Token Based Energy-efficient Offloading Schemes for IoV Networks

Pranshu Srivastava*, Kaustubh Ijardar*, Anuj Joshi*, Nikumani Choudhury*, Anakhi Hazarika†
* Dept. of Computer Science and Information Systems Birla Institute of Technology & Science Pilani, Hyderabad, India
† Dept. of ECE, Indian Institute of Information Technology Guwahati, India
Email: {h20211030083,h20211030081, h20211030119, nikumani}@hyderabad.bits-pilani.ac.in, anakhi.hazarika@ieee.org

***Abstract*—The emergence of applications in vehicles requires computational capability which poses a major challenge in mobile edge computing. In this paper, we have proposed the token-based predictive offloading scheme with the aim of providing the optimal cost of offloading and reducing the average delay. Specifically, the proposed scheme uses a token-based approach for offloading the data from vehicles to MEC(Mobile Edge Computing) server. We have designed a separate table for MEC servers to show the status of consumed tokens and available tokens and another table for vehicles to store the status for another vehicle for vehicle to vehicle (V 2 V) communication. Dedicated Short Range Service (DSRC) technology is used to facilitate the communication between vehicle to vehicle and vehicle to infrastructure(e.g.toll gate). Extensive simulations are conducted in highway scenarios and the results demonstrate the superiority of this offloading scheme. The proposed scheme achieves low delay performance and decreased computation cost over other competing schemes in typical urban and highway scenarios.**

***Index Terms*—Internet of Vehicles, Mobile-edge computing, Data Offloading, DSRC, Tokens .**

I. Introduction

The new era of the Internet of Things (IoT) applications [1], [2] is driving the evolution of conventional Vehicle Adhoc Networks into the Internet of Vehicles (IoV) [3]–[6]. In the IoV networks, vehicles and the embedded sensory devices are the smart objects that are connected with other similar smart/IoT devices for data sharing and communication. Additionally, the applications used in vehicles [7]–[10] are computationally intensive which requires high computation so to satisfy the computational requirement is a major challenge. Mobile edge computing facilitates offloading of task at the edge of the network, as a result we get reduction in latency, fast response, high reliability services [11]–[13]. The major challenges in mobile edge computing is how to offload the task, some task can be computed locally using the resources of vehicles but that depends on types of task to be computed [14], [15]. Generally there are three types of task which are crucial application, high priority application, low priority application. Crucial applications needs to be processed urgently as they are crucial like vehicle control, accident prevention. High priority applications prioritized after the crucial applications, these applications are like navigation, optional safety. Low priority application prioritized after high priority application as if they get delayed during processing then there is no damage (e.g. multimedia application). Crucial application need to be processed very fast so they can be allotted to our own vehicle computing resources for processing because if they would be allotted to edge server then there may be possibility that they may get delayed during processing and by then damage or accident may occurs. To facilitate the communication between vehicles to vehicles and vehicles to infrastructures, we have used DSRC(Dedicated Short Range Communication) [16] technology which is short range communication and helps in communicating information. Vehicles and infrastructures (e.g. toll gate [17]) are fitted with transponders and sensors which collects the data and offloads it to the server at the edge of the network. In this paper we have proposed the Token based predictive offloading scheme which offloads the data to the RSU equipped with MEC server with token based approach.

A. Motivation

The applications used in the vehicles need computational resources to compute the task. Vehicles have limited resources which are reserved for computing crucial applications so we need additional resources at the edge of the network which must be reliable to provide the services. The major challenge comes when we have to offload the task which have high computational requirements, for example, Augmented Reality can provide useful data and safety warnings through Heads Up Display(HUD) [18] in vehicles and enhance field of view [19], these requires high computation so it can't be computed in vehicle resource due to limitations of resource in vehicles. Other applications which are computationally intensive are speech recognition, natural language processing which provides assistance in helping drivers and passengers. With the emerging of the applications used in the vehicles, passengers focus on other activities also apart from journey such as entertainment, cloud based video games, smart navigation.

