Cost Minimization with Offloading to Vehicles in Two-tier Federated Edge and Vehicular-Fog Systems

Ying-Dar Lin, Jui-Chung Hu, Binayak Kar, and Li-Hsing Yen
Department of Computer Science, National Chiao Tung University, Taiwan.
Email: ydlin@cs.nctu.edu.tw, tony84822.cs02@g2.nctu.edu.tw, bkar@nctu.edu.tw, lhyen@cs.nctu.edu.tw

***Abstract*—Vehicular-fog system consists of vehicles with computing resources that are mostly under-utilized. Therefore, an edge system may offload some workloads for remote execution at nearby vehicular-fogs. Whether this is cost-effective depends on not only the costs and computation capacities of vehicles but also the amount of workloads and associated latency constraint. In this paper, we consider a two-tier federated Edge and Vehicular-Fog (EVF) architecture and aim to minimize overall cost while meeting latency constraint by setting up an appropriate offloading configuration. We model this to a single-objective mixed integer programming problem. To solve this mixed integer problem in real time we propose an iterative greedy algorithm using the queuing model. The results show, our proposed architecture reduces the cost of vehicular-fogs by 40–45% and the total cost by 35–40% compared to the existing architecture and help the edge to provide services beyond its capacity with specified latency constraint.**

***Keywords*— Edge, vehicular-fog, offloading, cost, latency**

I. Introduction

The rapid evolution of communication technologies helps in bringing the computing resources closer to the user day by day. As an example in the early days we used to rely on cloud computing technology that provides services such as Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) and Service-as-a-Service (SaaS) in datacenter [1]. Recently, service providers are re-architecting their central offices and base stations as a datacenter, so that the computation services are provided by the Mobile Edge Computing (MEC) [2] that reduce the communication and computation latencies as compared to cloud [3]. However, there is a limitation of resources in the edge which is raised as a big issue as the objective of the service provider is to meet the users' demand by handling traffic dynamically and avoid offloading to the cloud to satisfy the required latency.

At the same time, due to the advancement of the electric automobile industry and Information Communication Technology (ICT), Internet of Vehicles (IoV) [4] becomes the new issue in this generation. Because these vehicles are no more used only for transportation but can be a part of network communication and computation as having high computing resources. Vehicular Ad-Hoc Network (VANET) [5] is to build Cloud Edge Vehicular-Fog and maintain a communication network for a set of moving vehicles without any central base station, and provide state-of-the-art services such as traffic management.

However, these vehicles remain unutilized due to lack of infrastructure support and high mobility. Vehicular-fog computing [6] is an emerging technology, which provides these vehicles the necessary infrastructures to utilize their resources and get benefited financially by federating with the edges. An overview of the cloud, edge, and fog communication network is given in Fig. 1, where the edge can federate with the cloud as well as with the fogs to expand its available resource capacity.

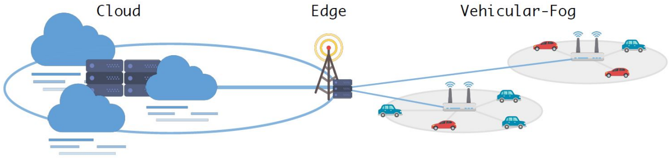

Fig. 1: Overview of cloud, edge and vehicular-fog communication network

In this paper, we address the above discussed two issues, *i.e.*, 1) capacity and latency limitations of the edge and 2) unutilized resources of the vehicles, together and propose a two-tier Edge and Vehicular-Fog (EVF) systems where the fogs are managed by a centralized road side unit (RSU) to manage the offloaded traffic from edge to fogs. When edge does not have adequate resource capacity to meet users'demand, it will offload the traffic to vehicular-fogs and the vehicles in the fog will send back results to the edge after completion of computation. In this paper, by determining the dynamic traffic offloading ratio from edge to vehicular-fogs and capacity of edge we will minimize the total cost from the perspective of service providers *i.e.*, edge. This process will have benefited in three ways: *First*, the edge does not face any shortage of resource capacity, *Second*, it will meet the required latency demand as vehicular-fogs are closer to the users and have high computing resources for computing, and *Third*, the resources of vehicles will get utilized better.

