Toward Efficient Content Delivery for Automated Driving Services: An Edge Computing Solution

Quan Yuan, Haibo Zhou, Jinglin Li, Zhihan Liu, Fangchun Yang, and Xuemin (Sherman) Shen

Abstract

Automated driving is coming with enormous potential for safer, more convenient, and more efficient transportation systems. Besides onboard sensing, autonomous vehicles can also access various cloud services such as high definition maps and dynamic path planning through cellular networks to precisely understand the real-time driving environments. However, these automated driving services, which have large content volume, are time-varying, location-dependent, and delay-constrained. Therefore, cellular networks will face the challenge of meeting this extreme performance demand. To cope with the challenge, by leveraging the emerging mobile edge computing technique, in this article, we first propose a two-level edge computing architecture for automated driving services in order to make full use of the intelligence at the wireless edge (i.e., base stations and autonomous vehicles) for coordinated content delivery. We then investigate the research challenges of wireless edge caching and vehicular content sharing. Finally, we propose potential solutions to these challenges and evaluate them using real and synthetic traces. Simulation results demonstrate that the proposed solutions can significantly reduce the backhaul and wireless bottlenecks of cellular networks while ensuring the quality of automated driving services.

Introduction

With the rapid advancement of automated driving (AD) technology, vehicles with varying levels of automation[1] are already on the road. It is widely expected that autonomous vehicles, especially fully autonomous vehicles, will offer safer and more efficient transportation [1]. A prerequisite for autonomous vehicles is the capability to understand their ambient environments in real time. Autonomous vehicles rely on a combination of sensors, such as cameras, radars, and lidars, to perceive their surroundings and make driving decisions. As these onboard sensors are limited to line of sight, communication devices are exploited as extended sensors to help the vehicles see beyond the reach of onboard sensors. Specifically, vehicles can exchange information with nearby vehicles and infrastructure through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications [2]. Moreover, vehicles can access automated driving services through vehicle-to-cloud (V2C) communications [3]. These online services provide essential information to enable the vehicles to plan precisely and manoeuver correctly. For instance, high definition (HD) maps,[2] and real-time traffic information and parking guidance are fundamental automated driving services. Considering the network coverage, vehicles generally use diverse cellular network techniques — Long Term Evolution (LTE) and fifth generation mobile networks — to access service contents anytime and everywhere. *The automated driving services, which have large content volume, are time-varying, location-dependent, and delay-constrained* [4]. Therefore, three main challenges are posed to cellular networks to ensure the safety, comfort, and efficiency of automated driving. First of all, with the rising popularity of autonomous vehicles, the wireless and backhaul capacity of cellular networks will become the first issue to meet the overwhelming content demands. In addition, it is challenging for the cellular networks to provide end-to-end delay guarantees for the cloud-based automated driving services. Third, it is crucial to consider the reliability and efficiency when the remote cloud collects the context (e.g., position, velocity, acceleration) and coordinates the behavior of autonomous vehicles.

Mobile edge computing (MEC) is proposed to empower the wireless edge [5]. The key concept of MEC is to place the computing and storage resources at the network edge, that is, base stations (BSs) and user equipment (UE, here refers to vehicles). Therefore, part of the data processing and storage can be pushed from the remote cloud to the edge, which is proximate to vehicles. Although the contents of automated driving services have large volume, the MEC paradigm provides opportunities to enhance the cellular networks to ensure the quality of automated driving. Since the automated driving services are location-dependent, co-located vehicles usually have shared service demands. The edge intelligence or computing capability can exploit the locality of service demands to assist the content delivery. On one hand, the edge intelligence can use the edge storage to elaborately cache the locally popular contents. On the other hand, the edge intelligence can monitor the status of nearby vehicles and coordinate the content delivery via vehicular networks. In this way, a large number of ser-

[1] SAE International defines six levels of driving automation: 0 (no automation), 1 (driver assistance), 2 (partial automation), 3 (conditional automation), 4 (high automation), and 5 (full automation).

[2] An HD map is a machine-readable map that models the surface of the road to an accuracy of 10–20 cm. It contains the three-dimensional representation of all crucial aspects of a roadway (e.g., slope and curvature, lane marking types, and roadside objects) and dynamic information that facilitates driving (e.g., lane level speed limit information, lane closures, potholes, accidents). Autonomous vehicles should localize themselves in the map and use the map to understand the real world.

Digital Object Identifier: 10.1109/MNET.2018.1700105

Quan Yuan, Jinglin Li (corresponding author), Zhihan Liu, and Fangchun Yang are with Beijing University of Posts and Telecommunications; Haibo Zhou is with the University of Waterloo and Nanjing University; Xuemin (Sherman) Shen is with the University of Waterloo.

0890-8044/18/$25.00 © 2018 IEEE

vice demands will be fulfilled at the mobile edge, which largely reduces the wireless and backhaul bottlenecks. In the meanwhile, the end-to-end delays are reduced when the service demands are fulfilled locally. Since the general computing and storage devices are becoming much cheaper, MEC will be a cost-effective solution to enable cellular networks to meet the requirements of automated driving services.

