ARVE: Augmented Reality Applications in Vehicle to Edge Networks

Pengyuan Zhou
University of Helsinki, Finland

Wenxiao Zhang
Hong Kong University of Science and Technology, Hong Kong

Tristan Braud
Hong Kong University of Science and Technology, Hong Kong

Pan Hui
Hong Kong University of Science and Technology, Hong Kong
University of Helsinki, Finland

Jussi Kangasharju
University of Helsinki, Finland

ABSTRACT

Vehicular communication applications, be it for driver-assisting augmented reality systems or fully driverless vehicles, require an efficient communication infrastructure for timely information delivery. Centralized, cloud-based infrastructures present latencies too high to satisfy the requirements of emergency information processing and transmission. In this paper, we present a novel Vehicle-to-Edge (ARVE) infrastructure, with computational units co-located with the base stations and aggregation points. Embedding computation at the edge of the network allows to reduce the overall latency compared to vehicle-to-cloud and significantly trim the complexity of vehicle-to-vehicle communication. To demonstrate the efficiency of our solution, we apply these principles on an augmented reality head-up display. In this use case, vehicular communication is exploited to connect vehicle's vision, and quickly propagate emergency information. ARVE is a general system framework, applicable to many practical scenarios. Our preliminary evaluation shows that ARVE noticeably decreases transmission latency with reasonable capital expenditure.

CCS CONCEPTS

• **Applied computing** → *Service-oriented architectures*; • **Computer systems organization** → *Distributed architectures*;

KEYWORDS

Edge computing, vehicle network, augmented reality

1 INTRODUCTION

Connected automated driving has recently become closer to being a reality. In 2018, California and Shanghai authorized the deployment of autonomous vehicles on public roads for testing purposes [3, 21]. Vehicular communication systems play a key role in sharing information between vehicles and roadside infrastructure units (RSU) . Use cases include emergency warning system for vehicles, cooperative adaptive cruise control, collision warning etc. Current solutions focus on three types of communication: vehicle-to-vehicle (V2V), vehicle-to-cloud (V2C), and vehicle-to-roadside infrastructure (V2I) [8, 10]. Although these solutions fulfill basic demands, *efficiently sharing complex and large volumes of data among vehicles at scale* remains a challenge.

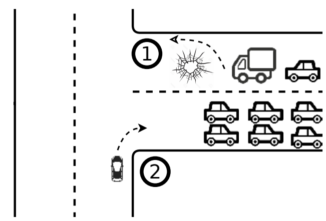

Figure 1: Common connected vehicles scenarios

Figure 1 illustrates two related scenarios: (1) The leading truck encounters an unexpected pothole. The truck notifies the following cars to avoid a potential accident. (2) Congested traffic is out of sight for cars planning to take the road on the right. Once aware, these cars will choose a better path. In V2C, even though the leading truck immediately uploads the captured pothole information, the combined latency of transmission, processing and distribution may be too high for the following vehicles to avoid it. Similarly, the connection establishment time of V2V communication with the complexity of forwarding information in a constantly varying crowd of nodes can lead to vehicles having only partial knowledge of the situation. V2I provides better data distribution; however, sharing accurate emergency information entails nontrivial computation and coordination. Roadside infrastructures should therefore integrate computing features for fast and reliable emergency information propagation.

Edge computing facilitates latency-sensitive workloads by performing data processing in Edge Servers (ESes) located close to the user. The gain in latency provided by edge computing can be considerable. In Table 1, we measured the round trip latency for various servers through an LTE network: the first pingable IP, noted as *Edge*, a cloud server located in the same city and another server 1000 km away. Unsurprisingly, the latency to the closest server is half the round trip time to the furthest cloud server. Moreover, the ES presents a 20% improvement compared to the nearest cloud

MECOMM'18, August 20, 2018, Budapest, Hungary

ACM ISBN 978-1-4503-5906-1/18/08…$15.00
https://doi.org/10.1145/3229556.3229564

Edge	Nearby Cloud	Far Cloud
19.9 ms	24.9 ms	52.4 ms

Table 1: Average network round-trip latency over LTE to different targets

server, making it an attractive location for latency-sensitive applications.