The remainder of the paper is organized as follows. Section II reviews the related works. Section III provides a brief overview of the proposed Two-tier EVF architecture. Section IV presents the proposed cost minimization problem in edge and vehicular-fog federation. Section V provides the solution and VI evaluate performance of our proposed architecture. Section VII concludes the paper.

II. Related Works

Recently, considering vehicular-fog computing technology, various concepts and architectures have been proposed. There are few novel types of architectures about vehicular-fog that we will discuss in this section.

Hou *et al.*, proposed a promising model that utilizes vehicles as infrastructures to attain more available resources and enhance the achievable capacities. They analyzed the scenarios of both slow-moving vehicles and parked vehicles for computation and communication [6]. In [7], Gu *et al.*, proposed a two-tier datacenter architecture where one is remote datacenter and the other is vehicular datacenter (VDC) in parking lots. The redundant storage resources in VDC can be leveraged to alleviate the burden on conventional datacenter. A programmable and flexible framework named BEGIN (Big data enabled EnerGy-efficient vehicular edge ComputiNg) was proposed in [8] to improve energy efficiency with big data. In [9], Wang *et al.*, proposed a vehicle-based computation relaying scheme for computation offloading in vehicular networks. Their objective was effective utilization of computing resources available in surrounding smart vehicles of the mobile device in the highly dynamic network environment. Most of these research papers focused on how the resources of the vehicles should get utilized. However, in the proposed two-tier EVF architecture, we are considering the RSU to manage the fog and also considering the latency limitation.

III. Two-tier Federated Edge and Vehicular-Fog Architecture

In this section, we propose two-tier EVF architecture adopted by the VFC as illustrated in Fig. 2. In this architecture, we assume that each RSU in the edge's coverage area can form a vehicular-fog by establishing communication with the vehicles around it. These RSUs are considered as the fog managers or fog nodes of the vehicular-fog that manage fogs and do not have any computing resources. To establish the communication for offloading, the edge will send specific requirements to each fog nodes in the corresponding vehicular-fogs and fog nodes also send their own context-aware information such as the number of vehicles, computation capacity, cost, etc. of the fog to the edge.

In this system, we assume the edge will receive all of traffics from the users with specified latency constraints of each traffic. Initially, the requests can be processed locally by the edge server. With the increase in the request from the users the edge will expand its capacity and try to utilize it fully. However, when the demands exceed the edges capacity limitation, the edge will choose to offload the traffic to vehicular-fogs located inside its coverage area to fulfill the user requirements. After receiving the request from the edge, the fog node of the fog will play the role of orchestrator to manage the traffic to different vehicles in their fog. Thus, the proposed federation system aggregates the resources of idle individual vehicles to utilizes quite a lot of computation potentialities. Following are the few assumptions of our EVF architecture in various scenarios:

- In a highly crowded area like metropolitan cities, there is a possibility the edge will receive more request due to a huge population. However, in such populated areas, there must be enough number of vehicles available to be part of the fogs and participate in edge and fog federation.
- During off-peak hours when traffic inputs are relatively very low, the edge can offload to the fogs instead activating its own servers. Otherwise, the high capacity servers of the edges will remain un-utilized.