B. Related Work

There are various framework proposed in edge computing with the aim of optimizing offloading cost and increasing the computational capabilities. A framework named Autonomous Vehicular Edge (AVE) [20] is proposed for edge computing and efficient job caching is used to schedule the jobs based on the information received in nearby vehicles. Ant Colony Optimisation (ACO) is proposed which is a scheduling algorithm to control the job assignment problem.The major concern in this framework is the convergence rate. In [21]

TABLE I
MAIN NOTATION DEFINITION

Symbols	Definition
T_{vs}	MEC server to vehicle propagation delay and vice versa.
T_{ss}	MEC server to MEC server propagation delay.
T_{quant}	quantum time of round robin scheduling.
T_{comp}	time taken by MEC server to compute the task.
T_{vv}	time taken to send the data from one vehicle to another vehicle.

Adaptive Learning based Task Offloading (ALTO) algorithm is proposed to reduce the average offloading delay and this algorithm is based on multi-armed Bandit Theory . In [22] The architecture for vehicular fog computing is designed which can use the idle resources of other vehicles nearby for the computation of task. But we need a rich computing resources to process the crucial task e.g. vehicle control, accident prevention so relying on these nearby vehicles resources is not safe. In [23] the author proposed the federated offloading scheme to find the optimal average delay results. Besides, distributed algorithm is proposed to find out the optimal vehicle to vehicle offloading relay routing. A Vehicle-Assisted Data Offloading [24] in Mobile Edge Computing Enabled Vehicular Networks is designed where a problem is formulated for highly intensive computational task and two approaches are proposed to solve the formulated problem 1) graph theory based approach 2) heuristic method.

C. Contribution & Organization

The main contributions of this paper are as follows:

- We propose the token based predictive offloading scheme which offloads the data to the server ahead in their running direction.
- DSRC technology facilitates the short range communication between the vehicles to vehicles and vehicles to infrastructure. For the purpose of this we have used the transponders in vehicles and infrastructure. .
- This model is based on MEC(Mobile Edge Computing) with a token(countable quantity which shows commutative property of the server like propagation delay, processing delay, memory consumption.)

The rest of this paper is organized as follows. Section II presents the proposed Token based Predictive Offloading Scheme. Offloading without using predictive offloading scheme and offloading with token based predictive offloading scheme are discussed in subsection II-A and subsection II-B, respectively. The performance evaluation results are presented in section III. Finally, the conclusions are drawn in section IV.

II. PROPOSED TOKEN BASED PREDICTIVE OFFLOADING SCHEME

Fig. 1 shows the architecture of the Mobile Edge Computing in vehicular networks. We have assumed that vehicles are moving in same direction in unidirectional path. There are Road Side Unit (RSUs) along the road and each RSU is equipped with MEC server. RSU provide the wireless service communication with the vehicles present inside the coverage range of RSU. The coverage range of RSU is shown by circle. The transmission range of RSU is upto L/2. The vehicles moving in transmission range of RSU can access only that RSU (MEC server also). RSUs communicate with each other through wireless backhaul. MEC server is equipped with each RSU and compute the task received from RSU with which it connects. All the vehicles move at constant speed on the road.

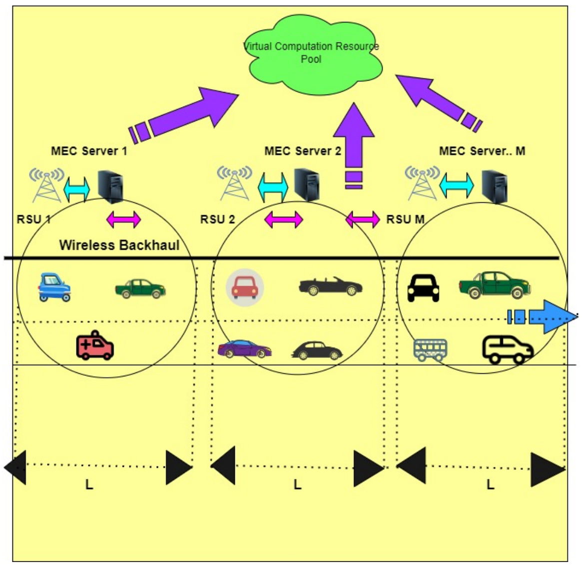

Fig. 1. mobile-edge cloud-enabled vehicular networks.