In this article, we propose a two-level edge computing architecture to coordinate the content delivery for automated driving services. The edge intelligence at BSs is in charge of the caching policies for both BSs and vehicles. Meanwhile, the intelligence in vehicles is responsible for cooperatively determining the content sharing schemes in vehicular networks. Our proposed architecture can adapt to the unique characteristics of automated driving services. Specifically, the edge caching policies consider the varying distribution of service demands and the dynamic topology of vehicular networks. The caching policies at BSs are designed through revealing the content demand patterns of automated driving services. The caching policies at vehicles are designed through analyzing each vehicle's influence on content dissemination in vehicular networks. Moreover, the content sharing schemes consider the strict delay constraints of services and utilize the broadcast nature of V2V communications. Coded broadcast is exploited to further improve the efficiency of content sharing among vehicles.

The remainder of this article is organized as follows. The recent literature on content caching and delivery is reviewed. Then an edge-assisted content delivery architecture for automated driving services is proposed. The details of research challenges are discussed, followed by potential solutions and their application. Finally, the work is concluded, and future research directions are discussed.

Literature Review

Edge caching is cost effective for automated driving related service content delivery due to the locality of service demands. Although not optimized for the automated driving services, some general edge caching solutions have been proposed recently to increase the spectral efficiency of cellular networks. Zeydan *et al.* [6] exploited the contextual information to predict users' spatiotemporal demands and proactively cache popular contents at BSs. Furthermore, considering the limited contact duration between users and small cell BSs (SBSs), Poularakis *et al.* [7] designed a mobility-aware caching policy for SBSs. Ji *et al.* [8] proposed a scheme that randomly caches contents on wireless devices and enables device-to-device (D2D) communications to deliver the contents. On the other hand, there are some studies that investigate the general data dissemination problems in heterogeneous vehicular networks. Different network resources are scheduled considering the application requirements and communication constraints. He *et al.* [9]and Liu *et al.* [10] applied software-defined networking (SDN) to schedule cooperative data transmission in heterogeneous vehicular networks. Taking into account the spatiotemporal constraints on content delivery, our previous work [11] relied on selected influential vehicles to offload cellular traffic to vehicular networks. In addition, Wang *et al.* [12] utilized coalition formation games to cooperatively deliver contents among vehicles, which can optimize the performance of average delay and power efficiency.

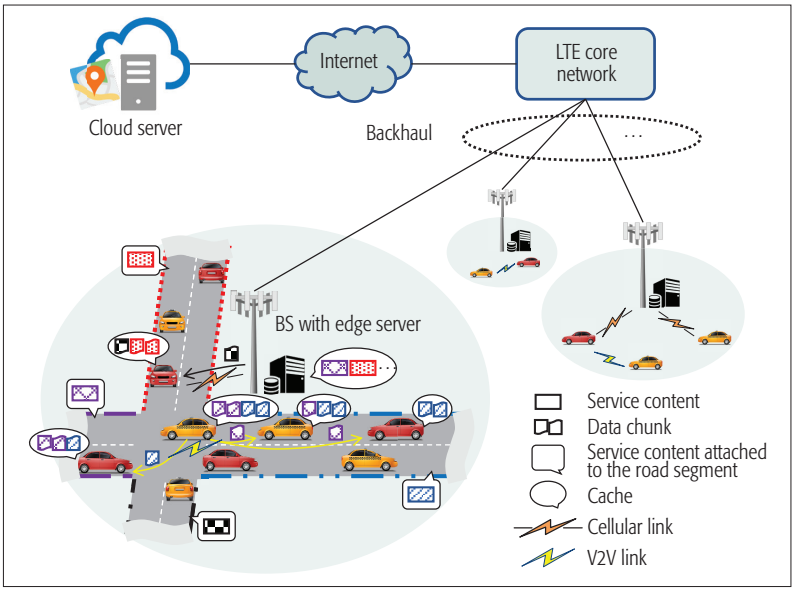


FIGURE 1. The architecture of edge-assisted content delivery for automated driving services.

The Architecture of Edge-Assisted Content Delivery for Automated Driving Services

In this section, we first propose a two-level edge computing architecture for automated driving related service content delivery, which is shown in Fig. 1. Then we introduce the basic procedures for edge intelligence to coordinate content delivery under the proposed architecture.

Two-Level Edge Computing Architecture

To provide online automated driving services (e.g., HD map, real-time traffic, parking guidance), service providers can deploy application servers on the cloud. Each service content describes the dynamic features of a specific road segment. Before traveling into a road segment, vehicles should request and completely fetch the corresponding service content. In other words, the content delivery is strictly space-constrained. The vehicles integrate the information acquired from onboard sensors and online services to understand their ambient environments. Since these services rely on an always-on network, autonomous vehicles are supposed to access the service contents through cellular networks.