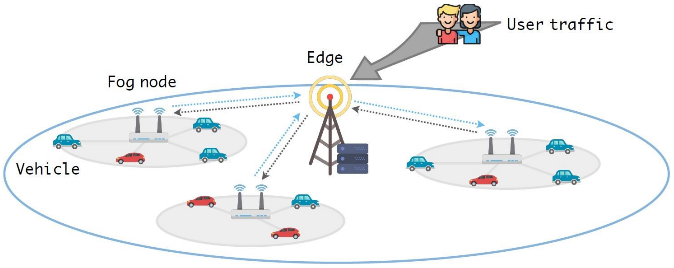

Fig. 2: Two-tier federated edge and vehicular-fog architecture

TABLE I: Notations of the model

Notation	Description
Sets and Elements	
E	Edge node
N_E	Maximum servers in E
n_E	Number of active servers in E
F	Set of vehicular-fogs
f_i	Vehicular-fog in F
N_i	Number of vehicles present in vehicular-fog i
n_i	Number of used vehicles in vehicular-fog i
$v_{i,j}$	Vehicle j in vehicular-fog i is used or not
Traffic	
λ	Input traffic from users to E
λ_E	Total traffic in E
λ_i	Total traffic in vehicular-fog i
ϵ	Ratio of the output/input traffic
β_i	Offloading probability from E to vehicular-fog i
R_E	Transmission rate from E
R_F	Transmission rate from vehicular-fog
Capacity	
μ_E	Capacity of single server in E
μ_V	Capacity of each vehicle in F
Cost	
c_E	Computation cost of single server in E
$c_{i,j}$	Mean cost of vehicle j in vehicular-fog i
c_{total}	Total cost
Latency	
l_E	Computation latency of E
l_i	Computation latency of vehicular-fog i
$l_{E,i}$	Communication latency from E to vehicular-fog i
$l_{i,E}$	Communication latency from vehicular-fog i to E
L^{max}	Maximum latency constraint
Distance and Speed	
$D_{E,i}$	Distance between edge E and vehicular-fog i
C	Speed of light

IV. Problem Formulation

First, we assume our two-tier system providing Infrastructure as a Service (IaaS) with computation resource to deal with the request from users. The computation workload in edge can either be executed locally by the edge servers or offloaded to vehicular-fogs. But the determination of allocated workload of available resources and estimation of the offloading ratio from edge to fog is really a big challenge. In this section, we formulate it to a workload and capacity optimization problem with the objective of minimizing the total cost from the edge perspective with the constraint of end-to-end latency. The variables and notations used in this paper are discussed in Table I.

We assume only one edge E having maximum N_E number of servers, and each server has a resource capacity μ_E with cost c_E. And a set of vehicular-fogs $F = \{f_i | 1 \leq i \leq |F|\}$. In each vehicular-fog f_i, there are N_i number of vehicles having capacity μ_V. For the specific requirements in user request, let λ be the input traffic from users to edge and L^{max} be the maximum latency constraint.

A. Latency Estimation

Assuming that computation workload in the edge are given according to the Poisson process, therefore the edge and vehicular-fogs traffic model can be considered as an M/M/c queuing model for computation and an M/M/1 queuing model for communication.

1) Computation Latency: Let l_E be the computation latency of edge and n_E be the number of active servers in the edge. Let λ_E be the total traffic processed by the edge which can be estimated as

$$\lambda_E = \lambda - \sum_{i=1}^{|F|} \lambda \cdot \beta_i,$$

where β_i is the offloading probability from the edge to vehicular-fog i. Then the computation latency of the edge which can be estimated as

$$l_E = \frac{\mathrm{C}(n_E, \lambda_E/\mu_E)}{n_E \cdot \mu_E - \lambda_E} + \frac{1}{\mu_E}, \tag{1}$$

where $\mathrm{C}(c, \lambda/\mu)$ is Erlang's C formula [10] determined as

$$\mathrm{C}(c, \lambda/\mu) = \frac{1}{1 + (1-\rho)\left(\frac{c!}{(c\rho)^c}\right)\sum_{k=0}^{c-1}\frac{(c\rho)^k}{k!}},$$

where λ is arrival traffic rate, μ is service rate and ρ means utilization given by $\rho = \dfrac{\lambda}{c \cdot \mu}$. Let l_i be the computation latency of vehicular-fog i and can be estimated as

$$l_i = \frac{\mathrm{C}(n_i, \lambda_i/\mu_V)}{n_i \cdot \mu_V - \lambda_i} + \frac{1}{\mu_V}, \tag{2}$$

where λ_i is the total traffic executed in the edge. $n_i = \sum_{j=0}^{N_i} v_{i,j}$ would be the total number of vehicles in use in the vehicular-fog i, and

$$v_{i,j} = \begin{cases} 1, & \text{if vehicle } j \text{ in fog } i \text{ is in use,} \\ 0, & \text{otherwise,} \end{cases}$$

where, $v_{i,j}$ is a binary number to decide this vehicle is in use or not.