A. Offloading without using predictive offloading

The vehicles in the highway offload the task to the MEC server through V2I communication mode. The computation of the task may take relatively longer time. Considering the vehicle runs in a highway with high speed, the vehicle may pass through various RSU (equipped with MEC server) during task execution time period. Therefore the transmission of the computed task output to the vehicle needs the transfer of computed output from MEC server 1 which accomplished the task execution to the MEC server n(fig 2) which vehicle newly access. Wireless Backhaul is used in communication between the RSU's, the transmission of computed output from MEC server 1 to MEC server n takes wireless multihop relay between various RSU. Wireless Backhaul transmits at a very low rate due to interference between wireless links. The average delay and transmission cost is increased due to the multihops relay transmission.

B. Offloading with predictive offloading

In this scheme, the vehicle k in MEC server 1 (Fig. 2) send the task to the MEC server n in its running direction. This can be done by keep sending the task from vehicle to vehicle till the final vehicle present in the coverage range of server n. Once the task is computed in server n, by then the vehicle k arrives in the range of server n(vehicle k in server n is shown as vehicle k') and the computed output of the task

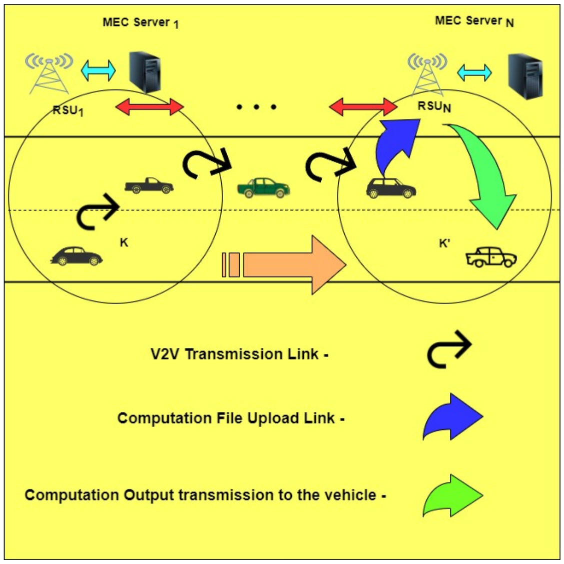

Fig. 2. predictive offloading approach

is send from the server n to the vehicle k through the V2I communication mode. Under this scheme there is no wireless backhaul relay, hence the transmission time and cost can be saved.

C. Token based predictive offloading approach

The development of the recommendation scheme comes under the current availability of the channels in the IOT environment . This scheme is token based predictive offloading scheme which are described as follows :-

- Model is based on the MEC (Mobile Edge Computing) with Token (Countable quantity which shows the cumulative properties of the server like propagation delay , processing delay , memory consumption) policy .
- In Fig. 3 Circular part shows the range of the MEC server and vehicles are the numerical numbers that are mentioned using squares. We are assuming that every vehicle is moving in the same direction , the vehicle will get the available token whenever it is in the range of the MEC server .
- In this Model (Fig. 2) a vehicle k enters the MEC server 1 and request for the token from the MEC server 1. The MEC server 1 checks if the computed result can be send back to the vehicle k before the vehicle k leaves the range of server 1 then the MEC server grants the token to the vehicle k otherwise decline the request.
- If in case the request is declined then vehicle k send the task to the server n in their running direction by sending the task from vehicle to vehicle communication, the vehicles keep on sending the task to other vehicles till the last vehicle present in the range of the next MEC server n (as shown in (Fig. 2).
- The Multi Level Feedback Queue (MLFQ) is implemented in MEC server that includes queue 1 and queue 2 . The MEC server process the task in queue 1 in round robin manner. Queue 1 contains the task of vehicles which are currently in the range of MEC server n and having a token. Queue 2 consist of the task of vehicles which are not in the range of MEC server n but want to offload the data to it.
- On every token grant , MEC server decrements its token count and increments token count after the vehicle comes out of the range of the MEC server and the status of consumed token and available token is stored into the MEC server table (Fig. 4).
- Every vehicle also creates its own table (Fig. 5). If any vehicle is not in the range of any MEC server then in that case the vehicle will check its own table which shows the status (i.e which vehicle is in which server's range) of the other vehicles .

