SAT SUBMISSION - II

Team No.: 09 **Date of submission: 28/ 10/ 2022**

Name of the student: BHARATHI SHRINIVASAN T R **ID No.: 2022H1400182P**

Paper Title with citation:

Arata Koike, Yoshiko Sueda (Meisei University – Tokyo, Japan) "**Contents Delivery for Autonomous Driving Cars in Conjunction with Car Navigation System**" – In proceedings of IEICE – The 20th Asia-Pacific Network Operations and Management Symposium (APNOMS) 2019

Motivation for selection of the paper:

In further to the topic chosen (Vehicular automation), the most trending area is autonomous driving vehicle and its supporting infrastructure. We wanted to dive into the infrastructure/architecture that support all-robot autonomous vehicle in road. Expecting to learn the current trend in proposal for architecture, communication systems, cellular network support for mobile vehicle information exchange. Autonomous driving is not future, but a reality now (i.e. *Tesla model S 2022, CMU Navlab 11 2001*) but the challenge being the transition from partial-human to full robot driving. The supporting infrastructure and on-going research in this area is very significant as it helps to pave the future of complete all-robot autonomous driving.

Summary:

Idea of autonomous driving system is been there since the inception of early radio-controlled (semi-autonomous) driving system. Since then, there has been a lot of improvement in this area, mainly due to the possibility of computers used in automobiles to compute/control huge volume of data, especially the image processing, and machine learning based adaptive algorithms implementation. This paper focuses on the future network architecture to support autonomous driving cars. Cars are mobile, and any data exchange from a network-system such as cloud, has various challenges i.e., handover. Most information such network would exchange be navigation system, traffic details, offloading some service to cloud, on-flight entertainment etc. The paper only addresses architecture proposal using wide area network, not peer-to-peer i.e. V2V (vehicle-to-vehicle communication to exchange some useful information between two or more vehicles in close proximity, the application of such would be like collision prevention, nearby lane availability etc), V2I (vehicle-to-Infrastructure communication, where roadside nodes/infra would exchange information with vehicle like speed limit information that helps the vehicle to set throttle threshold etc). The networked architecture to support moving vehicles to exchange information that could be a broadcast, request-respond type. The author proposed a unique methodology to address the handover issue in a mobile access, the essence being that the huge data is chunked into smaller piece of info, and the moving vehicle could pick such pieces along the path using nearby network resources. The infrastructure would include a cellular network of local caches at Wi-Fi access point (**MobilityFirst scheme*) at different locations alongside road, and some orchestrator server to plan the data chunking and placing it at appropriate local proxy/cache. This planning is more of deterministic, as the car navigation information would have the exact pathway the vehicle would cover at known time stamp, so the orchestrator server can chunk the data and distribute to all the local proxies at different timing just to fit the constrain that proxy hold data when the vehicle enters into its cellular zone. This way the proxies can place the chuck data before the vehicle leave its proximity area and there by higher probability to not reach handover point before completion. This paper elaborates a detailed scheme and evaluation of four different scenarios/schemes that would be possible infrastructure. The scheme proposes to have wireless base station (*eNode B*) with mini local proxy/cache placed at different intervals. Then the author evaluated the model for TCP throughput and elapsed time with various moving speed using NS-3 simulator. The shows that the scheme he proposed in paper has better throughput and nominal network storage resource.

My conclusion- This paper introduced us to the architectural challenges in building an autonomous vehicle communication network. The main challenge being the mobility, and handover issue due to it. The author proposed a novel way to address this challenge and has shown the results of its throughput using a simulator. We learned little about the simulation environment, little about the communication protocol (i.e TCP) used in major network infrastructure.

Plan of selection of paper for next week:

We are planning to read peer-to-peer based communication system in vehicular system i.e V2V and V2I. As these are some of the latest features in most autonomous cars, and scope not limited to i.e collision avoiding, lane changeover, traffic management system. Learn about the underlaying foundation in the architecture of such system, dwell into the communication aspect, and also cybersecurity related challenges with autonomous systems.