2) Communication Latency: Let R_E be transmission rate from the edge to a fog, $D_{E,i}$ is distance between the edge and a fog and C would be speed of light. Let R_F be transmission rate from a fog to the edge, and ϵ is the mean ratio of the input traffic and output traffic. The communication latency from the edge to vehicular-fog i can be estimated by (3).

$$l_{E,i} = \frac{1}{R_E - \lambda_i} + \frac{D_{E,i}}{C}, \tag{3}$$

where $\lambda_i < R_E$. The communication latency from vehicular-fog i to edge can be estimated by (4).

$$l_{i,E} = \frac{1}{R_F - \lambda_i \cdot \epsilon} + \frac{D_{E,i}}{C}, \tag{4}$$

where $\lambda_i \cdot \epsilon < R_F$. However the propagation delay compared to transmission delay is not essential, so we would ignore it in the simulation.

B. Objective Function

The objective function of our optimization problem is presented in (5), where c_{total} is the total cost which is the summation of the cost in edge and vehicular-fogs. 1) First part of (5) is the product of number of active servers and the cost of single server in the edge. 2) Second part of (5) is the summation of the product of binary value of vehicle in used or not and the cost of vehicle in each vehicular-fog.

$$\min \quad c_{total} = n_E \cdot c_E + \sum_{i=1}^{|F|}\sum_{j=1}^{N_i} v_{i,j} \cdot c_{i,j}, \tag{5}$$

$$\text{s.t.} \quad \lambda - \sum_{i=1}^{|F|} \lambda \cdot \beta_i < n_E \cdot \mu_E, \tag{6}$$

$$\lambda \cdot \beta_i < n_i \cdot \mu_V, \tag{7}$$

$$n_E \leq N_E, \tag{8}$$

$$v_{i,j} \in \{0, 1\}, \tag{9}$$

$$l_E \leq L^{max}, \tag{10}$$

$$l_{E,i} + l_i + l_{i,E} \leq L^{max}. \tag{11}$$

The constraints in (6) and (7) ensure the executed traffic would lower than the total available capacities in edge or in vehicular-fog. The constraint in (8) ensures the number of activated server would not be higher than maximum servers in edge. Wheather the vehicle is chosen to be used or not is estimated by (9). For the latency constraint from user requirements, we first considered the delay in edge should be bounded L^{max}, as presented in (10). The constraint in (11) ensures the sum of communication latency from edge to fog and fog to edge, and the computing latency in the fog should be bounded to maximum latency.

V. Solution Approach

The workload allocation problem discussed in the previous section is a mixed integer programming (MIP) problem. Certainly, such MIP problem can be proven to be NP-hard [11] due to its high complexity. For such complex problem and users' dynamic request and given constraints, we need an adaptable and real-time solution that is efficient both in terms of cost and time. Hence, we propose an iterative greedy algorithm that can be executed in a distributed way to match our two-tier systems. By adapting queuing system, in this algorithm, we assume, with the increase in the capacity of the system the more traffics be handled. This process helps vehicular-fogs to utilize more vehicles to handle the traffics.