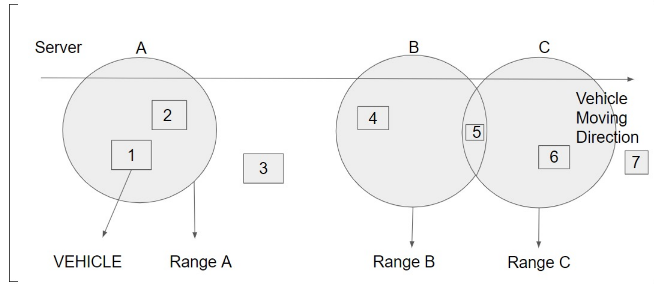

Fig. 3. Representation of server coverage range and vehicles

D. MEC Server Table creation

SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
A	1,2	NILL

SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
B	4	TWO

SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
C	5,6	TWO

Fig. 4. MEC server table

- These are the tables of server A , server B , server C . Vehicle 1 and vehicle 2 (shown in Fig. 3) are in the range of server A and after consuming the token , Server A has no more available token (assuming server A has only two tokens initially) and it fills available token column as NILL(token availability status is shown in Fig. 4) .
- Now any token request arrives to server A , it will deny that request because of no token availability but later on if any vehicle comes out of its range then it will increment its token and grant the pending request .
- Similarly for B and C servers , these servers also change their token status as per the vehicles coming into the range like here initially B server has three tokens and after accepting the request from 4th vehicle it will change the token count to two .

- Server C had four tokens and after accepting the token request from vehicle 5 and vehicle 6 , it changed the token count as two (as shown in Fig. 4).
- We have a special case if any vehicle is in the common range of servers then vehicle will send the request to the server which is in the moving direction of that vehicle like here vehicle 5 is in common range of server B and server C and vehicle 5 will send the request to server C because it is moving towards server C.
- When vehicles are moving in the random directions so in this case , those vehicles which wants to offload the data ask for the token requests to servers which are in its moving direction and if any vehicle is in common range then similarly it will send the request to the server which is in the moving direction of the vehicle.

E. Vehicle Table creation

	1	2	3	4	5	6	7
1	—	A	-1	-1	-1	-1	-1
2	A	—	-1	-1	-1	-1	-1
3	+1	+1	—	-1	-1	-1	-1
4	B	B	B	—	B	-1	-1
5	B,C	B,C	B,C	B,C	—	C	-1
6	C	C	C	C	C	—	-1
7	+1	+1	+1	+1	+1	+1	—

Fig. 5. vehicle table creation

- In Fig. 3 there are seven vehicles vehicle 1 to vehicle 7 and every vehicle storing the status of another vehicle for vehicle to vehicle communication e.g. in vehicle 3 column of vehicle table (Fig. 5), vehicle 1 is behind (-1 entry shows that the vehicle is behind) so offloading data to it is not worthy in this case, vehicle 2 is also behind(as -1 entry is there), vehicle 3 is not storing anything about itself(i.e blank), vehicle 4 is present inside server B, vehicle 5 is present inside server B and server C, vehicle 6 is present inside server C, vehicle 7 entry shows +1 (Here +1 entry shows that the vehicle 7 is currently not in any server range but is in the forward direction so vehicle 3 can offload the data to the vehicle 7).
- If in case like 3rd vehicle (Fig. 3) is not in range of any MEC server and it wants to offload its data but because of no token accessibility it firstly sends the data to its most neighbouring vehicle if this neighbouring vehicle is in the range of succeeding server then this neighbouring vehicle offloads the data to the MEC server otherwise neighbouring vehicle send this data to the next neighbouring vehicle and so on .

III. Performance Evaluations

A. Total time consumption in non predictive offloading scheme

In the non predictive scheme , vehicles comes into the range of any MEC server offloads its data without asking anything(tokens) from the MEC server . This will leads to computational load on MEC server and will also enhance the delays in computation. (Round robin scheduling of task using one ready queue policy through MEC server) Initially at time T0 , at server A, vehicle 3 is in range of it and MEC server B has no vehicle at that time (as shown in Fig. 6).