Picked up papers-

Biswas, Subir, Raymond Tatchikou, and Francois Dion."**Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety**" Communications Magazine, IEEE 44, no. 1 (2006): 74-82.

Mudalige, Priyantha. "**Connected autonomous driving: Electric networked vehicle (EN-V) technology."** In 18th ITS World Congress. 2011.

P. Mac , Z. Becvar, "**Mobile Edge Computing: A Survey on Architecture and Computation Offloading**," IEEE Communication Survey&Tutorial, Vol. 19 Issue 3, pp. 1628-1656, March 2017.

***MobilityFirst Project -** http://mobilityfirst.winlab.rutgers.edu/.

Contents Delivery for Autonomous Driving Cars in Conjunction with Car Navigation System

Arata Koike
Faculty of Humanities
Tokyo Kasei University
Tokyo, Japan
koike@ieee.org

Yoshiko Sueda
School of Information Science
Meisei University
Tokyo Japan
yoshiko.sueda@meisei-u.ac.jp

***Abstract*—We take a green field approach for architecting a network infrastructure to support autonomous driving cars. We present a possible use case scenario for it from the view point of wider information system in our society. Networks are one of the parts of our future infrastructure forming our society. In our scenario, we integrate decisions by car navigation systems and measurable information from a network. This makes it possible for us to enable predicting the future location of a car more precisely. Using this feature, and if we apply it for contents delivery using distributed caches on road sides, we can prefetch contents for a car to a next adjacent local cache server beforehand. We propose and compare several schemes to evaluate effectiveness of downloading a large content for a moving car.**

Keywords*—*autonomous driving car, future infrastructure, contents delivery network, performance, handover, content prefetching

I. INTRODUCTION

Autonomous driving technology will change our transportation infrastructures. Our interest is what kind of communication infrastructure would be suitable for new infrastructures in our future society. Existing road infrastructures supports cars operated by human. That means it provides required functions for driving cars by human. In such infrastructure environment, autonomous driving car must detect, read and recognize signals for human. Many people tackled these tasks using machine learning technologies and we have been seeing a big progress these days. On the other hand, such direction still suffers unsolved issues. The authors of [1] pointed out one such threat by malicious attack to create a confusion for signal detection algorithm. The origin of this problem is that we use our existing infrastructure for autonomous driving cars.

One specific feature of autonomous driving is that people only need to input their destination and preferences for route choices at the starting location. The autonomous driving cars inquire directions to a cloud-based car navigation system. The navigation system calculates a most appropriate route to the destination based on the inputs and other available information by the system on a cloud. For example, if the preference is to minimize the time to the destination, the system will select a route using toll-road. After finding such route, the navigation system notifies the car the result. Then the car moves without any interaction or intervention of a human. Cars, instead, will interact frequently with the navigation system to exchange information and to use functions on clouds. Cars will update their driving processes based on the information from the cloud and that sensed by themselves. This means autonomous driving cars need to have a capability to upload and download information to a cloud. If we think remotely controlled robotics device such as a robot arm, the robotics device continuously needs to send video image to a remotely located control server to notify its current situation so we need a larger bandwidth capacity for uploading direction [2][3]. For autonomous driving car, we may assume less traffic for upload link than download link since a car has actual controller equipped on itself so we can assume the majority of uploading traffic by low frequency short packet for inquiring messages. When a car is moving, it can communicate with other nearby vehicles using direct vehicle-to-vehicle (V2V) communication. V2V communications are important to avoid collisions and to obtain immediate responses but we exclude it from our consideration for communication infrastructure since they do not need to use a network.

In this paper, we do not make deep diving into details for investigating the types of actual contents of information of systems. We, rather, simply assume that cars continuously need to interact with network functions or cloud resources. That means frequencies of interactions are high. They need more network capacity for downloading than uploading. We, therefore, consider how and where we provide network resources and how we manage relationships between information and network resources servicing the above type of information flows.

We organize this paper in the following way. In section II, we briefly overview related works for content prefetching technologies for mobile users. In section III, we explain our use case and clarify what problems we want to solve. We propose our scenario in section IV and also explain our evaluation models for simulation. We show our evaluation results and discussion on section V. We conclude our paper in section IV.

