Service Chaining Offloading Decision in the EdgeAI: A Deep Reinforcement Learning Approach

1st Minkyung Lee
dept. of Computer Science and Engineering
Kyung Hee University
Yongin, Republic of Korea
minkyung0110@khu.ac.kr

* Choong Seon Hong
dept. of Computer Science and Engineering
Kyung Hee University
Yongin, Republic of Korea
cshong@khu.ac.kr

***Abstract*—Many mission critical devices are increasing with upcoming 5G network to fulfill a low latency for a real time network service on smart factory, autonomous vehicle, etc. Distributed cloud computing system also has a key role to execute the various mobile devices, because, an edge computing is the nearest from the mobile devices to provide low latency and computation energy consumption. In this paper, we consider the autonomous vehicles with video live streaming services. Especially, the vehicles require a low transmission delay as within 10 ms. To reduce a latency with low energy consumption, we propose a service chaining offloading decision with a deep reinforcement learning. We split tasks of the vehicle per service function blocks which have their own role. So it can do partial offloading and user association in a On-Device Edge of the vehicle and in the SBS at the same time . We can get results that service chaining offloading decision gives more optimal energy consumption with low-latency to autonomous vehicle users.**

***Index Terms*—service chaining, offloading decision, on-device edge, autonomous vehicle, deep reinforcement learning**

I. Introduction

With upcoming 5G network era, a edge computing has a key role to give users available resources especially to the delay-intensive and computation-constraint applications. The edge computing located at the nearest from mobile users can address various network requirements. Among them, mission-critical devices as smart factories, smart cities and autonomous vehicles require low energy based ultra-low delay and high reliability services[1][2]. In particular, autonomous vehicles need to process massive amounts of data in real time, from video streaming, voice and text to object recognition, mileage determination, road view, and route planning[3]. Thus, various offloading methods are being studied to solve the latency and information reliability problems that occur at this time.There are several researches related to the offloading methods for autonomous vehicle system with edge computing model[3][4]. First, in paper [5], for efficient task offloading, they research a fog network based car offloading that divides computational tasks to obtain optimal resource efficiency while predicting car movement to reduce offloading delay. At this time, a model-based reinforcement learning algorithm was applied to reduce the offload delay time by predicting the movement of the vehicle. Also in paper [6], they proposed a dynamic task offloading decision method for flexible management of sub-tasks occurring between automobile and mobile edge computing. In addition, they proposed a method for distributing the transmission queue and computational density of each vehicle by optimizing the allocation of computational resources for mobile edge computing.However, in the case of the papers aforementioned above, the requirements of each function blocks such as data collection, preprocessing, and object detection offloaded from an autonomous vehicle to an edge server are not considered. Therefore, we propose a method of offloading decision to an autonomous vehicle or and edge server according to the network resource and energy of each function blocks as a service chain blocks in this paper. To be more specific, the key contributions of our work are as follows:

- We consider service chain blocks which are divided by each functions when users transmit their own tasks on applications to the local edge computing or the small base station(SBS) server.
- We propose an optimal offloading decision way of the service chain blocks on the autonomous vehicle. Service chain blocks are composed of 6 function blocks: data collection, preprocessing, making clusters with user features, cache decision, prediction of the next preferences, and rendered video. When executing the applications, they can select how far to work locally on the On-Device Edge as a EdgeAI and how far to offload to the SBS server.
- We apply a deep reinforcement learning method on our proposed system model, especially Actor Critic algorithm which provide multi-agent system.

II. System Model

We consider the distributed multiple mobile edge computing system with Deep Learning services to provide an optimal task management in Fig.1. We study that each autonomous vehicles have their own EdgeAI named as the On-Device Edge locally. They can select a place to offload their tasks in the local or the offloading to SBS. In this case, we assume SBS has their own edge computing to calculate resources from the vehicles. It only takes the vehicles located within a boundary

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing) *Dr. CS Hong is the corresponding author.