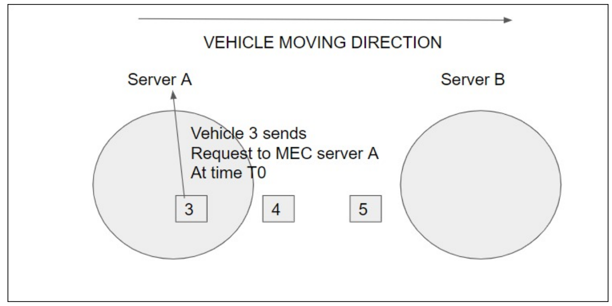

Fig. 6. Non predictive scenerio at time T0

TABLE OF MEC SERVER AT TIME T0

MEC SERVER	VEHICLE
A	3

MEC SERVER	VEHICLE
B	NIL

Fig. 7. MEC server table at time T0

Let's assume propagation delay from vehicle to server and server to vehicle are same and computational delay of task computation in MEC server for every vehicle is same and round robin policy is used. Here we are considering two time instance T_0 and T_t for both server A and server B.

At instance T_0 (Fig. 6):

Server A at time instance T_0 = propagation time $(T_{V3} - T_{S_A})$ + computational time (T_{S_A})

= T_{vs} sec + T_{comp} sec

Server B at time instance $T_0 = 0$ sec (no propagation and computation)

Table of MEC server A and MEC server B at time T0 is shown in Fig. 7.

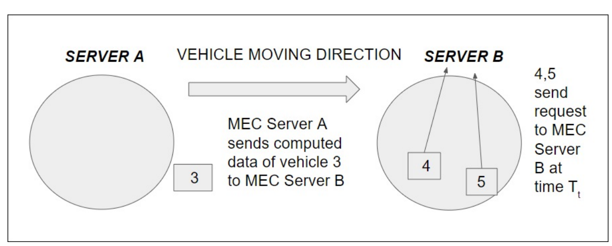

Fig. 8. Non predictive scenario at time Tt

TABLE OF MEC SERVER AT TIME T_t

MEC SERVER	VEHICLE
A	Nil

MEC SERVER	VEHICLE
B	3,4,5

Fig. 9. MEC server table at time Tt

At time instance Tt (Fig. 8), Server A send computed data to Server B, vehicle 4 and vehicle 5 send request to MEC server B.

Here Server A send the data to server B before vehicle 4 and vehicle 5 send their request and hence ready queue is 3 , 4 , 5.(as shown in Fig. 10)

Ready Queue of B			
	3	4	5

Fig. 10. Ready Queue of server B at time Tt

server A at time instance Tt : time taken to send the computed result from server A to server B = T_{ss} sec

server B at time instance Tt : time taken to offload the task of vehicle 4 and vehicle 5 to MEC server B = T_{vs} sec + T_{vs} sec .

Table of MEC server at time instance Tt is shown in fig 9.

At time instance Tt+n : Tt+n is the time when MEC server is computing data inside Queue in Round Robin fashion.

server A at time instance Tt+n : 0 sec

server B at time instance Tt+n : (T_{comp} sec + T_{comp} sec + y*T_{vs} sec) {Round Robin fashion and y=T_{comp}/ T_{quant} }

Total Time consumption in non predictive offloading mode = ($T_{vs} + T_{comp}$){at time T0} + ($T_{ss} + T_{vs} + T_{vs}$) {at time Tt } + ($T_{comp} + T_{comp} + y * T_{vs}$) {at time Tt+n} [overhead of T_{ss} and T_{vs}]

B. Total time consumption in token based predictive offloading mode

In the token based predictive offloading scheme , if any vehicle's data comes into the range of MEC server it will sends the request to the MEC server, if that MEC server accept the request and give token to vehicle then the MEC server will accept the request and process the data as per the round robin fashion till data gets computed , for that every MEC server will maintain its table and queues (which works on the Multi Level Feedback Queue strategy and Round Robin fashion).

Queue 1 :- consist of tasks of those vehicles which have tokens .

Queue 2 :- consist of tasks of those vehicles which are not in the range of any MEC server and don't have tokens but wants to offload their task through vehicle to vehicle communication .

Priority of processing of queue :- Queue 1 > Queue 2.

MEC server process only on the Queue 1. Any vehicle task which are in the Queue 2 will be transferred to Queue 1 when the server has available token at that time .

TABLE OF MEC SERVER AT TIME T_0

MEC SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
A	NIL	TWO

MEC SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
B	NIL	TWO

Fig. 11. MEC server table at time T0

At time T0 (Fig. 12) Vehicle 3 sends token request to MEC server A and it declined its request because MEC server will get an idea that vehicle 3 will leave its range before receiving the output so it will decline the token request of vehicle 3 at time T0 .

