Delay Minimization for Infotainment Services in Smart Vehicle Network

Tri Nguyen Dang, Choong Seon Hong
Department of Computer Science and Engineering, Kyung Hee University, 446-701,Korea
Email: {trind, cshong}@khu.ac.kr

Abstract

Infotainment services will become a basic necessity for future automobiles especially to enable autonomous driving and providing entertainment to its passengers. However, getting infotainment contents from a centralized data center will incur high delays and can hinder the performance in terms of required quality. Installation of Multi-access Edge Computing (MEC) servers as road side unit (RSU) can be considered as an alternative promising solution that can reduce the access delays and achieve the required performance by offering computation and caching services right next to smarts cars. However, MECs have limited caching and computation capacities. In this work, we aim to minimize the access delays for these infotainment services subject to the limited communication, caching and computation resources.

1 Introduction

The rise of autonomous car have revolutionized automobile industries. Recently, great interest is shown both by the academic and automobile industry to realize a network of smart cars due to massive advances in the sensing, computation and communication technologies [1], [2]. The goal of a network not only provide basic service: traffic condition, weather, road safety, etc., also provide an infotainment service. The infotainment service can be considered like: video, games, social media, augmented reality, etc,. However, delaying is one of the most challenges in this field. Basically, this service requires getting content from RSU to smart cars, or DC to RSU. Therefore, to address this problem, we consider a 3C model(Communication, Caching, Computation) show in Fig. 1. In which, the RSU decides to cache the content, upgrade from local quality to meet the requested format of smart cars, or download directly from DC.

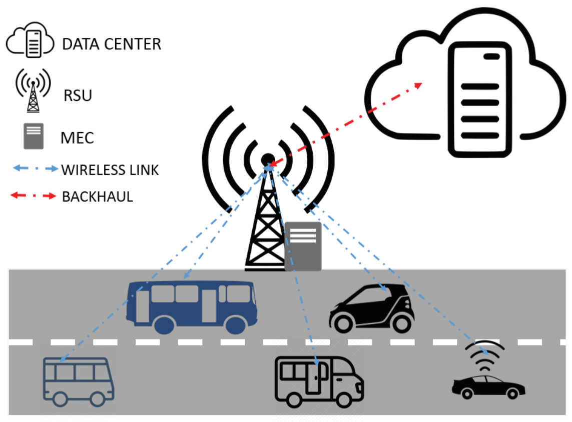

Fig. 1. Illustration of our system model.

2 System model and Problem Formulation

Let $\mathcal{N} = \{1,\dots,N\}$ be the set of N smart cars in the coverage range of RSU, and a set of M contents $\mathcal{M} = \{1,2,\dots,M\}$, where each content has K different formats based on the quality. We assume $K = \{1,2,3,4\}$ which represents four different formats of a content, i.e., $240p, 360p, 720p, 1040p$, and $s^{(m,k)}$ denoted the size of content in megabits. In the other hand, we assume RSU enabled MEC has both caching and computational capabilities. Let S denoted the cache capacity, and P denote the computational capacity of the MEC, respectively.

2.1 Communication Model

Communication is required in order to download the contents in both case: from Data Center(DC) to the RSU enabled MEC, and from RSU to the smart cars. Firstly, in the case of contents are downloaded from DC to RSU. Contents are downloaded from DC via a fiber backhaul link ω_0. Therefore, the transmission delay between RSU and DC can be calculated as follow:

$$\tau_0^{(m,k)} = \frac{s^{(m,k)}}{\omega_0}, \tag{1}$$

Secondly, the delay to download a content from RSU by smart car define by

$$\tau_i^{(m,k)} = \frac{s^{(m,k)}}{\omega_i}. \tag{2}$$

where ω_i is the data rate for a wireless link between a smart car $i \in \mathcal{N}$ and RSU

$$\omega_i = \zeta_i B \log_2\left(1 + \varphi |G_i|^2\right), \tag{3}$$

and ζ_i is the authorized bandwidth, B is the fraction of the RSU owned bandwidth , G_i is the channel gain between RSU and smart car i, and φ is the transmission power of RSU. On the other hand, t_i is the time for a smart car i to exit the coverage radius γ of an RSU j as follows:

$$t_i = \frac{2\gamma}{\mu_i}, \tag{4}$$

where μ_i represents the velocity of a smart car i.