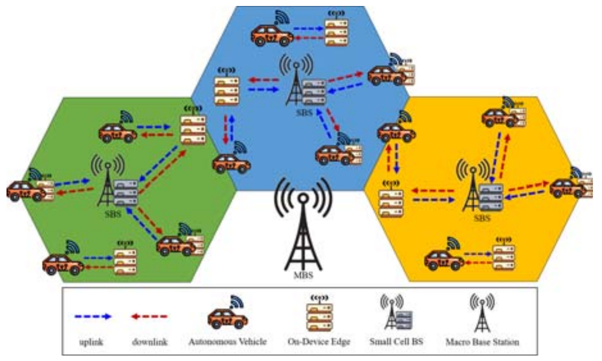

Fig. 1. Network System Model

of the cell. We consider a set of the autonomous vehicles $n = \{1, 2, ..., N\}$ associated with a set of the On-Device Edge $i = \{1, 2, ..., I\}$. Also, SBS servers are composed of a set $j = \{1, 2, ..., J\}$ and we consider a MBS as M_0. The vehicle i chooses the On-Device Edge or SBS/MBS to offload its tasks to reduce delay and computation cost. In this paper, we aim to get a optimal solution for low latency and computation cost through a way of the service chaining offloading decision with the On-Device Edge system.

A. Network Model

The SBS has limited resources to compute whole process of the vehicles. The vehicles n send their tasks to the On-Device Edges i locally. However, when the On-Device Edge is full, then the vehicle should choose offloading way to the SBS or MBS. We consider a partial offloading case also. Thus, in the partial offloading, the vehicle decides how to split its tasks and where to send them after checking available resources in the On-Device Edge, the SBS, and MBS. Here, MBS is only one and it has a same role as the cloud network. Several vehicles try to transmit their resources to the SBS at the same time in the cell. Thus, the offloading scenario has two parts: by the on-device edge and by the SBS. Furthermore, we consider user association with an on-device edge and a base station where $x_{i,j}$ denotes the association variable as follow

$$x_{n,j} = \begin{cases} 1, & \text{if n} \in N \text{ associated with } j \in J \\ 0, & \text{otherwise, execute locally} \end{cases} \tag{1}$$

Each base stations j have different communication and computation resources. Thus, when selecting SBS servers to offload, if it can not execute locally, we should consider a computation requirement of the vehicles within their delay constraints.

B. Communication Model

In our work, the On-Device Edge has a top priority to solve tasks of autonomous vehicles. So all vehicles calculate their tasks on the Edges locally. When there are no available resources, then the vehicle offloads the tasks to the SBS. In addition, there is no interference between the On-Device Edges but only among the SBS servers. So we just consider inter-cell

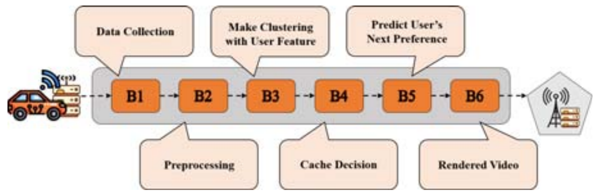

Fig. 2. Service Chain Blocks on the Autonomous Vehicle

interference between cells. It is the orthogonal communication from the vehicle to the Edge and to the SBS. The channel power gain G_{ni} between the vehicle $n \in N$ and the SBS $i \in I$. The transmission uplink data rate of the vehicle is as follow

$$R_{ni} = \frac{W_i}{\sum_{n=1}^{N} x_{n,i}} \log_2(1 + \lambda_{n,i}) \tag{2}$$

where W_i is the bandwidth of the On-Device Edge and $\lambda_{n,i}$ denotes the signal to interference with noise ratio of the uplink. Thus when the autonomous vehicles n in the autonomous vehicles are associated with the same server for offloading, then it should take interference and low data rate.

$$R_{nj} = \frac{W_j}{\sum_{n=1}^{N} x_{n,j}} \log_2(1 + \lambda_{n,j}) \tag{3}$$

Similarly with (2), when the autonomous vehicles are with the SBS server, then we can get above (3). However, in this case, $\lambda_{n,j} = P_n G_{n,j}/N_0$. P_n means the transmission power of the On-Device Edges, and N_0 denotes the basic noise power. Here, we do not consider a downlink data rate from the SBS server, because during a downlink, the computation rate is too little to be ignored in the whole process.

C. Computation Model

There are computation tasks related to a energy consumption and a processing time which is computational and logically independent. An application of autonomous vehicle n is composed of service chain blocks θ as shown in fig. 2. We represents θ^{th} computation serivce chain blocks as $T_n^\theta = \{d_n^\theta, c_n^\theta, t_n^{max}\}, n \in N$. Here, d_n^θ is the input data size of θ^{th} service chain blocks and c_n^θ shows the number of CPU cycles to compute one bit of task T_n^θ. Also, t_n^{max} denotes the maximum latency when the computation tasks are required. Each computation tasks can be split into several service chain blocks to accomplish more optimal computation offloading as seen in Fig. 2. Thus, now we introduce the computation offloading process for the task T_n^θ.

