Q. Yang, and Q. Qi, “Edge caching for layered video contents in mobile social networks,” *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2210–2221, 2017.

[25] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: Evidence and implications,” in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, vol. 1, pp. 126–134, IEEE, 1999.

[26] A. Zaidi, F. Athley, J. Medbo, U. Gustavsson, G. Durisi, and X. Chen, “5G physical layer: Principles, models and technology components,” *Elsevier*, 2018.

[27] R. Pörn and T. Westerlund, “A cutting plane method for minimizing pseudo-convex functions in the mixed-integer case,” *Computers & Chemical Engineering*, vol. 24, pp. 2655–2665, 2000.

[28] J. Clausen, “Branch and bound algorithm-principles and examples (technical report),” *University of Copenhagen*, 1999.

[29] X. Hong, M. Gerla, G. Pei, and C.-c. Chiang, “A group mobility model for ad hoc wireless networks,” *Proceedings of the 2nd ACM international workshop on modeling, analysis and simulation of wireless and mobile systems*, pp. 53–60, 1999, doi:10.1145/313237.313248.

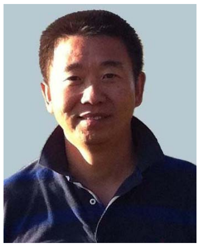
Cong Wang received the B.Eng. in Electronic and Information Engineering from Chongqing Jiaotong University, Chongqing, China, in 2016. Since 2016, she has been working on her PhD degree at Xidian University, Xi‘an, China. Her research interests include wireless communication, computer engineering, traffic information, and control engineering.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3137219, IEEE Transactions on Mobile Computing

Chen Chen (M'09-SM'18) received the B.Eng., M.Sc., and Ph.D. degrees in Telecommunication Engineering from Xidian University, Xi'an, China, in 2000, 2006, and 2008, respectively. He is currently a Professor with the Department of Telecommunication and a member of The State Key Laboratory of Integrated Service Networks at Xidian University. He is also the Director of the Xi'an Key Laboratory of Mobile Edge Computing and Security, and the Director of the Intelligent Transportation Research Laboratory at Xidian University. He was a visiting professor at the Department of EECS at the University of Tennessee and the Department of CS at the University of California. He serves as General Chair, PC Chair, Workshop Chair, and TPC Member of several conferences. He has authored/co-authored 2 books and over 130 scientific papers in international journals and conference proceedings. He has contributed to the development of 5 copyrighted software systems and invented over 100 patents. He is also a Senior Member of the China Computer Federation (CCF) and China Institute of Communications (CIC).

Qingqi Pei (SM'15) received the B.S., M.S., and Ph.D. degrees in Computer Science and Cryptography from Xidian University, in 1998, 2005, and 2008, respectively. He is currently a Professor and a member of the State Key Laboratory of Integrated Services Networks. He is also a Professional Member of the ACM and a Senior Member of the Chinese Institute of Electronics and China Computer Federation. His research interests focus on cognitive networks, data security, and physical layer security.

Zhiyuan Jiang (S'12-M'15) received the B.S. and Ph.D. degrees from the Electronic Engineering Department, Tsinghua University, China, in 2010 and 2015, respectively. He is currently a Professor at the School of Communication and Information Engineering, Shanghai University, Shanghai, China. He visited the WiDeS Group, University of Southern California, Los Angeles, CA, USA, from 2013 to 2014. He worked as an experienced researcher at Ericsson from 2015 to 2016. He visited ARNG at the University of Southern California, Los Angeles, CA, USA, from 2017 to 2018. He worked as a Wireless Signal Processing Scientist at Intel Labs, Hillsboro, OR, USA, in 2018. His current research interests include URLLC in wireless networked control systems and signal processing in MIMO systems. He serves as a TPC member for IEEE INFOCOM, ICC, GLOBECOM, and WCNC. He received the ITC Rising Scholar Award in 2020, the Best Paper Award at the IEEE ICC 2020, the best In-Session Presentation Award of IEEE INFOCOM 2019, and the Exemplary Reviewer Award of IEEE WCL in 2019. He serves as an Associate Editor for the IEEE/KICS Journal of Communications and Networks, and a Guest Editor for the IEEE IoT Journal.



Shugong Xu (M'98-SM'06-F'16) graduated from Wuhan University, China, in 1990, and received a master's degree in pattern recognition and intelligent control and a Ph.D. degree in EE from the Huazhong University of Science and Technology (HUST), China, in 1993 and 1996, respectively. He is currently a Professor with Shanghai University and the Head of the Shanghai Institute for Advanced Communication and Data Science (SICS). He was the Center Director and Intel Principal Investigator of the Intel Collaborative Research Institute for Mobile Networking and Computing (ICRI-MNC) before December 2016, when he joined Shanghai University. Before joining Intel in September 2013, he was the Research Director and Principal Scientist at the Communication Technologies Laboratory, Huawei Technologies. Among his responsibilities at Huawei, he founded and directed Huawei's Green Radio Research Program, Green Radio Excellence in Architecture and Technologies (GREAT). He was also the Chief Scientist and PI for the China National 863 project on EndtoEnd Energy Efficient Networks. He was one of the co-founders of the Green Touch Consortium together with Bell Labs. He has served as the Co-Chair of the Technical Committee for three terms in this international consortium. Before joining Huawei in 2008, he was with Sharp Laboratories of America as a Senior Research Scientist. Before that, he conducted research as a Research Fellow in City College of New York, Michigan State University, and Tsinghua University. He published over 100 peer-reviewed research papers in top international conferences and journals. One of his most referenced papers has over 1400 Google Scholar citations, in which the findings were among the major triggers for the research and standardization of the IEEE 802.11S. He has over 20 U.S. patents granted. Some of these technologies have been adopted in international standards, including the IEEE 802.11, 3GPP LTE, and DLNA. He was awarded 'National Innovation Leadership Talent' by China Government in 2013, was elevated to IEEE Fellow in 2015 for contributions to the improvement of wireless networks efficiency. He is also the Winner of the 2017 Award for Advances in Communication from IEEE Communications Society. His current research interests include wireless communication systems and machine learning.