Part of this work was supported by JSPS KAKENHI Grant Number JP 18K11279.

II. RELATED WORKS

There have been many studies on contents retrieval while a car is moving. The authors of [4] studied prediction-based contents prefetch scheme using local caches at WiFi access point for mobile users under MobilityFirst scheme [5][6], which makes an association between mobile user and upper layer application. They predict probabilities of user mobility by information available from the network. Then, they prepare contents to the most appropriate cache server indicated by the value of the probability and showed improvement of cache hit ratio. The authors of [7] extensively investigate probability of mobility by Markovian model and proposes Entropy based prediction model to prefetch contents.

There is also some proposal of direct utilization of lower layer information for upper layer. For example, 3GPP specifies push mechanism to notify the attachment of a mobile user to their network [8][9].

In [10][11], we considered how we could maintain TCP throughput at the time of handover for a cellular system. Our approach was not network centric approach but application centric one. If we took network centric one, in order to maintain a good TCP throughput, a mobile user needs to move along with the better path based on the radio signals. That means the mobile user might not be able to move straight ahead to his/her destination. The car selects paths based on the strength of radio signals, so that the routes to the destination would be a kind of zigzag path. Our approach, instead, was that a cellular system prepares radio resources by predicting the direction of a mobile user. This can be possible for autonomous driving cars since a route navigation system determines his/her route so that the cellular system can know the best appropriate base station beforehand. The results showed that if we can utilize the information not available inside a network, we could optimize network resources. We can see similar service centric approaches for managing network resources against simultaneous call arrival due to online event ticket sales or mass voting, such as American Idle. If we can know such event information in service side beforehand, a network operator can prepare their network resources for such mass calling. This kind of collaboration between service side and network side is essential to provide a better quality of experiences (QoE) for users as well as maintaining good Quality of Service (QoS) by mitigating traffic congestions for network operators.

III. USECASE AND PROBLEM STATEMENT

Let us clarify what kind of information exchange we will have for autonomous driving cars. One category is that cars receives broadcasting information from a cloud. All cars in the same area can receive the same broadcasting information. The other category is request-response type. Cars request certain information from or offload certain jobs to clouds and receives the results from the clouds. The receiving information is obviously different depending on the locations and requests. There are many discussions on Vehicle-to-Vehicle (V2V) communication for the sake of avoiding collisions. We also see Vehicle-to-Infrastructure (V2I) communication; a car communicates with road side equipment or Road Side Unite (RSU) to receive some signal, which may directly act upon a car's driving control. V2V or V2I communication will be effective to minimize the communication delay to receive information. As we mentioned at section I, we exclude these categories (i.e., V2V and V2I) from our focus in this paper as they do not utilize network other than local segment. We focus on communications using wider area of network. Our target is content distribution for broadcasting or request-response categories.

Now, we figure out the possible use cases for the above two categories (i.e., broadcasting and request-response). It might be helpful for assisting autonomous driving by downloading data for nearby area information, for example, 3-dimentional views of nearby intersections and precise navigation information from a car navigation system on a cloud. This might not give us an impression of mass amount of data now. We, however, expect that this impression will be untrue for future as we increase precision of information year by year. If we think that our autonomous driving car moves while continuously obtaining precise information, this rather gives us an impression that we need to handle huge amount of data transactions. For example, if we want to know the status of an intersection or roads where we will arrive there in a minute, we also need to obtain information of various other cars that arrives at the intersection at the same time. Even if we assume 1Mbyte for each cars, if there are 100 cars, it means 100Mbytes. We need such information for every intersection that we will enter. This might be the comparative amount of data with the high-precision video streaming.

Other usage cases include video downloads as entertainments for passengers on the car since none of the passenger need to concentrate on operating the car. This is the same usage case as in-flight entertainments. We call those categories as delivery of large amount of data.