*1) **Local Computing on the On-Device Edge**:* For local computing, task splits, $d_n^\theta - d_{ni}^\theta$ are executed locally on the On-Device Edge i. We represent f_n^θ as CPU cycles per second when processing service chain blocks θ. The computation time $H_{ni}^{\textbf{Local}}(\theta)$ of task T_n^θ is as

$$H_{ni}^{\textbf{Local}}(\theta) = \frac{(d_n^\theta - d_{ni}^\theta) c_n^\theta}{f_n^\theta} \tag{4}$$

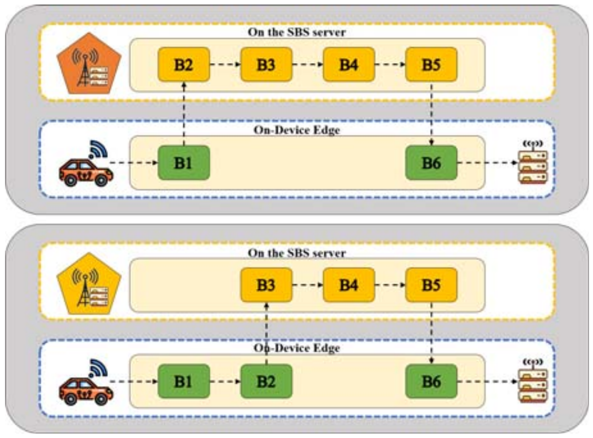

Fig. 3. Optimal Process for the Device and the Edge

Following the equation (4), we can get the energy consumption of each CPU cycle is $(f_n^\theta)^2$. We denotes the energy consumption for task T_n^θ by $E_n^{Local}(\theta)$ as follow

$$E_{ni}^{Local}(\theta) = (d_n^\theta - d_{ni}^\theta)c_n^\theta(f_n^\theta)^2 \tag{5}$$

*2) **Offloading to the SBS server**:* The autonomous vehicle n offloads a part of task d_{nj}^θ to the SBS server j. In general, d_{nj}^θ divide into two parts: $d_{nj}^{'\theta}$ and $d_{nj}^\theta - d_{nj}^{'\theta}$ depicted in Fig. 3. f_{nj}^θ shows the computation resources of SBS server j, allocated to service chain blocks θ for the vehicle n. $H_{nj}^{\mathbf{SBS}}(\theta)$ denotes the execution time of task T_n^θ for the vehicle n as

$$H_{nj}^{\mathbf{SBS}}(\theta) = \frac{d_{nj}^\theta}{R_{nj}} + \frac{(d_{nj}^\theta - d_{nj}^{'\theta})c_n^\theta}{f_{nj}^\theta} \tag{6}$$

With the equation (5), the energy consumption $E_{nj}^{SBS}(\theta)$ is given by

$$E_{nj}^{SBS}(\theta) = \frac{P_n d_{nj}^\theta}{R_{nj}} + (d_{nj}^\theta - d_{nj}^{'\theta})c_n^\theta e_j \tag{7}$$

where e_j shows the energy consumption per CPU cycle of SBS. In this case, the first term represents the transmission energy consumption to offload $P_n d_{nj}^\theta$ of task θ by wireless channel. The second term is about the computation energy consumption of $d_{nj}^\theta - d_{nj}^{'\theta}$ processed by SBS server.

III. Problem Formulation

We formulate the overall problem formulation for the computation offloading and user association. First, the application of the vehicle n is composed of Θ service chain blocks which have a sequential dependency. Each blocks are processed locally on the On-Device Edge and at the SBS concurrently. Thus, the overall time of the application is as

$$H_n^{Total} = \sum_{\theta=1}^{\Theta} max(\sum_{n=0}^{N} x_{n,j}T_{ni}^{Local}, \sum_{n=1}^{N} x_{n,j}T_{nj}^{SBS}) \tag{8}$$

Moreover, the energy consumption of service chain blocks θ is $E_{ni}^{Local}(\theta) + E_{nj}^{SBS}(\theta)$. Thus, overall energy consumption to process the application considering with user association is as

$$E_n^{Total} = \sum_{n=0}^{N} x_{n,j}E_{ni}^{Local}(\theta) + \sum_{n=1}^{N} x_{n,j}E_{nj}^{SBS}(\theta) \tag{9}$$