Algorithm 1 Iterative Greedy Algorithm

```
Require: Total traffic $\lambda$; Maximum latency $L^{max}$;
$S \leftarrow \emptyset$; $K \leftarrow (E \cup F)$
repeat
    for all $k \in K$ do
        if $flag = true$ then
            $(\lambda_k, c_k) \leftarrow$ Best-Traffic Algorithm $2(k, \lambda, L^{max})$
        else
            $(\lambda_k, c_k) \leftarrow$ Max-Traffic Algorithm $2(k, \lambda, L^{max})$
        end if
    end for
    if $K = \emptyset$ then
        $flag \leftarrow false$
        continue
    end if
    $G \leftarrow \emptyset$; $U \leftarrow K$;
    repeat
        select $u \in U$ that maximizes $\lambda_u / c_u$
        if $\sum_{g \in G} \lambda_g + \lambda_u \leq \lambda$ then
            $G \leftarrow G \cup \{u\}$
        end if
        $U \leftarrow U \setminus \{u\}$
    until $U = \emptyset$
    $\lambda \leftarrow \lambda - \sum_{g \in G} \lambda_g$
    $S \leftarrow S \cup G$
    $K \leftarrow K \setminus G$
until $\lambda = 0$
$(\lambda_e, c_e) \leftarrow$ Max-Traffic Algorithm $2(E, \lambda, L^{max})$
if $\lambda_e \geq \lambda$ and $c_e \leq \sum_{s \in S} c_s$ then
    return $\{e\}$
else
    return $S$
end if
```

The proposed Iterative Greedy Algorithm is presented in Algorithm 1. First, we initialize the solution set S which would contain the results that we expect and the set K that consist of the edge and vehicular-fogs. When the users' request arrived at the edge, the edge would send the information such as total traffic λ and maximum latency constraint L^{max} to each fog nodes in its area. Then each element in K used Best-Traffic Algorithm (BTA) to calculate the best traffic λ_k and the price c_k it would charge. Then the edge would choose candidates by the cost–traffic ratio order, *i.e.*, the one whose traffic per unit cost was higher would be chosen first. After a candidate was chosen if it is a fog, the traffic λ_k offloaded to it and deducted from the total traffic λ. If there are still remained traffic, the edge would send the current traffic again to fogs nodes whom had not collaborated with until there is no any traffic. To handle traffic by edge using its own servers can be done after comparing the cost between the edge server c_e and the solution set S, as discussed above and the lower one will be preferred.

However, when the user traffic grows up, and the BTA would not be able to carry out the total traffic, the Max-Traffic Algorithm (MTA) which is a modified version of BTA, is used. This modified algorithm only focus on maximum traffic that the vehicular-fogs can carry out to handle more traffic. In other words, each vehicular-fog node would find out the traffic by using the two versions of *i.e.* BTA and MTA in different scenario. These two algorithms are mostly similar but have little difference only in choosing the vehicles.

Algorithm 2 *Distributed*: Best-Traffic Algorithm

```
Require: Vehicular-fog $k$; Total traffic $\lambda$; Maximum latency $L^{max}$;
for all vehicle $v \in k$ do
    Obtain the usage time $t_v \leftarrow (p_{v,k}^{ini} - p_{v,k}^{th}) / p_{v,k}^{rate}$
    $t_v \leftarrow \min(t_v, L^{max})$
end for
$G \leftarrow \emptyset$; $U \leftarrow$k;
for $v \in U$ do
    select $v$ that maximizes $t_v / c_v$
    if $t_v \geq L^{max}$ then
        $G \leftarrow G \cup \{v\}$
        $U \leftarrow U \setminus \{v\}$
    else
        break
    end if
end for
Calculate total latency $L'$ with $\lambda$, $|G|$ using equation (2), (3), and (4)
if $L' > L^{max}$ then
    $\lambda^{max} \leftarrow$ Bisection Method$(\lambda, L^{max}, |G|)$
    for $v \in U$ do
        select $v$ that maximizes $t_v / c_v$
        $\lambda' \leftarrow$ Bisection Method$(\lambda, L^{max}, |G \cup v|, t_v)$
        if $\lambda' > \lambda^{max}$ then
            $\lambda^{max} \leftarrow \lambda'$
            $G \leftarrow G \cup \{v\}$
            $U \leftarrow U \setminus \{v\}$
        else
            break
        end if
    end for
else
    Get the optimal $G$ with decreasing vehicles strategy
end if
return $(\lambda^{max}, \ \sum_{v \in G} c_v)$
```