Based on the concept of MEC, in our architecture, computing and storage resources are added to the two-level wireless edge (i.e., both BSs and autonomous vehicles). Specifically, an edge server is implemented at each BS to serve as an agent or intermediary between cloud servers and vehicles. It can efficiently be aware of the context of associated vehicles. The storage resources at BSs can be used to cache contents for all kinds of mobile services (e.g., automated driving, multimedia, augmented reality, news). Therefore, the cache space

To initiate content sharing, each vehicle can inform neighbors of its own service demands and corresponding delivery deadlines by piggybacking on period beacons. Thanks to the broadcast nature of V2V communications, the demands of multiple vehicles can be fulfilled by one transmission.

reserved for automated driving services is limited. Meanwhile, the onboard storage resources of vehicles are used as distributed caches, and the cache capacity contributed by vehicles in an area grows linearly with the density of vehicles. Besides the cellular radio interface, we consider that each autonomous vehicle also has a dedicated short-range communication (DSRC) interface, and the vehicles can share cached contents with other vehicles using DSRC.

The Coordination of Edge Intelligence

Edge intelligence, in this article, is a term for the computing capability at BSs and autonomous vehicles. Edge intelligence is able to perform data processing, analytics, and decision making. To illustrate the basic procedures for edge intelligence to coordinate content delivery under the proposed architecture, we first consider that the edge server is responsible for the scheduling of content delivery. The service requests submitted by autonomous vehicles are first processed by the edge server. The requests can be satisfied locally if the requested contents are already cached at BSs and/or nearby vehicles. Otherwise, the requests are handed over to the remote cloud server. There is a gap between the large volumes of service content and the limited cache capacity at BSs. To bridge the gap and improve cache performance, the edge server should determine what to cache at a BS. Since the service contents are constantly updated, the cached replicas at the BS should keep pace with their originals in the cloud. To this end, the edge server needs to notify the cloud server of the caching behavior at the BS. Once a cached content becomes out of date, the cloud server will push an up-to-date replica to the edge server. In addition, vehicles only cache the contents they are going to consume, so the caching can be regarded as content prefetching and will not cause any extra storage overheads. To reduce the wireless bottleneck of cellular networks, the edge server will not directly push all the required contents to vehicles. Generally, a content c is divided into data chunks $\{d_1, d_2, ..., d_n\}$ that are all of the same length l (except for the last chunk if l does not divide the content length). The edge server only injects data chunks to some selected vehicles and makes full use of vehicular networks to further disseminate the content.

To initiate content sharing, each vehicle can inform neighbors of its own service demands and corresponding delivery deadlines by piggybacking on period beacons. Thanks to the broadcast nature of V2V communications, the demands of multiple vehicles can be fulfilled by one transmission. To offload more data traffic from cellular networks, the vehicles should cooperatively determine who should share cached contents and what to share. Moreover, to make the edge server aware of the content delivery progress, vehicles should acknowledge the edge server when they receive each content from other vehicles. Lastly, the edge server will inject all remaining data chunks to the vehicles with impending delivery deadlines.

Key Research Challenges

Driving safety, comfort, and efficiency largely depend on timely and complete content delivery for online automated driving services. In this section, we discuss the content placement challenges at the wireless edge and the content sharing challenges among vehicles, respectively.

Content Placement at the Wireless Edge

The contents of automated driving services have large data volume. However, the cache space at BSs specific to automated driving services is limited, and only some of the service contents can be cached. Content placement policy will have a great impact on cache performance, which is evaluated by cache hit ratio. A higher cache hit ratio means a larger reduction of duplicate transmissions on backhaul as well as end-to-end delay. The best content placement policy is to place the most popular content in the cache [7]. The popularity of a content is the ratio of the number of requests for the content to the total number of requests. However, we notice that *the content popularity of automated driving services varies with time as a result of the dynamic traffic flow*. Conventional online caching policies, such as least recently used (LRU) and least frequently used (LFU), may not perform well for automated driving services because they usually lag behind the changes in real popularity. In other words, recent content popularity cannot accurately reflect the popularity in the near future. To optimize the cache hit ratio and reduce the backhaul bottleneck, the edge server should have the ability to predict the short-term content popularity. Thanks to the location-dependent nature of automated driving services, the content popularity may be predictable by revealing the potential patterns of service demands.

Caching at vehicles can facilitate the content sharing among vehicles and alleviate the burden of cellular networks. Additionally, *the location-dependent nature of automated driving services imposes strict spatial constraints on content delivery*. The service content should be completely fetched before the space-constrained deadline; otherwise, the quality of automated driving will deteriorate. It is challenging for the edge server to optimize (minimize) the data chunks transmitted through cellular networks while meeting the content delivery deadlines. Both the dynamic topology of vehicular networks and the spatial distribution of data chunks have a great impact on content dissemination speed [11]. To make the caching more beneficial to the V2V dissemination, the edge server should inject/place appropriate data chunks to appropriate vehicles at appropriate times. Intuitively, the edge server supervises the progress of content dissemination and promptly injects data chunks to some influential vehicles to boost the content dissemination. Influential vehicles are those that can meet more content demands of other vehicles. Therefore, it is critical to find

these influential vehicles in highly dynamic vehicular networks.