We propose to *use vehicle-to-edge (V2E) to enhance vehicular communications*. Our design, ARVE, is a framework designed to be independent of the actual protocols, in order to allow it to apply equally to current as well as future networks. We choose to apply those principle to Connected Vehicle Views (CVV), a concrete use case of ARVE (see Section 3 for a detailed discussion of this use case).

Our contribution is threefold:

- We present the design of ARVE, which equips RSUs with computation and cache capacity.
- Concrete application of ARVE to CVV using Augmented Reality Head-up Display (ARHUD). This use case displays the advantages of ARVE while scaling the problem of vehicle vision from a network perspective.
- We present a preliminary evaluation to show that ARVE offers noticeable performance improvements with reasonable expenditure in infrastructure.

The rest of paper is structured as follows. After a review of related work in section 2, we introduce the ARHUD use case in section 3. We then describe the system design, implementation and communication process of ARVE in section 4. The potential network protocols are discussed in section 5. Preliminary evaluation results are given in section 6. We conclude the paper in section 7.

2 RELATED WORK

Emerging technologies enable various functions for autonomous vehicles but also bring new challenges. Different protocols and standards, i.e., Direct Short Range Communication (DSRC), Device to Device (D2D) and 5G, improve data transmission [1, 15, 16]. However, the large volumes of data will challenge current computation resource deployments and risk making them bottlenecks. Edge computing as a solution to bring computation close to user, has attracted attention, such as [12] which explores an integration of 5G, SDN, MEC and vehicular network. Uncoordinated strategies for edge service placement have been investigated in [2] and the results have shown that they work well for this problem. Paper [11] discusses the direction of utilizing Information Centric Network and MEC for connected vehicles. Meanwhile, the fundamental issues, i.e., architecture design, communication process, network protocols and implementation concerns are largely yet to be explored. Efforts on developing vehicular applications have achieved some results [14, 19], but without an improvement from system and networking point of view, those applications face difficulties to scale in realistic situation.

3 USE CASE: CONNECTED ARHUD

One of the main concerns in the automotive world is safety. According to the 2015 National Motor Vehicle Crash Causation Survey by National Highway Traffic Safety Administration (NHTSA), 93% of crashes are attributed to drivers, of which, around 74% are due to erroneous recognition or decision [20]. However, autonomous vehicles can also get "confused" easily. For instance, GM Cruise autonomous cars sometimes slow down or stop if they see a bush on the roadside [13] and similar issues exist in lane changes. As such, the intervention of a human driver is still critical for safety. To assist the driver, vehicles are outfitted with sensors and display devices which provide valuable information to the driver, such as about environmental condition and the driver's driving habits. The most common display method is a heads-up-display (HUD).

In recent years, augmented-reality (AR) HUDs have attracted both academic and industrial attention [14, 22]. AR can embed 3-D views of the information into the rendering background on the HUD, enabling accurate obstacle recognition and emergency notification. Previous work has put effort on matching the embedded information with the real environment, cognitive usability, visibility, among others [17, 18]. However, connecting vehicle views via ARHUD remains a challenge. Recently, the authors in [19] explored how to share vision between two vehicles. Although the work proposed solutions for basic view transformation, it is still not enough to connect vehicle vision at scale under realistic concerns of bandwidth, latency and computational resources.

Challenges are multiple: (1) A crowd-sourced map which is the combined 3D point cloud from the independent real-time views of the connected vehicles, acts as a reference coordinate system to localize incidents. This map is too voluminous for both real time generation and transmission, hence we need to develop additional mechanisms to address its proper generation and maintenance. (2) Proper network protocol stack needs to be explored. There are several protocols proposed and tested in vehicular network. An integrate protocol stack fit for different applications is still beingless. (3) Privacy and security concerns may arise in distributed V2V communications.