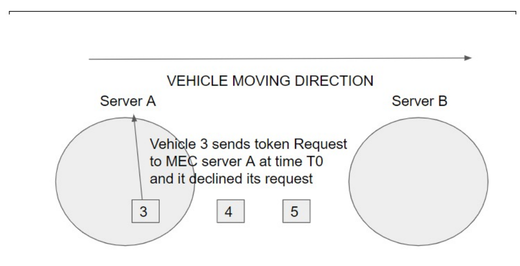

Fig. 12. Token based predictive offloading scenario at time T0

Figure 13 shows the status of ready queue at time T0.

Table of MEC server at time T0 is shown in Fig.11.

At time Tt (as shown in Fig. 14) vehicle 3 send the data to its succeeding vehicles to offload its data on the another next MEC server like here vehicle 3 send the data to vehicle 4 and vehicle 4 will send the request to MEC server B for both its own data and data of vehicle 3 .

Initially MEC server A has three token count and MEC server B has two token count .

The MEC server table at time instance Tt is shown in Fig. 15.

Ready queues of A at T0

Q1 :-	NIL
Q2:-	NIL

Ready queues of B at T0

Q1 :-	NIL
Q2:-	NIL

Fig. 13. Status of ready queues of server A and server B at time T0

At time instance Tt we are considering two scenarios

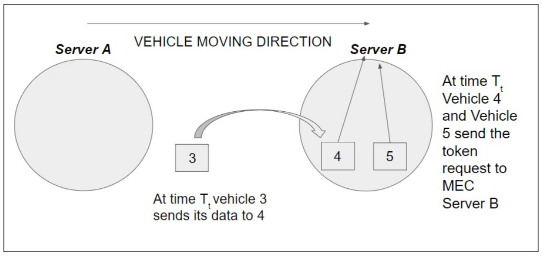

Fig. 14. token based predictive offloading scenerio at time Tt

TABLE OF MEC SERVER AT TIME T_T

MEC SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
A	NIL	TWO

MEC SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
B	4,5	ONE

MEC SERVER	CONSUMED TOKEN	AVAILABLE TOKEN
B	4,5,3	NIL

Fig. 15. MEC server table at time Tt

Case 1:- MEC server B has available token. When data of vehicle 3 is in the range of it so MEC server B will put the data of vehicle 3 into its Queue 1 and compute according to round robin fashion (Fig. 16) .

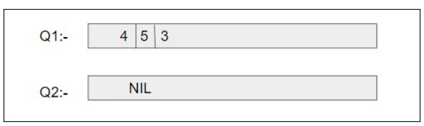

Fig. 16. Status of ready queues in case 1

Case 2:- MEC server B has no available token. When data of vehicle 3 is in range of it so MEC server B put the data of vehicle 3 into Queue 2 (Fig. 17) and firstly it computes the data of those vehicles which are in Queue 1 then later on if any vehicle get its computed data from MEC server B , it will leaves the token and then that token will allot to vehicle 3 and then put the data of vehicle 3 into Queue 1 .

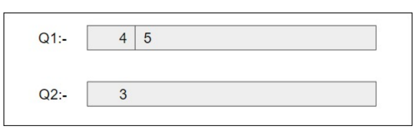

Fig. 17. status of ready queues in case 2

At time Tt (Fig. 14) -

MEC server A :- 0 empty

MEC server B : $(T_{vs} + T_{vs})\{$ for vehicle 4 and 5 $\}$ + $(x * T_{vv})$ $\{$ vehicle 3 send its data through vehicle to vehicle$\}$

(here x is the number of vehicles in between vehicle 3 and MEC Server B)

At time Tt+n $\{$ Tt+n is the time where MEC server is computing data of Queue 1 in Round Robin fashion $\}$

MEC Server A :- 0 $\{$ empty $\}$

MEC server B :- $T_{comp} + T_{comp} + T_{comp}$ for vehicle 4 , 5 and 3 in Round Robin fashion .