In these algorithms, initially, we calculate the usage time by t_v for each vehicle. In BTA, we select the vehicle whose usage time per cost (t_v / c_v) was most one-by-one until the usage time is lesser than the latency constraint. But in MTA, we only focus on usage time instead of cost, so change to select the vehicle whose usage time was most one-by-one. Then, we calculate the latency L' by using the vehicles we selected before. If the latency was higher than latency constraint which means the total traffic cannot be executed in that vehicular-fog. In such situation we used the Bisection method [12] to find out how much traffics could be handled by these vehicles within the specified latency limit. For BTA, we added the vehicles whose t_v / c_v was most and for MTA the vehicles having high t_v was added one-by-one to the used vehicles set G. Then estimate the traffic again by using the Bisection method until there is no increase for traffic. To maintain the low cost, we simply remove the vehicles having a high cost in the fog one by one until there is a violation of latency limitation. This helps to satisfy the latency constraint and minimize the cost. The detail

TABLE II: Simulation parameters

Name	Description	Default Value
N_E	Maximum servers in E	5
N_i	Number of vehicles present in vehicular-fog i	10-100
μ_E	Capacity of single server in E	200 MB/s
μ_V	Capacity of each vehicle in F	5 MB/s
c_E	Computation cost of single server in E	\$ 200
$c_{i,j}$	Mean cost of vehicle j in fog i	\$ 1-50
R_E	Transmission rate from edge	1250 MB/s
R_F	Transmission rate from fog	1250 MB/s
ϵ	Ratio of the output/input traffic	0.01

BTA algorithm is discussed in Algorithm 2. We skip the MTA algorithm due to its similarity to BTA and page limitation.

VI. Simulation Results

A. Simulation Settings

In this experiment, we considered a single edge has a coverage area of 100 km and there are 10 vehicular-fogs distributed within edge's coverage area. Each edge consists of 5 servers and each vehicular-fog consist of 10 to 100 number of vehicles. The capacity of each server of the edge is 200 MB/s and each vehicle is 5 MB/s. The cost of the vehicles is assigned randomly with a range of \$ 1-50. In the latency part, we assume our transmission rate between the edge and fog was set in 1250 MB/s. Compared to the transmission delay, we assume the propagation delay was negligible in this scenario, hence we ignore it. The details of the parameter settings are discussed in Table II.

B. Performance Analysis

1) Cost Analysis: Fig. 3 shows that with the increase in arrival traffic the total cost increases. While the traffic was larger than 1000, all of the servers in edge would be activated. After which the remainder of traffic would be offloaded to the vehicular-fogs, so the total cost was affected by the total cost charged by the vehicular-fogs. However, when the input traffic was below 1000, the edge will choose whether to process by its own server or offload to the fogs. Starting from 0, the arrival traffic would be preferring to offload to the vehicular-fogs instead of processing in local edge server. That's because the capacity of each vehicle is 5 whereas the capacity of a single server is 200. When there are low traffic and processed by the edge server, most of the capacity will remain unutilized however, it will be charged fully. For example, when the incoming traffic was 100, if we carried out all of traffic in edge, we had to open one server to serve it, and then this would take us 200 cost. If it offload the same traffic to the vehicular-fogs, the cost will be less and this process will reduce the total cost.

2) Single-Server Edge vs. Multi-Server Edge: In this part, we carried out two different experiments. In one case, we considered the total capacity of edge server into a single server named Single-Server Edge (SSE) and in the other one is our default setting, *i.e.*, total capacity is equally divided among 5 servers named as Multi-Server Edge (MSE). The results, in Fig. 4 show, the cost of the MSE was lower than the SSE. This is because activating one server in SSE cost more compared to activating one server in MSE. Although in high traffic both are equally expensive, however, in low traffic scenarios, SSE is more expensive and it will mostly depend on the fogs.