The objective is to accomplish the optimal service chaining offloading decision in the vehicles with the On-Device Edge. We can define the optimization problem with partial offloading and user association as follow:

$$\begin{aligned}
\underset{f_n^\theta, d_{nj}^\theta, c_n^\theta}{\textbf{minimize}} \quad & \Phi\left(H_n^{Total}, E_n^{Total}\right) \\
& = \sum_{n=0}^{N}\left(\omega_0 \sum_{j=1}^{J}\left((1-x_{n,j})\left(\frac{E_n^{Local}}{H_n^{Local}}\right)\right)\right. \\
& \left. + \,\omega \sum_{j=1}^{J}\left(x_{n,j}\left(\frac{E_n^{SBS}}{H_n^{SBS}}\right)\right)\right) \\
\text{subject to} \quad & 0 \le \omega_0, \omega \le 1 \\
& f_n^\theta \ge 0, \forall n, j, \theta \ge 0 \\
& d_{nj}^\theta \ge 0, \forall n, j, \theta \ge 0 \\
& c_n^\theta \ge 0, \forall n, j, \theta \ge 0 \\
& H_n^{Local}, H_n^{SBS} \ge 0, \forall n \ge 0 \\
& E_n^{Local}, H_n^{SBS} \ge 0, \forall n \ge 0
\end{aligned} \tag{10}$$

IV. Deep Reinforcement Learning Based Service Chaining Offloading Decision

Here, we check available resources on the On-Device Edge and the SBS to decide whether to do a partial offloading based on equation (6) and (7). As Fig. 2, the vehicle has 6 service chain blocks: (1) Data Collection, (2) Preprocessing, (3) Make Clustering with User Feature, (4) Cache Decision, (5) Predict User's Next Preference, and (6) Rendered Video. We consider the a delay-intensive live video application in the vehicle. However, if whole service blocks execute on the On-Device Edge or on the SBS, then it occur a network overhead problem. Thus we care about the partial offloading decision way with Actor Critic to check an available resource and then execute the service chain block in Fig. 3.

*1) **State**(S_n) :* State is composed of two parts: an available computation energy E_n^{Total} and transmission delay time H_n^{Total} following problem (8) and (9). Thus, we can explicit $S_n = (E_n^{Total}, H_n^{Total})$.

*2) **Action**(A_n) :* Action as a goal of our work is a optimal value when Reward is a minimum with State S_n. So, A_n denotes $\pi_n(A_n|S_n)$ which is a decision value to get a service chaining offloading for autonomous vehicle n with the On-Device Edge i and with the SBS server j.

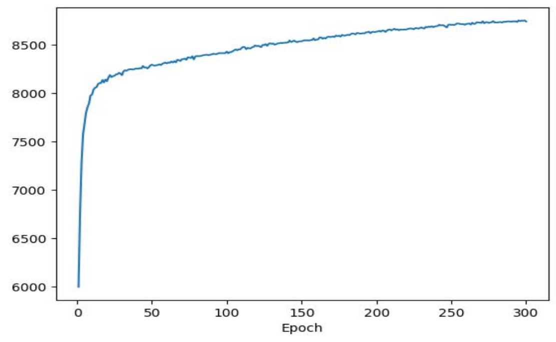

Fig. 4. Checking whether proposed method work well or not

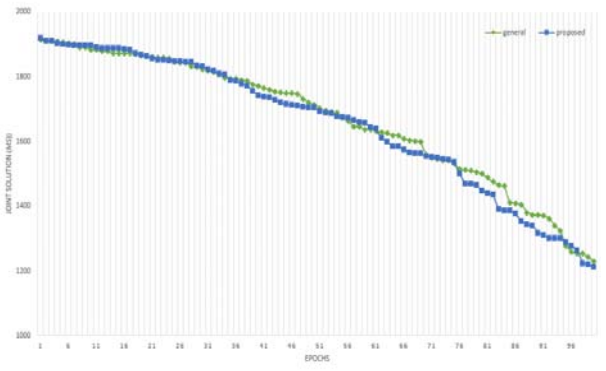

Fig. 5. Comparison the proposed model with general model during 100 epochs

*3) **Reward**(R_n)* : As explained by the problem (10), a optimal reward value is a minimum reward R_n given A_n during many episodes.