Total time :- (T_{vs}) $\{$at time T0$\}$ + ($T_{vs} + T_{vs} + x * T_{vv}$) $\{$at time Tt$\}$ + ($T_{comp} + T_{comp} + T_{comp}$) $\{$at Tt+n $\}$

here because of token based predictive approach we are capable of doing vehicle to vehicle communication and due to this we can remove the cost of MEC server to Server communication and overhead of vehicles using Queues

Now if we compare total time of non predictive scheme with token based predictive scheme

Non predictive scheme :-

$$= (T_{vs} + T_{comp}) + (T_{ss} + T_{vs} + T_{vs}) + (T_{comp} + T_{comp} + y * T_{vs}) \tag{1}$$

Token based predictive scheme :

$$\begin{aligned} &= (T_{vs}) + (T_{vs} + T_{vs} + x * T_{vv}) \\ &+ (T_{comp} + T_{comp} + T_{comp}) \end{aligned} \tag{2}$$

We will get the difference as :-

$$\begin{aligned} &= [(T_{vs} + T_{comp}) + (T_{ss} + T_{vs} + T_{vs}) + (T_{comp} + T_{comp} \\ &+ y * T_{vs})] - [(T_{vs}) + (T_{vs} + T_{vs} + x * T_{vv}) \\ &+ (T_{comp} + T_{comp} + T_{comp})] \\ &= [(3T_{vs}) + (3T_{comp}) + T_{ss} + (y * T_{vs})] - [(3T_{vs}) + (x * T_{vv}) \\ &+ (3T_{comp})] \\ &= [(T_{ss}) + y * (T_{vs})] - [(x * T_{vv})] \\ &= [T_{ss} + y * T_{vs}] - [x * T_{vv}] \\ &= [(T_{ss} + y * T_{vs}) - x * T_{vv}] \end{aligned} \tag{3}$$

[where y=t_{comp} / t_{quant}]

$T_{ss} >>> T_{vv}$ (MEC server to MEC server communication is much greater than vehicle to vehicle communication due to multihop wireless relay between several MEC servers). Also $T_{vs} >> T_{vv}$

= $[T_{ss} + y * T_{vs}] >>>> [x * T_{vv}]$

This results demonstrate that token based predictive offloading reduces the average delay and offloading cost as compared to non predictive offloading scheme.

Non Predictive Offloading scheme(fig 18)-

At initial time instance T0 : As the number of task of vehicles in Queue increases, the time taken to compute task gets increases linearly in non predictive mode i.e y=x

Token based Predictive offloading scheme(Fig. 19)-

At initial time instance T0: As the number of task of vehicles in Queue 1 increases, the time taken to compute task gets increases linearly i.e y=x .

At time instance Tt(In fig 19 Tt=4 sec): Tt is the time instance when Queue 1 is filled and no more token is available. Now incoming task of vehicles will be placed in Queue 2. Queue 1 task will be processed in round robin fashion and task of Queue 2 will be transferred to Queue 1 whenever execution of any task of Queue 1 gets completed. Therefore the time taken in Queue 1 to process the task will be constant after time instance Tt=4 sec i.e y=c (where c is constant).

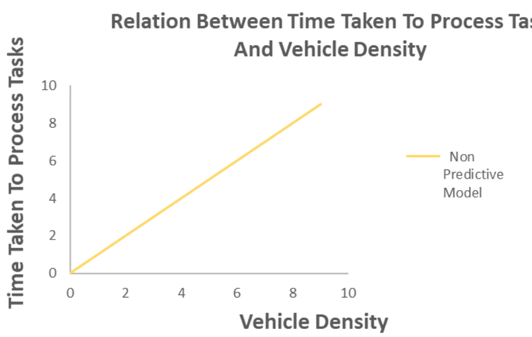

Fig. 18. Relation between time taken to process the task and vehicle density in non predictive offloading mode

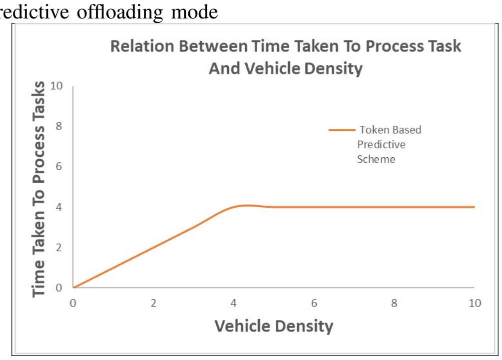

Fig. 19. Relation between time taken to process the task and vehicle density in token based predictive offloading mode