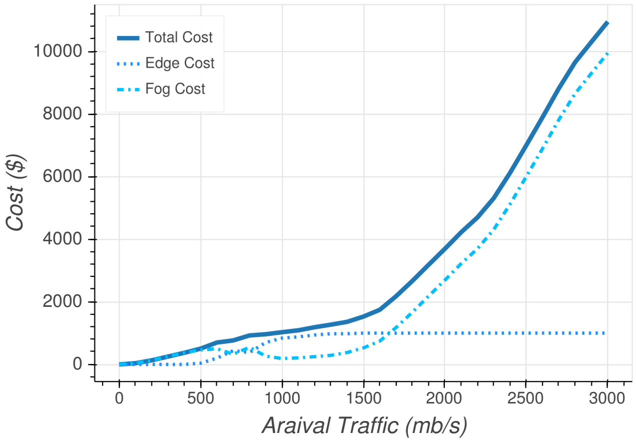

Fig. 3: Total cost affected by arrival traffic

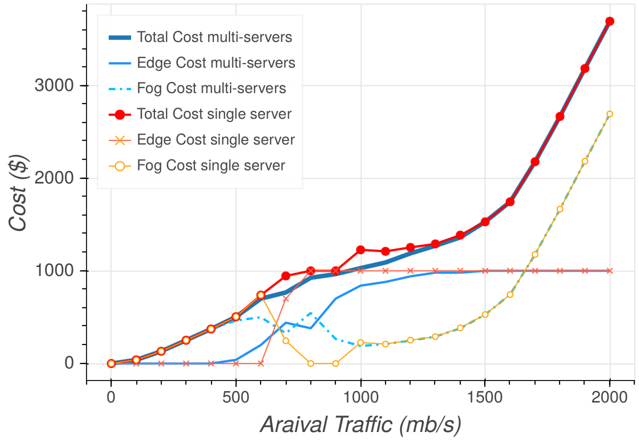

Fig. 4: Total cost affected by Single-Server Edge and Multi-Server Edge

3) Cost Analysis with Latency Constraint: The results in Fig. 5 and Fig. 6 show the cost consumption in low and high traffic, respectively, with different latency limitations. In both cases, with the decrease in latency the total cost increases. In Fig. 5, we consider the traffic of 100 MB/s, and in such low input traffic, the edge chose to offload it to the fogs in high latency limit *(i.e.*, higher than 0.2 sec), to save the cost. However, in low latency limit, it chooses to handle the traffic by its local servers to satisfy the latency constraint, as a result, the cost goes up. While the maximum latency constraint started decreasing, if we let all of the traffic offloaded to vehicular-fogs, we have to increase more vehicles in vehicular-fogs to meet the latency constraint, which means it would take more cost. Fig. 6 shows, in high traffic inputs of 1500 MB/s with maximum latency constraint was lower than 0.3 sec, all of the computation resources in edge and vehicular-fogs would be in use and the total cost increased rapidly. Because the latency constraint is too low, we have to increase capacities to lower

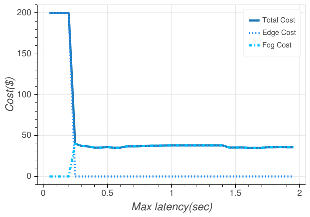

Fig. 5: Total cost affected by maximum latency constraint in 100 traffic

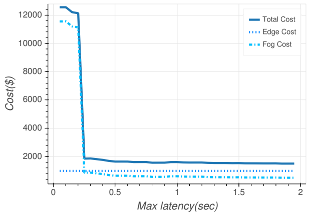

Fig. 6: Total cost affected by maximum latency constraint in 1500 traffic

down the computation latency to meet the latency constraint.

4) Two-tier EVF w/ RSU vs. Two-tier EVF w/o RSU: In this part, we compared our proposed architecture (Two-tier EVF with RSU) with the two-tier vehicular-edge without RSU architecture. In two-tier vehicular-edge without RSU scenario, the edge would directly offload the traffic to vehicles without passing through the vehicular-fog node. So the edge needed to know every vehicles in its area and calculate the offloading ratio to each vehicle. This is an important difference because in our architecture the edge does not need to attain information about all of the vehicles in its coverage area. In our architecture edge just need to know about all of the vehicular-fogs and fog nodes and the RSU will keep the necessary information of its fog.