Algorithm 1 shows the process of service chaining offloading decision, $\pi_n\left(A_n|S_n\right)^*$ using Actor Critic.

Algorithm 1 Service Chain Blocks Offloading Decision process with Actor Critic

```
0: **for** Till optimal $\pi_n\left(A_n|S_n\right)^*$ **do**
0:     **for** $\forall n \in N$ **do**
0:         **Action**: $A_n\ \pi_n\left(A_n|S_n\right)$
0:         **Get**: $o_n = \langle S_n, A_n, R_n, S_{n+1}\rangle$
0:         **Assess**: $A\left(\theta_n\right)$ with Eq. 10
0:         **Policy**: $\theta \leftarrow \theta + A_n Q_\omega\left(S, A\right)\nabla_\theta \ln \pi_n\left(A|S\right)$
0:         **Update**: $\theta_n = \theta_n +$ **Policy**
0:         **ActionValue**: $\delta_t \leftarrow r_t + \gamma Q\left(S, A\right)\nabla_\theta \in \pi_\theta\left(A, S\right)$
0:         **Update**: $\delta_t = \delta_t +$**ActionValue**
0:     **end for**
0:     **Update**: $A \leftarrow A_{n+1} and S \leftarrow S_{n+1}$
0:     **Append**: $o_n \in O$
1: **return** $\pi_n\left(A_n|S_n\right)^*, O$
```

V. Evaluation

We consider 5 autonomous vehicles with 6 service chain blocks and the edge computing by Kubernetes. We use a movie-lens dataset for mobile user's applications. Also, on each Edge, we input a number of mobile users uniform distribution random variable. Fig.4 represents the proposed algorithm works well or not. During 300 epochs, our proposal works well on increasing of learning time almost 9000 times. We use the same learning rate, 0.001, on whole On-Device Edges. Based on Fig. 4, we can check our joint communication and computation model with Actor Critic model. In Fig.5, comparing with the general case, we can check our proposed method little bit fast to offload tasks during 100 epochs.

VI. Conclusion

Here, we get the optimal service chaining offloading decision on the autonomous driving system with Actor Critic model. However, we only consider the video live streaming service. So some limitations are to calculate various network requirements concurrently from the vehicle such as sensors, camera, voice, etc. We did not consider the waiting time when the vehicle selects the task offloading or the task transmission to the On-Device Edge or the SBS. Also, we did not use real datasets on the evaluation. In future, we will consider various applications in the vehicles. We should consider a model compression on each service chain blocks to reduce a utility cost to propose a framework of the EdgeAI system.

References

[1] X. Wang, Y. Han, V.C.M Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, pp.1, Jan 2020.

[2] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward Computation Offloading in Edge Computing: A Survey,"IEEE Access, pp.131543-131558, vol.7, August 2019.

[3] J. Park, S. Samarakoon, M. Bennis, and M. Debbah,"Wireless Network Intelligence at the Edge,"Proceedings of the IEEE, vol.107, Issue.11, Nov 2019

[4] A. Ndikumana, N.H.Tran, D.H.Kim, K.T.Kim, and C.S.Hong,"Deep Learning Based Caching for Self-Driving Cars in Multi-Access Edge Computing,"IEEE Transactions on Intelligent Transportation Systems, pp.1-16, March 2020.

[5] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu,"Exploiting Moving Intelligence: Delay-Optimized Computation Offloading in Vehicular Fog Networks,"IEEE Communication Magazine, vol.57, pp.49-55, issue.5, May, 2019.

[6] X. Huang, K. Xu, C. Lai, Q. Chen, and J. Zhang "Energy-efficient offloading decision-making for mobile edge computing in vehicular networks,"EURASIP Journal on Wireless Communications and Networking, 2020.

[7] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint Computation Offloading and User Association in Multi-Task Mobile Edge Computing,"IEEE Transactions on Vehicular Technology, vol.67, no.12, Dec 2018.

[8] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, Z. Peng, L. Pan, and Y. Zhang, "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks,"IEEE Access, pp.2169-3536, vol.4, August, 2016.

[9] Z. Ding, P. Fan, and H.V. Poor, "Impact of Non-Orthogonal Multiple Access on the Offloading of Mobile Edge Computing,"IEEE Transactions on Communications, vol.67, no.1, Jan 2019.