In this paper, we design ARVE, a framework designed to enable Connected Vehicle Views (CVV) where nearby vehicles are able to share their views, assisted by the edge components, and form a more holistic view of their current situation. This enables fast distribution of critical information, i.e., obstacle detection, emergency report and collision notification. While we use CVV as an example, ARVE can serve any similar application, which requires computation and short latency.

4 SYSTEM DESIGN

In this section, we describe the ARVE design. First, we explain our system architecture and describe the major communication processes in the system. Then, we propose an implementation scheme and present how to apply it to CVV.

4.1 System Architecture

We now introduce the ARVE architecture model. It has three key elements: environment, vehicles and edge servers. Environment includes the background road network, roadside buildings, infrastructures and pedestrians, etc., while the others represent the computational elements in the system. Figure 2 depicts our system architecture in which ESes are distributed hierarchically in two tiers. Some are co-located with base stations[1], while others are co-located with aggregation points. We name the former Tier1 Edge Server (T1 ES) and the latter are Tier2 Edge Servers (T2 ES).

The edge layer is the amalgamation of T1 and T2 ESes and is where ARVE operates. Edge layer communicates with vehicles and RSUs via nearby radio access network, and transmits data with remote cloud for synchronization. Each T1 ES has a range over the area covered by its connected macrocell and surrounded small cells. The hierarchical design of the edge layer allows applications with different requirements to be processed differently for better performance. T2 ES collects data from multiple areas (multiple T1 ESes) to provide larger scale of service and data backup, e.g., to improve traffic flow by sending cruise control messages. T1 ES, which is closer to vehicles, serves applications with higher latency-sensitivity, e.g., emergency notifications.

4.2 ARVE Basic Operation

The basic operation of ARVE relies on the generation of a map around the vehicle, to enable awareness of the surroundings. The generation of the crowd-sourced map involves multiple steps. First. for each vehicle, we generate a 3D point cloud of the road in front of the vehicle using visual sensors present in the vehicle (e.g., LiDAR, RGBD camera). Then. the point clouds from multiple connected vehicles are transmitted to the edge server and combined into a 3D street view. Finally, the combined point cloud of the street is transmitted back to the vehicles, and each vehicle can display the street view according to its own position, so that the driver would be able to see the extended view of the whole street on the HUD.

4.3 Communication Process

Next we describe the six basic steps (marked in Figure 2) in ARVE: neighbor notification, data processing, transmission, dissemination, aggregation, and upload. The exact details depend on the actual application; here we use an emergency notification application to showcase the communication process:

(1) **Neighbor notification:** The nearest vehicles require the fastest notification of emergencies. Therefore, upon emergency detection, a vehicle needs to warn its neighbor vehicles immediately, by sending simple notification via V2V. The notification includes only critical messages, e.g., name/type and coordinates of the emergency, to minimize V2V bandwidth usage and latency. The V2V notification is relayed until reaching a predefined maximum number of hops. Meanwhile, the vehicle sends a detailed report to nearest T1 ES via V2I. The report includes collected sensor and camera data with only the minimal, necessary data compression.
(2) **Data processing:** Once a T1 ES receives a report, it processes the data and caches it for passing on to later passing vehicles. As discussed in [19], sharing views of incidents among vehicles is nontrivial. ESes maintain and update local map in real time, by collecting data from passing vehicles and synchronize it with a cloud data center. With the up-to-date map, T1 ESes serve as calibration points which map the reported incident onto absolute coordinates and notify nearby vehicles more efficiently.
(3) **Data transmission:** The maintained map or other data, e.g., emergency or congestion information, can be transmitted between ESes via wired or wireless channels.
(4) **Data dissemination:** Upon data updates, ESes disseminate data to vehicles in their coverage areas.
(5) **Data aggregation:** T1 ESes aggregate data before sending it to T2 ESes for applications requiring larger amounts of data (naturally assuming aggregation is acceptable for the application).
(6) **Data upload:** T2 ESes synchronize with cloud to update data and enable synchronization across a larger geographical area.