Content Sharing through Vehicular Networks

As co-located vehicles usually have shared content demands, the performance of content delivery as well as the quality of automated driving services will benefit from the broadcast nature of V2V communications. Although carrier sense multiple access with collision avoidance (CSMA/CA) is applied in 802.11p, the request to send/clear to send (RTS/CTS) handshake is disabled in broadcast mode. As a result, V2V broadcast suffers from a severe hidden terminal problem, which causes potential collisions. A medium access control (MAC) mechanism is needed to avoid the simultaneous broadcasts of multiple vehicles colliding at receivers.

The space-constrained deadlines make the content demands possess different urgency levels, even for vehicles that require the same content. Therefore, in a particular area, broadcasting different data chunks will have different gains for offloading the cellular traffic. To maximize the total amount of data transmitted through vehicular networks, which content to broadcast, when, and by which vehicle should be carefully planned. As the edge server knows the real-time positions and content demands of the associated vehicles, it can coordinate their transmissions to maximize the transmission gain and ensure no collisions among associated vehicles. However, a single edge server cannot realize the collisions of vehicular broadcasts across BSs. Frequent interactions between edge servers in adjacent BSs are needed to completely avoid the collisions. Moreover, the sophisticated control messages between the edge server and its associated vehicles will cause extra overhead for cellular networks. Therefore, it is better for vehicles to distributedly and cooperatively determine their content sharing schemes.

Potential Solutions and Application

In this section, we first present potential solutions to the above-mentioned research challenges, which include the content placement at the wireless edge and the content sharing through vehicular networks. Then we evaluate the proposed solutions using a real trace set in Beijing as well as a synthetic trace set.

Content Placement at the Wireless Edge

The popularity of service contents should be predicted to optimize the caching performance (i.e., cache hit ratio) at BSs. As automated driving services are location-dependent, the content demands of vehicles largely depend on their travel paths. Fortunately, urban traffic presents obvious patterns [13], which means that the content demands also have potential patterns. For example, the content request behavior may be correlated with the hour of the day, day of the week, weather, and so on. The cumulative content request logs can be used to learn content demand patterns and predict short-term content popularity. Time series analysis is widely used to model the data points taken over time that have an internal structure (e.g., autocorrelation, trend, or seasonal variation). Therefore, the time series analysis can be used to model the content requests. Furthermore, the content request time series possesses a seasonal component that repeats every week. We adopt a seasonal auto-regressive integrated moving average (SARIMA) model. The model makes full use of the weekly relationship between observations for the same day of the week in successive weeks. Once the content demand popularity is forecast, the edge server can greedily cache the most popular contents to improve the cache hit ratio. In addition, tools from machine learning can also be used to predict the content popularity. For example, an artificial neural network (ANN) can take into account features such as the hour of day, day of the week, and content requests in previous intervals and nearby regions.

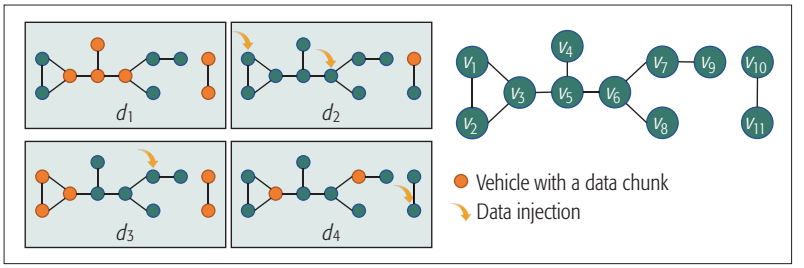
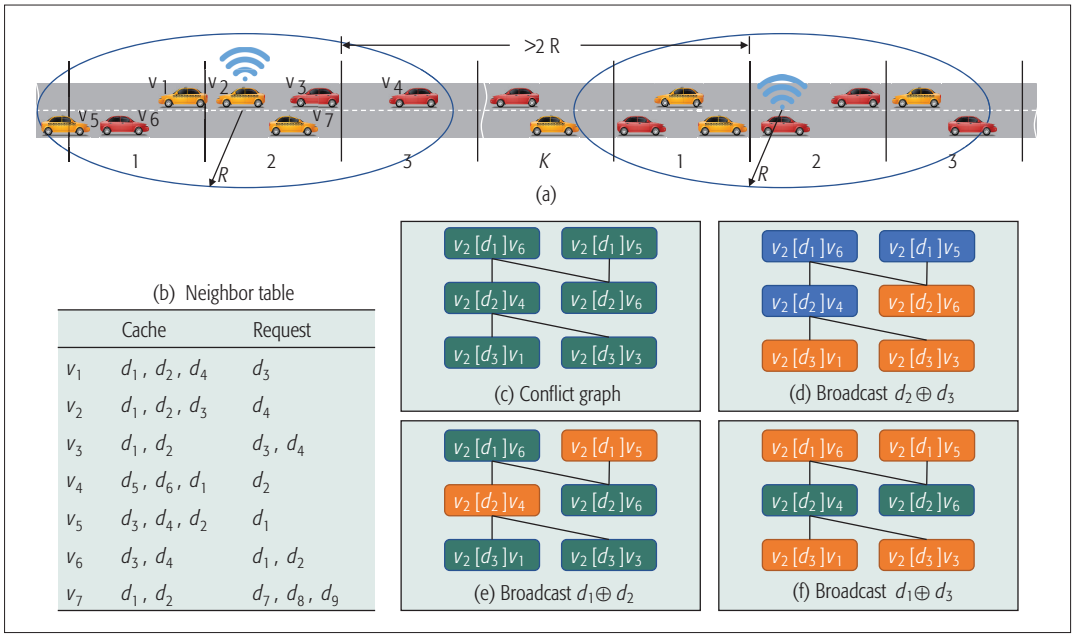


FIGURE 2. A snapshot of the contact graph $\mathtt{CTG}_c$ and the spatial distribution of content $c = \{d_1, d_2, d_3, d_4\}$. The injection approach of the edge server to achieve two-hop domination is shown.