The problem associated with mobility is handover. Handover creates delay or instantaneous blackout and that affect performances. Communication protocol using feedback loop control, such as TCP, suffers this problem so much. Thus, reducing number of handovers is one requirement. One solution to mitigate this effect is dividing large contents into small chunks of data. One advantage of the usage of chunk is that we can reduce the transmission time so that we have higher probability to transmit it before reaching a handover point. This of course means it decreases the performance if we cannot complete the chunk transmission within the coverage area due to additional overhead. Since the deployment of micro cell is a recent trend in cellular system to increase its radio capacity, the usage of chunk is a possible direction for content distribution.

Using chunk creates another benefit. We can spread each chunk over different server as content caches. Copying a whole content requires a certain storage space. It is waste of resources if we place the same whole content in many cache servers. If we use chunks, it might not be a problem to put the same chunk in multiple of servers since each chunk has limited size. This approach creates a problem on how to distribute each chunk in network wide efficiently. This is the problem we want to solve in this paper. Our approach is to combine prediction-based chunk prefetching at upper layer and prediction-based handover in lower layer to optimize the overall system performance for autonomous driving car infrastructure. We also note that our approach has intimacy with the recent progresses on Mobile Edge Cloud (MEC) [12] development.

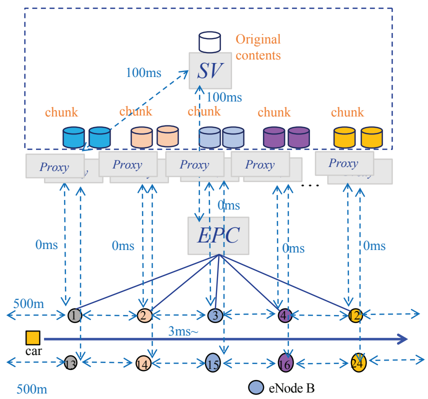


Fig. 1. Transition property for transffered data with diffent speeds.

IV. Proposed Scheme and Evaluation Scenario

A. Proposed scheme

We assume a network configuration where each wireless base station (eNode B) has a mini local proxy/cache server. Using information provided by car navigation system on a cloud, we preload contents to appropriate local proxy/cache servers beforehand. This is our main idea. As we mentioned in the previous sections, this is not a probabilistic approach but deterministic one since we can predict the location of a car using car navigation information. We then evaluate our model for TCP throughput and elapsed time with various moving speed using simulation. In order to focus on TCP feedback loop, we made the following four scenarios for simulation. We use NS-3 simulator [13] for our simulation.

B. Scenario 1

Each eNode B has proxy/cache server. Two consecutive chunks are preloaded to each proxy/cache serer. A sample behavior is as follows; if a car can download chunk #1 and #2 during the car is in coverage area of the first eNode B, then the car further downloads the chunk #3 from the original server since chunk #3 is not pre-fetched. If the car does not finish downloading the chunk #3 during its stay at the first eNode B and make an handover to the second eNode B, the car receives the rest of chunk #3 by X2' transfer and terminate the TCP session. Then the UE reinitiate TCP session to the next proxy/cache server to get chunk #4.

C. Scenario 2

For this scenario, we do not divide a content to chunks. Instead, we preload the whole content to each proxy/cache server. Thus, total amount of cached data is huge. The behavior is almost the same as the scenario 1 so there is a higher risk to rely on X2' transfer when the car handed over.

D. Scenario 3

Difference of this scenario from Scenario 1 is that it starts prefetching chunks to nearby proxy/cache server after a car arrived within a coverage area of an eNode B. Thus it always need to download first chunk from the original server but prefetching the next chunk to the local proxy/cache might be completed before completing the transmission of the first chunk.

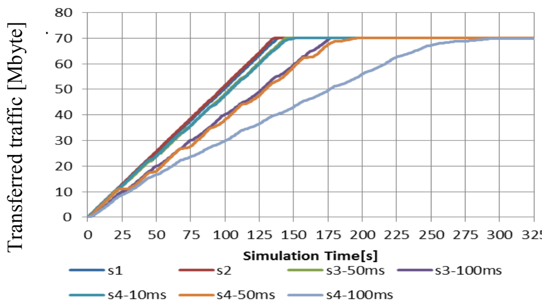
E. Scenario 4

This scenario does not prefetch a content nor chunk from the original server. The car always needs to download a content from the original server. For this scenario we varies link delay between original server and eNode B with 10, 50, and 100ms.