This communication model has several important benefits, namely:

(1) **Neighbor notification** combines two methods of which the simple notification warns closest vehicles with the lowest delay, while the detailed report sends all information for ESes to generate AR data for CVV.
(2) **Cache capacity** of an ES noticeably improves the performance of vehicular communication system. A common scenario is that vehicle detects an anomaly on road without nearby vehicles. The vehicle therefore cannot pass on the notification to other vehicles, but instead must upload it into the cloud. However, for a later vehicle to receive the notification in a timely manner, it needs to get the data all the way from cloud, or be in the coverage area of nearby RSUs when the data is still in transmission. ESes change this shortcoming by caching the data and broadcasting within predefined period, so that later vehicles receive the notification with lower latency.
(3) **Hierarchical edge** enables efficient handling of workloads with different requirement by processing the data at the different tiers, depending on the application requirements..

4.4 Implementation

In terms of implementation, two key issues arise: the deployment and placement of ESes. Deployment refers to the internal implementation of an ES while placement refers to the physical placement of the ES.

Deployment: Our proposal to co-locate ESes with base stations and aggregation points are motivated by existing trends. The MEC standard developed by European Telecommunications Standards Institute (ETSI) proposes to deploy servers at the cellular base station to serve local mobile subscribers with fast response times [4]. Co-location with existing infrastructure also achieves cost-efficiency. For these reasons, T1 ESes co-located with base stations is a straightforward solution. Next, we need to take a look at cellular network deployment in near future to understand the rationale of T1 ES deployment. According to the 2017 survey of Small Cell Forum (SCF), by 2025, new non-residential small cell deployments will reach almost 8.5 million, which is 22 times higher than in 2015 [9]. On the other hand, 5G will also accelerate the deployment of small cells.

[1]Base station in this paper refers to the entity at the edge of the fixed network, e.g., BTS, eNB and gNB etc.

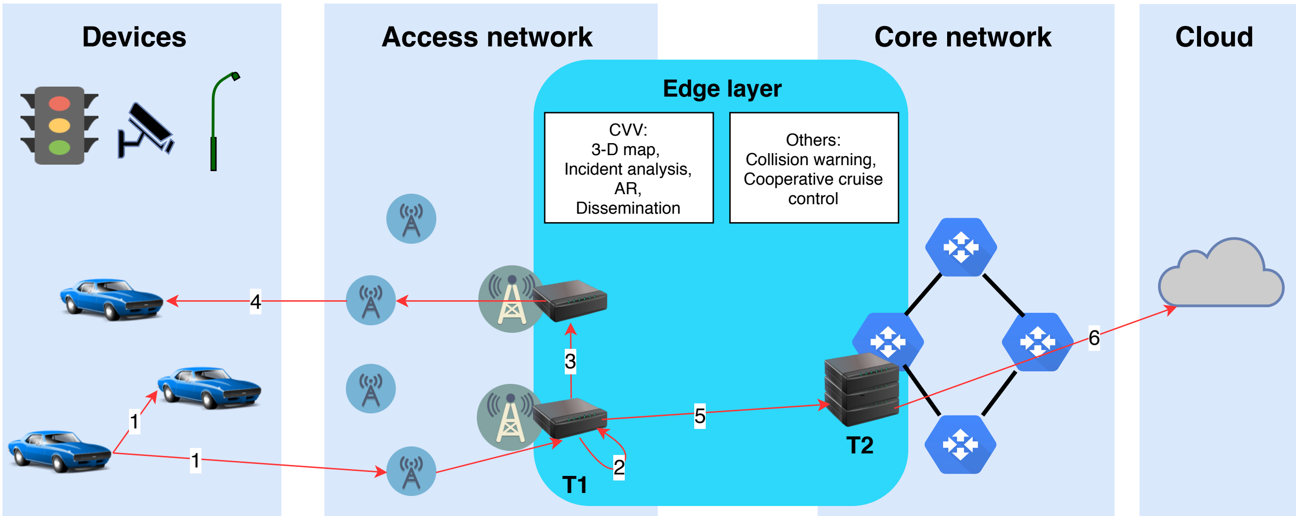

Figure 2: ARVE System Model. The numbers refer to the steps in the communication process (see Section 4.3)