As for the content placement at vehicles, it is determined by the influence of vehicles on content dissemination. By collecting periodic position reports[3] from vehicles, the edge server can construct a contact graph to represent the snapshot of communication opportunities between vehicles. Then the contact graph can be used to derive a proper content placement solution. Let $\mathcal{V}$ be the set of vehicles associated with the BS and $\mathtt{CTG}_c = (\mathcal{V}_c, \mathcal{E}_c)$ be the contact graph whose vertex set $\mathcal{V}_c \subseteq \mathcal{V}$ consists of all vehicles requiring content c. For two vertices $v_1, v_2 \in \mathcal{V}_c$, an edge $(v_1, v_2) \in \mathcal{E}_c$ exists if and only if v_1 and v_2 are in the transmission range of each other. An example of the contact graph is shown in Fig. 2. Moreover, by gathering the acknowledgments from vehicles, the edge server knows the distribution of data chunks on the contact graph. The edge server injects data chunks $\{d_1, d_2, ..., d_n\}$ of content c according to the contact graph $\mathtt{CTG}_c$. A data chunk should be injected to the vehicles that are influential enough to meet the demands of other vehicles. To find these influential vehicles, we can utilize the dominating set of the contact graph. A k-hop dominating set (kHDS) of $\mathtt{CTG}_c$ is a subset $\mathcal{D}_c \subseteq \mathcal{V}_c$ such that every vertex in $\mathcal{V}_c \setminus \mathcal{D}_c$ is within the k-hop neighborhood of some vertex in $\mathcal{D}_c$. For instance, $\{v_5, v_9, v_{10}\}$ is a two-hop dominating set of the contact graph in Fig. 2. To reduce the cellular traffic, the edge server prefers to inject each data chunk to a k-hop minimum dominating set (kHMDS). However, it is NP-hard to find a kHMDS for a general graph. A tree-based heuristic method [14] can be exploited, which extracts a spanning tree of the graph using breadth-first search and finds the kHMDS for the tree in polynomial time. In addition, the edge server is supposed to periodically check the progress of content delivery. When the vehicles

[3] Generally, autonomous vehicles rely on off-board/online navigation services. Their planned paths can be known to the edge server, and their real-time positions can be estimated using the planned paths and dynamic traffic conditions. Therefore, the edge server is able to construct the contact graph with low-frequency position reports, which only introduces limited overhead.



	Cache	Request
v_1	d_1, d_2, d_4	d_3
v_2	d_1, d_2, d_3	d_4
v_3	d_1, d_2	d_3, d_4
v_4	d_5, d_6, d_1	d_2
v_5	d_3, d_4, d_2	d_1
v_6	d_3, d_4	d_1, d_2
v_7	d_1, d_2	d_7, d_8, d_9

FIGURE 3. Content sharing through vehicular networks: a) SDMA-based MAC protocol; b) neighbor table of v_2; c) conflict graph of v_2; d)–f) possible coded broadcast schemes of v_2.

By collecting the periodic position reports from vehicles, the edge server can construct a contact graph to represent the snapshot of communication opportunities between vehicles. Then the contact graph can be used to derive a proper content placement solution.

that have cached data chunk $d \in c$ cannot k-hop dominate the contact graph $\mathtt{CTG}_c$, the edge server should find a kHMDS for the undominated component of $\mathtt{CTG}_c$ and directly inject d to these selected vehicles. For example, the contact graph in Fig. 2 has been two-hop dominated by vehicles that have cached d_1. The edge server can inject d_2 to v_1 and v_6, inject d_3 to v_7, and inject d_4 to v_{11} to achieve two-hop domination and boost the content delivery.

Content Sharing through Vehicular Networks

A MAC protocol is needed to achieve spatial reusability of the wireless transmission medium, that is, multiple vehicles can transmit simultaneously without interference. Taking advantage of the locality of service demands, we use a space-division multiple access (SDMA)-based MAC, where it is actually the service contents instead of the vehicles that contend for the medium.[4] The basic idea of SDMA is to partition roads into small divisions and assign each division a time slot. Let the maximum transmission range of DSRC be R. The divisions assigned with the same time slot should be separated by a large enough distance (i.e., greater than $2R$). In addition, the vehicles in the same division are within the transmission range of each other. A K-slot schedule is illustrated in Fig. 3a. As for the space division information, it can be a special layer of the HD map. Each vehicle can use the combination of GPS and HD map to accurately localize itself in a division and use the corresponding time slot. However, the vehicles in the same division collide with each other. To maximize the gain of V2V dissemination, the vehicles in the same division should cooperatively determine what to transmit, and by which vehicle.