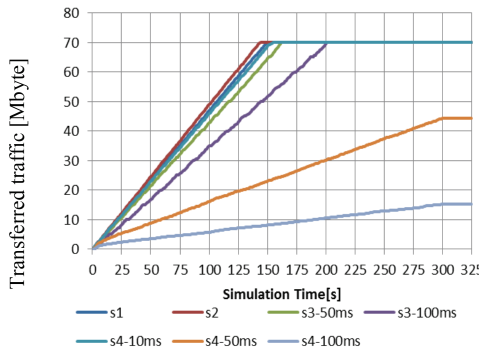
V. Evaluation Result and Discussion

We show our evaluations for the above four scenarios. We change the speed of a car and compare time transition properties of transferred amount of contents for our scenarios. For this evaluation, we set distance between adjacent eNode Bs to 50m to model micro cell environment in urban populated areas. We changed the velocity of a car from 8km/h to 72km/h. Fig. 1 shows our network model. The sequence shows that Chunk #1 and #2 are prefetched to Proxy1 and the car in focus retrieves these chunks in the order. When the car competed the reception of the two chunks before leaving a coverage area of eNB1, it starts retrieving Chunk #3 from the source contents server as Chunk # 3 is not prefetched to proxy1.Then, before the completion of Chunk #3 retrieval, the car makes a hand over to eNB2. Then, the rest of Chunk #3 data is sent using X2 transfer to eNB2 from eNB1. Since Chunk #4 is prefetched to proxy 2 at eNB2, it starts retrieving Chunk #4 soon after it finished Chunk #3 retrieval. This corresponds to the Scenario 1 in Section IV B. For Scenario 2, each proxy has preloaded whole contents but it requires X2 transfer if a car make a handover during the retrieval. For Scenario 3, a car must download Chunk #1 from the original server but other chunks are prefetched to proxies. For Scenario 4, we always need to retrieve contents from the original server so we use the sequence for Chunk #3 by reading Chunk #3 as a whole content.

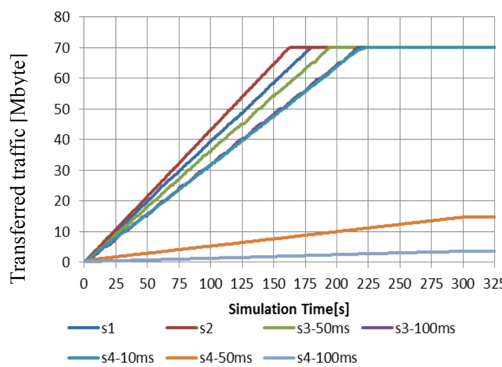
We put series of eNode Bs in line at each grid point (Fig. 1). Distance between two eNode Bs are 50m. We set the number of Radio Bands (RBs) as 6 and each RB has 4Mbit/s bandwidth. In this physical configuration, a car moves at along the eNode Bs with a constant speed at 72km/h from left to right. We set the chunk size as 5 Mbyte each. We tried three different link delays, 10ms, 50ms, and 100ms, between eNode B and a server. We assumed 12Kbyte TCP buffer and 100Kbyte Radio Control Link (RCL) buffer. Our TCP maximum segment size is 1,000 bytes. Figs. 2(a) to (c) show the results. In those figures, s1 means scenario 1, s2 is scenario 2, s3 is scenario 3, and s4 is scenario 4. Numeric after '-' indicates link delay. For example, s3-50ms means scenario 3 with its link delay between the original content server and proxy/cache servers or Evolved Packet Core (EPC) node is 50ms. We only use content prefetching mechanism here. We can find that the link delays affect the overall performance in any velocity. Especially, for scenario 4, which does not use prefetching at all, affect so much. The more speed increase, the more throughput performance degrades for all cases but scenario 2 yields the best results. This is because the scenario 2 caches whole contents at each proxy/cache server beforehand with the sacrifice of the consumption of local storage. It copies the whole contents to all local storages so that the amount proportionally increases with the number of proxy/caches. The second-best choice is scenario 1 regarding to the throughput. Actually, the scenario 1 only requires local storage for two chunks and thus it has scalable nature for serving large areas. By comparing those 4 scenarios, we can say the scenario 1 is the best choice to satisfy both throughput and storage capacity requirements.