58% of the operators, according to the same survey of SCF, expect to focus primarily on small cells in the first 2-3 years of deploying 5G. However, the number of macrocell seems to grow much slower. According to Nokia, traffic density of a very busy US city increased fourfold from 2004 to 2014, yet the average density of macrocell sites did not change [5]. We conjecture that small cell deployment will increase much faster than macrocell deployment in the near future. As a result, capacities of T1 ESes are facing increasing challenges and therefore we propose to locate the T2 ESes at a higher layer in the network to enable more efficient aggregation and backup.

Placement: To avoid unnecessary investment and complexity, the ESes location should be carefully determined. While T2 ESes are locate typically at aggregation points, which are relatively few in number, locations of T1 ESes have much more candidates, namely the macrocells. Cities like New York have macrocell deployments with 500 m inter-site distance or less [6]. Deploying one T1 ES per macrocell would be excessive in terms of investment, so to improve efficiency, we need to optimize the selection of locations in some manner. Our proposed algorithm includes two steps, namely average traffic clustering and edge capacity assignment. We opt for a hierarchical clustering algorithm since our edge layer already is constructed hierarchically. Edge capacity assignment is solved as a primary facility location problem, where we simply assign edge capacity to each cluster center (both Tier 1 and 2), proportional to its average traffic. The order of edge capacity is calculated through edge server capacity, traffic density, and resource consumption of the application (section 6).

4.5 Use case solution Overview

Now we describe how to implement an ARHUD-based CVV. To solve the challenges described in section 3, we need to implement the following components: (1) **Map maintenance:** Vehicles record the surrounding 3-D features with the coordinates of traversed streets and send to nearest T1 ESes. T1 ESes stitch together the collected segments to form 3-D neighborhood maps. (2) **Incident report:** Once a vehicle detects an incident, it sends the simple notification and detailed report to the nearest vehicle and T1 ES, respectively. (3) **Data process:** The ES extracts the data from the received report and localizes the incident in the map it maintains. The localization can use either the received coordinates or map the observed 3-D features within the map. (4) **Transmission and dissemination:** Meanwhile, the ES forwards the notification to nearby T1 ESes (directly or via T2 ES) and disseminates to vehicles within its coverage. The range of the dissemination area depends on the magnitude of the incident and the coverage area of the ES. (5) **Aggregation:** T2 ES aggregates data from T1 ESes to gather information of larger area. Use cases include, for example, crowd–sourcing the neighborhood maps to build an urban 3-D map or traffic light control. (6) **Synchronization:** ESes synchronize with the cloud periodically or when triggered by specified incidents.

5 NETWORK ISSUES

In this section, we discuss possible network protocols for V2E powered vehicle communication system. ARVE does not have any specific requirements on the networking technologies or protocols that are used. We can accommodate different technologies including cellular, Wi-Fi, D2D and DSRC so that they complement each other to fulfill different kinds of workloads and constitute an integrated networking system. Considering Figure 2 as an example, the device layer includes V2V and V2I communication where DSRC and D2D protocols coexist to provide better performance. Stand-alone D2D (Wi-Fi Direct) and DSRC could support V2V in scenarios even without network coverage. Another D2D protocol, LTE Direct, needs network assist and supports long distance connection. As shown in [19], Wi-Fi Direct has better performance than LTE and higher theoretical throughput than DSRC. However, WLAN chipsets are unlikely to fulfill ad-hoc communication at high speeds which makes them unreliable for vehicle network [7]. Here we propose to use a combination of D2D and DSRC to serve large volumes of data and fast data transmission, respectively. For instance, in our