The coded broadcast is more efficient than uncoded broadcast. We can exploit index coding to forward multiple data chunks in a single transmission. For example, as shown in Fig. 3b, vehicles v_2, v_4, and v_5 all require data chunks d_1 and d_2. In addition, v_2 has cached d_1 and d_2, v_4 has cached d_1, and v_5 has cached d_2. Both v_4 and v_5 are within the transmission range of v_2. In this case, v_2 only needs to broadcast $d_1 \oplus d_2$ in a single transmission, where $\oplus$ represents the bit-wise XOR operation. Then v_4 can obtain $d_2 = d_1 \oplus (d_1 \oplus d_2)$, and v_5 can obtain $d_1 = d_2 \oplus (d_1 \oplus d_2)$. To seek coded broadcast opportunities, a conflict graph is constructed for each source vehicle. We use v_1 $[d_1]$ v_2 to denote a potential transmission in which v_1 transmits d_1 to v_2. Two transmissions are conflicting if they cannot be performed in a single coded broadcast. Specifically, two transmissions v_1 $[d_1]$ v_2 and v_1 $[d_2]$ v_3 are conflicting if $d_1 \neq d_2$ and at least one of the following conditions holds:
1. d_1 has not been cached in v_3.
2. d_2 has not been cached in v_2.

In other words, if two conflicting transmissions are combined into a coded broadcast, at least one of the receivers cannot decode its required data chunk. Particularly, to identify the conflicting transmissions, the vehicles are required to broadcast the catalog of their cached data chunks by piggybacking on period beacons. Let $\mathtt{CFG}_v$ be the conflict graph of source vehicle v. The vertex set $\mathcal{U}$ of $\mathtt{CFG}_v$ consists of all potential transmissions of v, and an edge exists between two vertices if they are conflicting. In a conflict graph, two vertices without a direct edge can be satisfied simultaneously. Figure 3c shows a conflict graph, and Figs. 3d–3f present possible coded broadcast schemes, where non-conflicting transmissions are colored orange. For instance, in Fig. 3f, v_2 $[d_1]$ v_5, v_2 $[d_1]$ v_6, v_2 $[d_3]$ v_1, and v_2 $[d_3]$ v_3 can be combined into a single coded broadcast $d_1 \oplus d_3$.

To maximize the gain of coded broadcast, we propose to find the optimal non-conflicting trans-

[4] In our scenario, not all vehicles need to access the medium to broadcast contents. With the proposed SDMA, in a division, only the vehicle holding the most desirable content has the right to access the medium. However, time-division multiple access (TDMA) and code-division multiple access (CDMA) schemes usually assign every vehicle a share of radio resources, and are inefficient in our studied location-dependent vehicular contents sharing scenario.

missions on the conflict graph by solving a maximum weighted independent set (MWIS) problem, which is NP-hard. Let g_u be the gain/weight of transmission u in $\mathtt{CFG}_v$, where g_u is positively correlated with the urgency level of u. A set $\mathcal{I}$ is an MWIS of $\mathtt{CFG}_v$ if any two vertices in $\mathcal{I}$ are not adjacent and the total gain $g = \Sigma_{u\in\mathcal{I}}\, g_u$ is maximized. An approximation method in [15], which outputs an independent set of weight at least $\Sigma_{u\in\mathcal{U}}\, g_u/(\rho_u + 1)$, can be used to solve the MWIS problem, where ρ_u is the degree of u in $\mathtt{CFG}_v$. Specifically, vehicle v computes the metric $g_u/(\rho'_u + 1)$ for each vertex u in the current conflict graph, where ρ'_u is the current degree of u. Then vehicle v finds the vertex with the largest metric, removes all neighbors of the vertex from the conflict graph, and puts the vertex to $\mathcal{I}$. It repeats the above procedures until the conflict graph goes empty. Finally, $\mathcal{I}$ contains the preferred transmissions for vehicle v. Vehicle v can combine all the transmissions in $\mathcal{I}$ into a single coded broadcast. However, in a division, only one vehicle can use the time slot to broadcast. By finding the MWIS of the conflict graph, each vehicle in the division knows what it prefers to broadcast. A vehicle with a larger gain g is expected to have priority to broadcast in the time slot. To this end, a backoff mechanism can be used for vehicles in the same division to compete for the time slot. Finally, the vehicle with the largest gain should have the shortest backoff and win the right to access the time slot.

Performance Analysis

The performance of caching service contents at BSs is validated using real taxicab traces in Beijing, China. The taxicab traces contain the GPS trajectories recorded by over 12,000 taxicabs in November 2012. We select a commercial area of about 2 × 2 km^2 to simulate service requests from vehicles. The data in the first three weeks are used to train the SARIMA model. Then the content popularity of the fourth week is forecast using the trained model. Figure 4 shows the caching performance for the content requests in the fourth week. It can be seen that the popularity-prediction-based method outstrips LRU and LFU for different cache sizes.