2.2 Caching Model

The objective of caching contents is minimize the access delay and reduce the backhaul bandwidth as much as possible. In this model, we assume that RSU has limited cache capacity C. Therefore, the RSU need to cache a content with have high probability requested by smart cars. Let p^m, q^k represent the probabilities of a request content m in representation k, and distribution of $p,$ and q are independent with each other. Therefore, the request of content (m,k) is formulated as $p^{(m,k)} = p^m q^k$. Then, the RSU need to follow the caching contrains which can be expressed as:

$$\sum_{m=1}^{M}\sum_{k=1}^{K} x^{(m,k)} s^{(m,k)} \leq C, \tag{5}$$

where $x^{(m,k)}$ is the caching decision indicator variable which defined as follows:

$$x^{(m,k)} = \begin{cases} 1, & \text{if RSU caches content } (m,k), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Furthermore, we assume that the base layer of all contents are available at the RSUs, i.e., a base layer of content m is represented as $(m,0)$.

2.3 Computation Model

In this subsection, we will discussed the computation model for the cached contents. In this case, we assume that, the content m is cached in different format with the requested format by smart car. Therefore, the RSU need to tranfrom the quality of current available content to the specify requested content. This method can be done by: Transcoding[3], Super-Resolution[4] [5], Low-Resolution[6].

In our model, converting a cached content (m,k') to content (m,k) will consumed an amount of resources $h^{(m,k)}$ of MEC enabled RSUs, where computational resource allocation $h^{(m,k)}$ is defined as

$$h^{(m,k)} = \pi^{(m,k)} s^{(m,k)}, \tag{7}$$

where $\pi_j^{(m,k)}$ number CPU cycles per bit required for converting content, $y^{(m,k)}$ represents the computation decision variable, which is expressed as:

$$y^{(m,k)} = \begin{cases} 1, & \text{if (m,k) is converted,} \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

Therefore, the computation capacity contrains can be expressed as

$$\sum_{m=1}^{M}\sum_{k=1}^{K} h^{(m,k)} y^{(m,k)} \leq P, \tag{9}$$

Then, the computational delay to convert cached content $(m,0)$ into desired content (m,k) at an RSU j is given by

$$l^{(m,k)} = \frac{(s^{(m,k)})^2}{h^{(m,k)}} \tag{10}$$

Note that, if the RSU does not have enough resource to cache, or process the content, it forwards the request to the DC.

2.4 Problem formulation

The delay of smart i downloading a content (m,k) is define as follow:

$$T_i^{(m,k)} = x^{(m,k)}\tau_i^{(m,k)} + \left(1 - x^{(m,k)}\right) \\ \left[y^{(m,k)}\left(\tau_i^{(m,k)} + l^{m,k}\right) + \left(1 - y^{m,k}\right)\left(\tau_i^{(m,k)} + \tau_0^{(m,k)}\right)\right]. \tag{11}$$

Then, the total delay of a given smart car i with the set of contents M define by

$$\Phi_i = \sum_{m=1}^{M}\sum_{k=1}^{K} p^{(m,k)} T_i^{(m,k)} \tag{12}$$

Due to the binary variables x, and y the optimization problem is a large-scale multi-dimensional Knapsack problem which falls in the NP-hard category. We then proposal a relaxation methods aims to reduce the complexity, and yield a novel idea is that: x fraction of content is cache, the other y fraction is upgrade from local quality, and the remaining fraction of content can be downloaded from DC. We introduce a new variable z represent for the fraction of content can be dowloaded from DC. Threfore, the delay of a given smart car i when download a content (m,k) define by

$$T_i^{(m,k)} = x^{(m,k)}\tau_i^{(m,k)} + y^{(m,k)}\left(\tau_i^{(m,k)} + l^{(m,k)}\right) \\ + z^{(m,k)}\left(\tau_i^{(m,k)} + \tau_0^{(m,k)}\right), \tag{13}$$