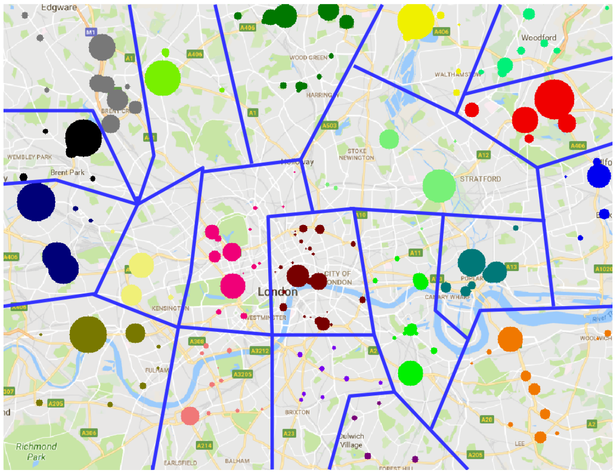

Figure 3: Traffic distribution in London.

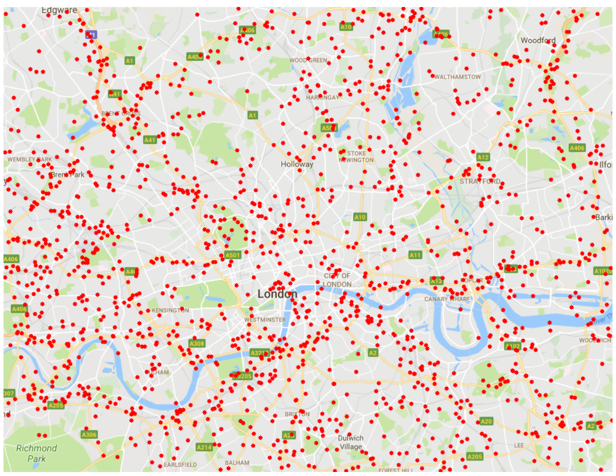

Figure 4: LTE base station (with coverage radius > 3000m) distribution in the selected area of London.

communication process, vehicle sends out the simple notification to closest vehicle by DSRC, while sending the detailed report to nearby ES by D2D. The rest of the system communicates via wired and LTE or 5G network. Today's cell phone connects to internet via cellular or Wi-Fi network, depending on local network coverage and subscription etc. Likewise, vehicular networks should also use multiple complementary protocols to function in different scenarios.

6 PRELIMINARY EVALUATION

In this section, we will present a primary ES placement solution for a CVV application based on ARVE and elaborate the system improvement over current vehicular network.

Data Collection and Analysis: To address the edge server placement problem, we study the base station and traffic distribution pattern in the center area of London as an example. The selected area has a size of 26km * 20km, and we collect the LTE base station location data[2] and traffic volume data[3] that fall into this area according to GPS coordinates.

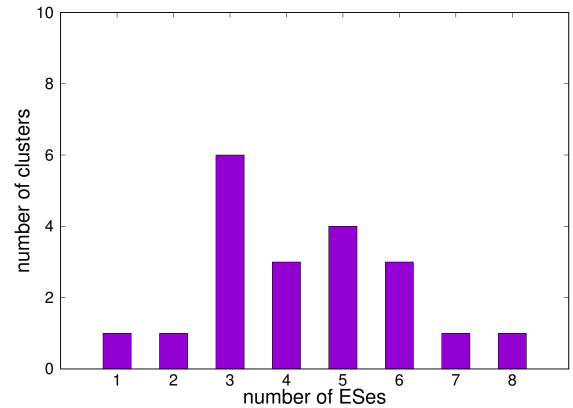

Figure 5: Number of edge servers needed by different areas.

First we cluster the traffic volume data according to their GPS coordinates, and divide the selected area into 20 small areas according to the clustering result. The traffic distribution and area partition results are shown in Figure 3, where each colored dot represents the location of the aggregated traffic, and the different sizes of the dots reflect the different traffic volumes in 12 hours during daytime.