As the temporal resolution of the above taxicab dataset is unable to simulate the communications between vehicles, we use a synthetic trace set generated by the Simulation of Urban Mobility (SUMO) software to validate the performance of vehicular caching and content sharing. The scenario is a 1.8 × 1.8 km^2 area with a grid road network (total length of roads is 10.8 km). We make realistic assumptions on the channel models for the cellular links and the V2V links. The cellular data rate is set to 30 Mb/s, while the vehicular data rate is set to 12 Mb/s. The maximum radio range of vehicles to establish reliable communications is 200 m. In the simulations, vehicles do not transceive data simultaneously on the cellular interface and the vehicular interface. Therefore, the simulation results can be applied to other techniques that enable V2V communications (e.g., LTE-V2X and 5G-V2X). The amounts of data offloaded to the vehicular networks for different traffic densities and different content sizes are shown in Fig. 5. It can be seen that more service data can be offloaded to vehicular networks when more vehicles are involved. Although the proportion of data traffic offloaded to vehicular networks declines as the content size grows, the total amount of offloaded data still increases.

For the contents that are not cached at BSs, the edge server can prefetch these contents from cloud server for the coming vehicles. To make the prefetching effective, the edge server should take into account the mobility of vehicles, network conditions and its available storage capacity.

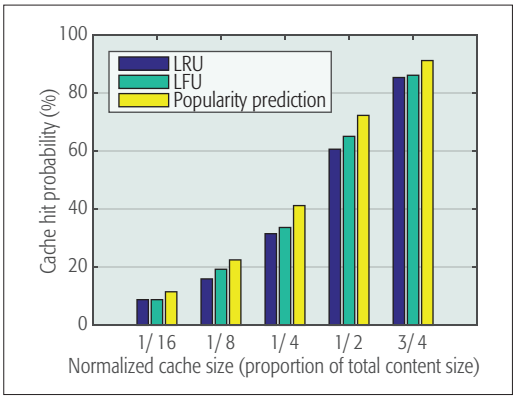


FIGURE 4. Performance of caching at BSs.

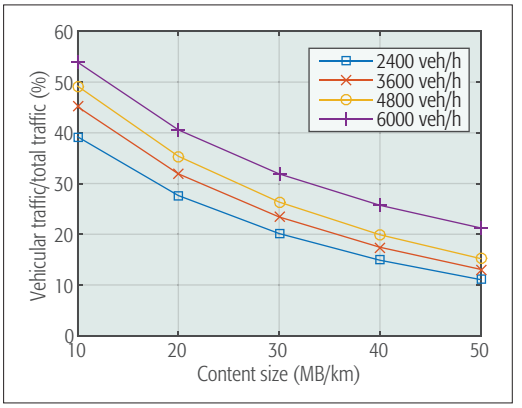


FIGURE 5. Performance of caching and content sharing among vehicles.

Conclusion and Future Directions

In this article, we have studied the key issues of edge-assisted content delivery for automated driving services. We have demonstrated that content caching at the wireless edge and content sharing through vehicular networks can significantly reduce resource utilization while improving the quality of automated driving services. Specifically, time series analysis has been utilized to forecast the service content demands and cache the popular contents at BSs accordingly. Dominating set has been exploited to cache contents at influential vehicles. An SDMA-based MAC and index coding enabled broadcasting have been proposed to improve the efficiency of content sharing among vehicles. To fully leverage the power of the wireless edge, further research issues are discussed as follows.

Edge-assisted service content prefetching: Some automated driving related services are delay-sensitive. An edge server can prefetch contents that are not cached at BSs from a cloud

To process data efficiently and accurately, autonomous vehicles can offload certain computing tasks to the edge server. The vehicles should determine which computing task to offload based on the communication cost, the QoS requirements of the task, network conditions, and the available computing capability on the edge server.

server for coming vehicles. To make prefetching effective, the edge server should take into account the mobility of vehicles, network conditions, and its available storage capacity.

Edge-assisted data gathering and utilization: For automated driving, the service contents are usually generated and consumed locally. If the service contents are managed by the edge server instead of the remote cloud sever, the service latency can be reduced and the backhaul resource can be saved. Specifically, the edge server allocates sensing tasks to associated vehicles and gathers sensing data from them. The edge server is responsible for guaranteeing the accuracy, completeness, timeliness, validity, and consistency of the sensing data.

Edge-assisted computing: The computing capability of the edge server is more powerful than that of vehicles. To process data efficiently and accurately, autonomous vehicles can offload certain computing tasks to the edge server. The vehicles should determine which computing task to offload based on the communication cost, the QoS requirements of the task, network conditions, and the available computing capability on the edge server.

Acknowledgment

This work is supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant No. 2016ZX03001025-003, the Natural Science Foundation of China under Grant No. 91638204, the Special Fund for Beijing Common Construction Project, and the Natural Sciences and Engineering Research Council of Canada.