IV. Conclusion

In this paper, a token based predictive offloading scheme is proposed, offloading of task of vehicles depends on the tokens availability at the MEC server. The round robin scheduling algorithm is followed in order to process the task that enters into the queue of MEc server.The separate table for each MEC server is designed to store the information of consumed tokens and available tokens. Also separate table for vehicles is designed so that vehicle can check the status about which vehicles present in which server's range. The proposed scheme uses two queues (queue1 and queue2), the task in queue 2 will enter into queue 1 after any task processing gets over in queue 1. Use of two queues in MEC server enhances the speedup of the execution of task. Based on the framework, we have evaluated the total time consumption in non predictive mode and predictive mode, the results demonstrate that the time consumption in predictive mode is greatly reduced and total offloading cost is also get reduced.

References

[1] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[2] N. Choudhury, R. Matam, M. Mukherjee, and L. Shu, "Beacon Synchronization and Duty-cycling in IEEE 802.15.4 Cluster-tree Networks: A Review," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1765–1788, June 2018.

[3] C. Chen, J. Wu, H. Lin, W. Chen, and Z. Zheng, "A secure and efficient blockchain-based data trading approach for internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 9110–9121, 2019.

[4] H. Vasudev, V. Deshpande, D. Das, and S. K. Das, "A lightweight mutual authentication protocol for v2v communication in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6709–6717, 2020.

[5] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE internet of things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017.

[6] B. Ji, X. Zhang, S. Mumtaz, C. Han, C. Li, H. Wen, and D. Wang, "Survey on the internet of vehicles: Network architectures and applications," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 34–41, 2020.

[7] C. Sonmez, C. Tunca, A. Ozgovde, and C. Ersoy, "Machine learning-based workload orchestrator for vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2239–2251, 2020.

[8] D.-J. Deng, S.-Y. Lien, C.-C. Lin, S.-C. Hung, and W.-B. Chen, "Latency control in software-defined mobile-edge vehicular networking," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 87–93, 2017.

[9] N. Choudhury and M. M. Nasralla, "A proposed resource-aware time-constrained scheduling mechanism for dsme based iov networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, Oct. 2021, pp. 1–7.

[10] N. Choudhury, R. Matam, M. Mukherjee, and J. Lloret, "A beacon and gts scheduling scheme for ieee 802.15. 4 dsme networks," *IEEE Internet of Things Journal*, 2021.

[11] 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.

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[13] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[14] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions," in *Proceedings of the Workshop on Mobile Edge Communications*, 2017, pp. 7–12.

[15] E. Ahmed and M. H. Rehmani, "Mobile edge computing: opportunities, solutions, and challenges," pp. 59–63, 2017.

[16] C. Cseh, "Architecture of the dedicated short-range communications (dsrc) protocol," in *VTC'98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*, vol. 3. IEEE, 1998, pp. 2095–2099.

[17] B. R. Senapati, P. M. Khilar, and N. K. Sabat, "An automated toll gate system using vanet," in *2019 IEEE 1st international conference on energy, systems and information processing (ICESIP)*. IEEE, 2019, pp. 1–5.

[18] H. S. Park, M. W. Park, K. H. Won, K.-H. Kim, and S. K. Jung, "In-vehicle ar-hud system to provide driving-safety information," *ETRI journal*, vol. 35, no. 6, pp. 1038–1047, 2013.

[19] C.-T. Lin, Y.-C. Lin, L.-T. Chen, and Y.-F. Wang, "Enhancing vehicular safety in adverse weather using computer vision analysis," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. IEEE, 2014, pp. 1–7.

[20] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Ave: Autonomous vehicular edge computing framework with aco-based scheduling," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 660–10 675, 2017.

[21] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Transactions on vehicular technology*, vol. 68, no. 4, pp. 3061–3074, 2019.

[22] 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.

[23] H. Wang, X. Li, H. Ji, and H. Zhang, "Federated offloading scheme to minimize latency in mec-enabled vehicular networks," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–6.

[24] J. Huang, Y. Qian, and R. Q. Hu, "A vehicle-assisted data offloading in mobile edge computing enabled vehicular networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.