Next we want to see if base stations distribute differently from traffic, to understand if this would influence our co-located ES placement. There are 22041 LTE base stations located within this area, among which 1538 base stations have a coverage radius larger than 3000m, comparable to macrocell. We plot these 1538 base stations on the map, as shown in Figure 4. It can be easily observed that the base stations distribute evenly and reasonably match the amount of traffic in dense areas. As a result, using base stations as deployment points is not going to deviate the ES placement from the actual traffic patterns.

Edge Server Placement: H. Qiu et al. reported a typical Augmented Vehicle Reality system [19], where the AR related processing (e.g., point cloud manipulation) takes 1.337 sec on average. Considering this processing as the AR workload of the edge servers, one edge server is able to handle 2692 requests per hour, that is, serving around 32k vehicles during each daytime.

The edge server placement is correlated with the traffic volume distribution, which is not uniform among the 20 small areas. The numbers of edge servers needed by each area are shown in Figure 5. In total 90 edge servers are needed in the selected center area of London and the largest clusters of ESes have a total of 8 ESes, while the bulk of them contain 3–4 ESes.

Latency Comparison between Edge Server and Cloud Server. Edge servers bring the processing capability to the vicinity of vehicles. The latency of augmented vehicle reality consists of mainly two parts: the data transmission time and the server processing time. The data processing time taken by the edge server and the cloud server would not differ significantly, but the data transmission time is greatly influenced by the transmission distance.

[2]https://unwiredlabs.com
[3]https://data.gov.uk/dataset/gb-road-traffic-counts

As reported in [19], the point cloud data size of the view generated by a 720P resolution stereo camera is 14.75 MB. Considering the edge server scenario, the uplink bandwidth between the vehicle and the LTE base station achieves on average 25 Mbps[4], so that the transmission of the point cloud finishes in 4.72 sec. On the other hand, the transmission between vehicles and cloud servers is obviously slower, as the data needs to traverse through the Internet. Taking Google Cloud Platform as an example, the average uplink bandwidth is 4.4 Mbps[5], so that the transmission of the point cloud could take up to 26.82 sec.

Our preliminary evaluation shows that ARVE would decrease transmission latency noticeably.

7 CONCLUSION AND FUTURE WORK

In this paper, we have presented ARVE, an architectural framework for vehicle-to-edge applications. Our system could serve applications with different requirements in vehicular communication system. We choose CVV as the representative use case and proposed corresponding solution in details. With our preliminary evaluation using real data from London, we have shown that ARVE could improve vehicular network significantly with only reasonable requirements on the number of installed edge servers. In our future work, we will solve the specific challenges in CVV especially regarding ARHUD. The complex computational process involves several steps and may require orchestration of edge and vehicle resource, to improve utilization and computation efficiency while decreasing latency. We also plan to implement parts of the solution using real hardware and networking devices.

ACKNOWLEDGEMENTS

This research was funded by the joint EU FP7 Marie Curie Actions Cleansky Project, Contract No. 607584. This research has been supported, in part, by projects 26211515 and 16214817 from the Research Grants Council of Hong Kong.

REFERENCES

[1] Gerardo Daalderop Paul D. Alexander Franz Schober Alessio Filippi, Kees Moerman and Werner Pfliegl. 2016. Ready to roll: Why 802.11p beats LTE and 5G for V2x. (2016).
[2] Onur Ascigil, Truong Khoa Phan, Argyrios G Tasiopoulos, Vasilis Sourlas, Ioannis Psaras, and George Pavlou. 2017. On Uncoordinated Service Placement in Edge-Clouds. In Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on. IEEE, 41–48.