References

[1] D. Watzenig and M. Horn, *Automated Driving: Safer and More Efficient Future Driving*, Springer, 2016.
[2] D. Tian *et al.*, "Robust Energy-Efficient MIMO Transmission for Cognitive Vehicular Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 6, June 2016, pp. 3845–59.
[3] H. Zhou *et al.*, "ChainCluster: Engineering A Cooperative Content Distribution Framework for Highway Vehicular Communications," *IEEE Trans. Intell. Transp. Sys.*, vol. 15, no. 6, Dec. 2014, pp. 2644–57.
[4] M. Amadeo, C. Campolo, and A. Molinaro, "Information-Centric Networking for Connected Vehicles: A Survey and Future Perspectives," *IEEE Commun. Mag.*, vol. 54, no. 2, Feb. 2016, pp. 98–104.
[5] J. Ren *et al.*, "Serving at the Edge: A Scalable IoT Architecture Based on Transparent Computing," *IEEE Network*, vol. 31, no. 5, Sept./Oct. 2017, pp. 96–105.
[6] E. Zeydan *et al.*, "Big Data Caching for Networking: Moving from Cloud to Edge," *IEEE Commun. Mag.*, vol. 54, no. 9, Sept. 2016, pp. 36–42.
[7] K. Poularakis and L. Tassiulas, "Code, Cache and Deliver on the Move: A Novel Caching Paradigm in Hyper-Dense Small-Cell Networks," *IEEE Trans. Mobile Comp.*, vol. 16, no. 3, Mar. 2017, pp. 675–87.
[8] M. Ji, G. Caire, and A. F. Molisch, "Wireless Device-to-Device Caching Networks: Basic Principles and System Performance," *IEEE JSAC*, vol. 34, no. 1, Jan. 2016, pp. 176–89.
[9] Z. He, D. Zhang, and J. Liang, "Cost-Efficient Sensory Data Transmission in Heterogeneous Software-Defined Vehicular Networks," *IEEE Sensors J.*, vol. 16, no. 20, Oct. 2016, pp. 7342–54.
[10] K. Liu *et al.*, "Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as a Software Defined Network," *IEEE/ACM Trans. Net.*, vol. 24, no. 3, June 2016, pp. 1759–73.
[11] Q. Yuan *et al.*, "Space and Time Constrained Data Offloading in Vehicular Networks," *Proc. IEEE HPCC*, Sydney, Australia, 2016, pp. 398–405.
[12] T. Wang *et al.*, "Dynamic Popular Content Distribution in Vehicular Networks Using Coalition Formation Games," *IEEE JSAC*, vol. 31, no. 9, Sept. 2013, pp. 538–47.
[13] Y. Zheng, "Trajectory Data Mining: An Overview," *ACM Trans. Intell. Sys. Tech.*, vol. 6, no. 3, May 2015, pp. 1–41.
[14] S. Datta *et al.*, "The Habits of Highly Effective Researchers: An Empirical Study," *IEEE Trans. Big Data*, vol. 3, no. 1, Mar. 2017, pp. 3–17.
[15] S. Sakai, M. Togasaki, and K. Yamazaki, "A Note on Greedy Algorithms for the Maximum Weighted Independent Set Problem," *Discrete Applied Mathematics*, vol. 126, nos. 2–3, Mar. 2003, pp. 313–22.

Biographies

Quan Yuan [S] (yuanquan@bupt.edu.cn) received his B.S. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT), China, in 2011. He is currently a Ph.D. candidate at the State Key Laboratory of Networking and Switching Technology, BUPT. His current research interests include mobile computing, crowdsensing, and vehicular networks.

Haibo Zhou [M] (h53zhou@uwaterloo.ca) received his Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, China, in 2014. From 2014 to 2017, he worked as a postdoctoral fellow with the Broadband Communications Research Group, ECE Department, University of Waterloo, Canada. Currently, he is an associate professor with the School of Electronic Science and Engineering, Nanjing University. His research interests include resource management and protocol design in cognitive radio networks and vehicular networks.

Jinglin Li [M] (jlli@bupt.edu.cn) received his Ph.D. degree in computer science and technology from BUPT. He is currently an associate professor at the State Key Laboratory of Networking and Switching Technology, BUPT. His research interests are mainly in the areas of network intelligence, mobile Internet, the Internet of Things, and the Internet of Vehicles.

Zhihan Liu (zhihan@bupt.edu.cn) received his M.S. degree in computer science and technology from BUPT in 2004. He is currently a researcher at the State Key Laboratory of Networking and Switching Technology, BUPT. His research interests include the Internet of Vehicles, the Internet of Things, and mobile Internet.

Fangchun Yang [SM] (fcyang@bupt.edu.cn) received his Ph.D. degree in communications and electronic systems from BUPT. He is currently a professor at the State Key Laboratory of Networking and Switching Technology, BUPT. He is a Fellow of the IET. His current research interests include network intelligence, service computing, and the Internet of Vehicles.

Xuemin (Sherman) Shen [F] (sshen@uwaterloo.ca) is a University Professor, Department of Electrical and Computer Engineering, University of Waterloo. He serves/has served as the Editor-in-Chief for the *IEEE Internet of Things Journal*, *IEEE Network*, *Peer-to-Peer Networking and Applications*, and *IET Communications*. He is an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, and a Royal Society of Canada Fellow.