(a) speed=8km/h



(b) speed=32km/h



(c) speed=72km/h

Fig. 2. Transition property for transffered data with diffent speeds.

VI. Conclusion

We focus on a future communication infrastructure to support autonomous driving cars. If we consider it based on the existing environment, there are many things that we need to tackle with to realize autonomous driving. In this paper, we rather started from an ideal situation where no human operated cars allowed on our infrastructure. Then, we pointed out that we can obtain location information of a car in a priori for such environment. Once we can precisely predict the location and direction of cars, we can efficiently implement content prefetching scheme to improve the total system efficiency.

Of course, it is ideal if we can determine everything at the time of inputting a destination. Unfortunately, even for our ideal environment for autonomous driving car, road traffic situation changes time to time due to emergence of new events. If a car navigation system on a cloud can grasp the ever-changing information, we can assign a role of a car traffic control center the car navigation system as a role of a car traffic control center. This means the information available from the control center always has good precision of car locations and directions. By taking green field approach, we therefore can argue that content prefetching by location prediction is one possible solution for our future infrastructure. We also relate prediction-based handover at lower layer and content prefetching at upper layer in our proposed scenarios. Our evaluation shows that the above our approach is effective in terms of reducing contents download time and increase throughput to obtain contents.

References

[1] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.1625-1634, Salt Lake City, Utah, U.S.A., June 2018.

[2] N. Higo, Y. Sueda, and A. Koike, “COG: Overlay network functions assisting COnnected next Generation society systems,” Proc of the 23rd IEEE International Symposium on Local and Metropolitan Area Networks, (IEEE LANMAN 2017), Osaka Japan, June 2017.

[3] A. Koike, N. Higo, and Y. Sueda, “Proxy-based Network Function to Assist Robotic Feedback Control System,” Proc of the 24th IEEE International Symposium on Local and Metropolitan Area Networks, (IEEE LANMAN 2018), pp. 116-118, Washington, DC, USA, June 25-27, 2018.

[4] F. Zhang, C. Xu, Y. Zhang, K. K. Ramakrishnan, S. Mukherjee, R Yates , T. Nguyen, “EdgeBuffer: Caching and prefetching content at the edge in the MobilityFirst future Internet architecture,” Proc. of the IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, 14-17 June 2015.

[5] MobilityFirst Project, http://mobilityfirst.winlab.rutgers.edu/.

[6] D. Raychaudhuri, K. Nagaraja and A. Venkataramani, "MobilityFirst: A Robust and Trustworthy MobilityCentric Architecture for the Future Internet", ACM SIGMobile Mobile Computing and Communication Review (MC2R), Volume 16 Issue 4, October 2012.

[7] N. Abani, T. Braun, and M. Gerla, “Proactive Caching with Mobility Prediction under Uncertainty in Information-centric Networks,” Proc. of ICN ’17, Berlin, Germany, pp. 88-97, September 26–28, 2017.

[8] 3GPP TS 23.682 V14.2.0.

[9] 3GPP TS 23.401 V14.2.0.

[10] Y. Sueda, and A. Koike, “Service centric mobility management for improving Quality of Experience toward future mobile network,” Proc. of the 22nd IEEE International Symposium on Local and Metropolitan Area Networks, (IEEE LANMAN2016), Rome, Italy, June 2016.

[11] A. Koike, and Y. Sueda, “Changes in Key Performance Indicators by Phase Transition in Internet of Things,” Proc. of the 15th IEEE/ACIS International Conference on Computer and Information Science (IEEE/ACIS ICIS2016), pp.1001-1006, Okayama, Japan, July 2016.

[12] P. Mac , Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," IEEE Communication Survey&Tutorial, Vol. 19 Issue 3, pp. 1628-1656, March 2017.

[13] The ns-3 project, https://www/nsnam.org/.