[4] https://www.4g.co.uk/how-fast-is-4g/
[5] https://testmy.net/hoststats/google_cloud

[3] CHINADAILY. 2018. Shanghai allows autonomous tests. (2018). http://www.chinadaily.com.cn/business/motoring/2017-11/13/content_34469664.htm
[4] Mobile-Edge Computing. 2014. European Telecomm. Standards Inst.(ETSI). (2014).
[5] Stephane Daeuble. 2017. No technical challenge too big for small cells. (2017). https://www.nokia.com/en_int/blog/no-technical-challenge-big-small-cells
[6] Ming Ding, David Lopez-Perez, Holger Claussen, and Mohamed Ali Kaafar. 2017. On the Fundamental Characteristics of Ultra-Dense Small Cell Networks. arXiv preprint arXiv:1710.05297 (2017).
[7] Sebastian Engel, Claudia Kratzsch, and Klaus David. 2013. Car2pedestrian-communication: Protection of vulnerable road users using smartphones. In Advanced Microsystems for Automotive Applications 2013. Springer, 31–41.
[8] Miad Faezipour, Mehrdad Nourani, Adnan Saeed, and Sateesh Addepalli. 2012. Progress and challenges in intelligent vehicle area networks. *Commun. ACM* 55, 2 (2012), 90–100.
[9] Small Cell Forum. 2017. Small Cell Forum Unveils Operator Research Showing Accelerating Densification and Enterprise Deployments on Road to 5G. (2017). https://www.smallcellforum.org/press-releases/small-cell-forum-unveils-operator-research-showing-accelerating-densification-enterprise-deployments-road-5g/
[10] Mario Gerla, Eun-Kyu Lee, Giovanni Pau, and Uichin Lee. 2014. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In Internet of Things (WF-IoT), 2014 IEEE World Forum on. IEEE, 241–246.
[11] Dennis Grewe, Marco Wagner, Mayutan Arumaithurai, Ioannis Psaras, and Dirk Kutscher. 2017. Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions. In Proceedings of the Workshop on Mobile Edge Communications. ACM, 7–12.
[12] Xumin Huang, Rong Yu, Jiawen Kang, Yejun He, and Yan Zhang. 2017. Exploring mobile edge computing for 5G-enabled software defined vehicular networks. IEEE Wireless Communications 24, 6 (2017), 55–63.
[13] The Information. 2018. Inside CruiseâĂŹs Bumpy Ride: The Limits of Self-Driving Cars. (2018). https://www.theinformation.com/go/c5d3e2ec78
[14] Hyungil Kim, Xuefang Wu, Joseph L Gabbard, and Nicholas F Polys. 2013. Exploring head-up augmented reality interfaces for crash warning systems. In Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. ACM, 224–227.
[15] MOVIMENTO. 2018. Connected Car Battle Lines Are Drawn Between 5G and DSRC. (2018). https://movimentogroup.com/media-coverage/connected-car-battle-lines-drawn-5g-dsrc/
[16] Amy Nordrum. 2016. Autonomous driving experts weigh 5G cellular network against dedicated short range communications. IEEE Spectrum, Cars That Think (2016).
[17] Hyesun Park and Kyong-ho Kim. 2013. Efficient information representation method for driver-centered AR-HUD system. In International Conference of Design, User Experience, and Usability. Springer, 393–400.
[18] Hye Sun Park, Min Woo Park, Kwang Hee Won, Kyong-Ho Kim, and Soon Ki Jung. 2013. In-Vehicle AR-HUD System to Provide Driving-Safety Information. ETRI journal 35, 6 (2013), 1038–1047.
[19] Hang Qiu, Fawad Ahmad, Ramesh Govindan, Marco Gruteser, Fan Bai, and Gorkem Kar. 2017. Augmented Vehicular Reality: Enabling Extended Vision for Future Vehicles. In Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications. ACM, 67–72.
[20] Santokh Singh. 2015. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical Report.
[21] THEVERGE. 2018. California green lights fully driverless cars for testing on public roads. (2018). https://www.theverge.com/2018/2/26/17054000/self-driving-car-california-dmv-regulations
[22] WAYRAY. 2018. The first holographic AR navigation system for cars. (2018). https://wayray.com/navion