And the relax problem as follows:

$$\min_{x,y,z} \sum_{i=1}^{N} \Phi_i \tag{14a}$$

$$\text{s.t.} T_i^{(m,k)} \leq t_i, \forall i \in N, \tag{14b}$$

$$\sum_{m=1}^{M}\sum_{k=1}^{K} x^{(m,k)} s^{(m,k)} \leq C \tag{14c}$$

$$\sum_{m=1}^{M}\sum_{k=1}^{K} h^{(m,k)} y^{(m,k)} \leq P, \tag{14d}$$

$$x^{(m,k)} + y^{(m,k)} + z^{(m,k)} = 1, \forall m \in M, \forall k \in K \tag{14e}$$

$$0 \leq x^{(m,k)}, y^{(m,k)}, z^{(m,k)} \leq 1, \forall m \in M, \forall k \in K, \tag{14f}$$

The objective function is convex, the constraints are an either linear, affine or closed convex set. Therefore, our problem becomes a convex optimization problem. We then use the Convex.jl [7] to solve the problem and numerical analysis by Julia language. Due to the limit of space, we could not present the detail solutions here.

3 Simulation

In our model, We use the Julia language 0.6.2 as our simulation tool to evaluate our problem. We numerically evaluate the results on a single computer with the following specifications: Intel(R) Core(TM) i5-4690 CPU 3.50 (GHz), RAM 28.0(GB), GPU GTX 1060 3(GB). For the simulation setup, number of smart cars $N = 50$, $|\mathcal{M}| = 50, K = 4$ contents, cache capacity ranging from $C = \{10 \sim 40\}$, computational capacity ranging from $P = \{300 \sim 1000\}$, $\omega_0 = 50(Mbps)$, $\omega_i = \{1 \sim 5\}(Mbps)$.

We observe that the proposed approach outperforms all other approaches: Greedy, and Random in in Fig. 2. Moreover, we can infer that the average total delay decreases proportionally with the cache size.

4 Conclusion

In this paper, we proposed the joint communication, caching, and computation model for infotainment for smart vehicle network. Our numerical results shown that the proposed approach has a reasonably good performance compared to the Greedy, and Random algorithm.

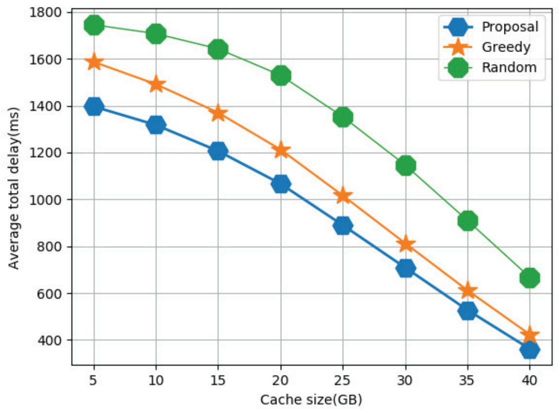

Fig. 2. Total delay vs. cache size.

Acknowledgments

This work was supported by Institute for Information communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2015-0-00557,Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices) *Dr. CS Hong is the corresponding author

References

[1] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, "Self-driving cars," *Computer*, vol. 50, no. 12, pp. 18–23, 2017.

[2] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview," *ieee vehicular technology magazine*, vol. 14, no. 1, pp. 62–70, 2019.

[3] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges," *arXiv preprint arXiv:1612.03184*, 2016.

[4] K. Hayat, "Super-resolution via deep learning," *arXiv preprint arXiv:1706.09077*, 2017.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[6] M. Chevalier, N. Thome, M. Cord, J. Fournier, G. Henaff, and E. Dusch, "Low resolution convolutional neural network for automatic target recognition," in *7th International Symposium on Optronics in Defence and Security*, 2016.

[7] J. Bezanzon, S. Karpinski, V. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," in *Lang.NEXT*, Apr. 2012. [Online]. Available: http://julialang.org/images/lang.next.pdf