Fig. 7 shows that the total cost in our system would always lower than in Two-tier w/o RSU architecture. When the arrival traffic was proximity to 2500, the cost of the fogs can be decreased by 40–45% as a result the total cost can be reduced by about 35–40% in our systems. The reason behind this result is the utilization of the vehicles in EVF w/o RSU is much lower than the vehicles in EVF w/ RSU as the latency in the former model was estimated by M/M/1 model whereas M/M/c was used for our model where c is considered as the number of vehicles in the vehicular-fog. From the result, we can conclude that if we consider the centralized vehicle management mechanism where the vehicles in the vehicular-fogs are managed by the fog managers, we will have better performance in terms of utilization of resources of the vehicles, as a result, the cost decreases.

VII. Conclusions

In this paper, we have presented a two-tier EVF architecture in the edge-based vehicular network where each vehicular-fog is managed by fog node. Base on the architecture, we investigated the traffic offloading scheme and formulated it as a mixed integer programming problem. We proposed an iterative greedy approach to solve the problem. The results show, the proposed two-tier EVF with RSU architecture reduced the cost of vehicular-fogs by 40–45% and the total cost by 35–40% compared to two-tier architecture without RSU. Our simulation results demonstrate in low traffic inputs the fogs will reduce the total cost by avoiding local computation in the edge. And in high traffic inputs the edge provide the service by offloading the request to the vehicular-fogs.

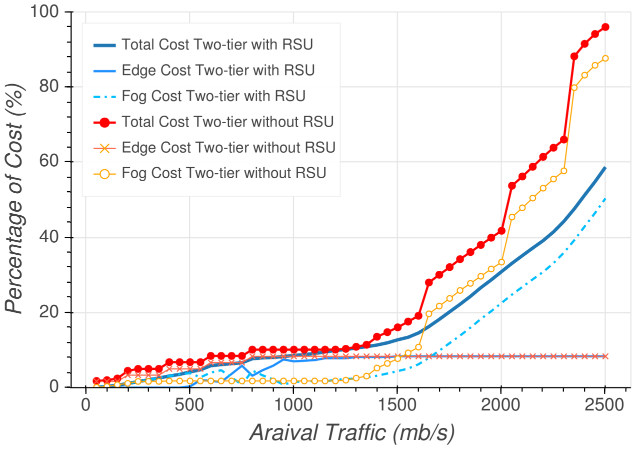

Fig. 7: Total cost compared in different system

In the future work, we will consider the energy consumption issues of the vehicles in the vehicular-fog as they are battery dependent and also we will consider the vehicle arrival and departure rate into the fogs.

Acknowledgements

This work was supported in part by H2020 EU-TW 5G-Coral Project under the project number MOST 107-2923-E-009-005.

References

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, p. 7–18, 2010.

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[3] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[4] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C.-T. Lin, and X. Liu, "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspectss," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.

[5] S. ur Rehman, M. A. Khan, T. A. Zia, and L. Zheng, "Vehicular ad-hoc networks (vanets) - an overview and challenges," *Wireless Networking and Communications*, vol. 3, no. 3, pp. 29–38, 2013.

[6] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, , and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[7] L. Gu, D. Zeng, S. Guo, and B. Ye, "Leverage parking cars in a two-tier data center," *IEEE Wireless Communications and Networking Conference*, pp. 4665–4670, 2013.

[8] Z. Zhou, H. Yu, C. Xu, Z. Chang, S. Mumtaz, , and J. Rodriguez, "Begin: Big data enabled energy-efficient vehicular edge computing," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 82–89, 2018.

[9] Z. Wang, Z. Zhong, D. Zhao, , and M. Ni, "Vehicle-based cloudlet relaying for mobile computation offloading," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 11 181–11 191, 2018.

[10] K. Leonar, *Queueing Systems. Volume 1: Theory*, 1975.

[11] R. KannanClyde and L. Monma, *On the Computational Complexity of Integer Programming Problems*, 1978, vol. 157.

[12] R. L. Burden and J. D. Faires, *Numerical Analysis: 2.1 The Bisection Algorithm*, 1985.