

LOAN: Latency-Aware Task Offloading in Association-Free Social Fog-IoV Networks

Minu Tiwari, Ilora Maity, and Sudip Misra

Department of Computer Science and Engg., Indian Institute of Technology Kharagpur, 721302, India
Email: minutiwari@iitkgp.ac.in, imaity@iitkgp.ac.in, smisra@cse.iitkgp.ac.in

Abstract—A social fog-IoV network involves time-critical tasks because it is highly dynamic due to rapid changes in the network topology. Therefore, completing tasks within the allowable delay is a challenge for social fog-IoV networks. In this context, we present a latency-aware task offloading scheme, named LOAN, in a fog-enabled association-free social Internet of Vehicle (IoV) network that aims to minimize the delay of time-critical tasks while saving the starvation of best-effort tasks. Our work considers different priorities for the service of time-critical tasks while efficiently utilizing fog resources. Different from the works in the literature that provide privilege to the time-critical tasks, LOAN handles the service of best-effort tasks efficiently without making them suffer conditions such as starvation. LOAN manages the priority levels of the tasks by incrementing their priorities based on their waiting time for the task service. We formulate the problem of efficient task service by suitable fog node as a coalition formulation game. Numerical results show that the LOAN achieves a reduction in delay compared to Greedy method by 36.5%.

Index Terms—Social fog-IoV, Task Offloading, Time-Critical, Coalition Game, Cooperative Service

I. INTRODUCTION

The introduction of fog computing in Internet of Things (IoT) networks has eased the service of the user tasks. However, fog-enabled IoT introduces the challenge of serving all the user tasks fulfilling all the demands due to its own resource limitation. The issue becomes serious in an environment such as the social Internet of Vehicles (IoV) where the vehicle applications have strict QoS requirements and are latency-sensitive [1]. In a social IoV environment, vehicles form dynamic associations and exchange real-time information. Therefore, social IoV has several useful application domains including road safety, vehicle mobility monitoring, analysis of driver behavior, and tracking stolen vehicles. The challenge is to provide the task services to the vehicles during their mobility without any hindrance. With each movement, the association of vehicles with the fog nodes (FNs) changes, and the FNs need to migrate the services. Moreover, the vehicles are capable to process some of their tasks based on their availability of resources. However, they need to offload their tasks with stringent requirements to some other entity. The fog-enabled social IoVs offer the provision of offloading the tasks to FNs such as road side units (RSUs) which can serve the vehicle tasks, while fulfilling the task requirements. Further, in an association-free social IoV networks, the vehicles can obtain the best services without

associating to a particular FN. Nevertheless, to efficiently utilize the resources, the vehicles with idle resources serve the tasks of requesting vehicles [2].

Essentially, it is important to decide which tasks to be served by whom. The tasks can be prioritized according to their requirement to make the decision [3]. However, this prioritization of the tasks leads to the starvation of low-priority tasks. Further, it is possible that a single FN is not able to serve the tasks which require the FNs to cooperatively provide their services [4].

A. Motivation

In an IoV environment, the vehicles serve their own task if they are capable of serving it. However, the onboard units (OBU) on the vehicles do not possess the resources to serve all the tasks. The reason may be the features that the task service requires is not available, the processing power is not enough, or the vehicle is not able to process the task immediately. Thus, the vehicle is required to offload the task to a capable entity. Notably, the vehicles are in a range of many entities that are willing to offer their services to the offloaded task. Thus, the social interaction among the entities allows the vehicles to obtain services from others. However, the vehicle needs to choose a suitable entity that can handle its task fulfilling the user requirements and easing the service. Moreover, the entities offering the service also have limited resources. To solve the issues of who should serve the task, the existing literature [5], [6] discusses the works where the IoT devices are not associated with a particular FN and FNs form a social network to collaboratively decide who should serve the task. The past studies [7], [8] depict that the vehicles can collaboratively serve the tasks communicating with the RSU forming a social IoV network. However, the issue of delay becomes crucial in social IoV networks where sensitive tasks related to various road network aspects such as traffic and safety are requested and served. Moreover, the problem arises when best-effort tasks co-exist with time-critical tasks.

To handle the time-critical tasks, authors in the existing works [9], [10] discuss the offloading and planning in fog networks. These works discuss the prioritizing of the time-critical tasks and cooperatively serving the task. Moreover, the entities serving the tasks have limited resources. Therefore, the entities may fail to fulfill all the task requirements. The social interaction of the FNs and the vehicles allow them to

offer and obtain task services, respectively. But, when a high number of tasks are offloaded to an associated FN, it becomes overloaded with the task request which in turn searches for a suitable entity to handle the task. This scenario increases the delay of the task. Therefore, these issues raise the question of

- How to handle the task offloading in the environment where the best-effort and time-critical tasks co-exist?
- How to select/motivate the FNs to collaborate for serving the vehicle tasks?

B. Contribution

We propose a latency-aware offloading scheme, named LOAN, prioritizing the service of time-sensitive while rescuing the starvation of low-priority tasks. In particular, the contributions of this work are,

- We present a task service framework in a social IoV network considering time-critical and best-effort tasks with a utility optimization problem for FNs considering the task requirements and resource availability at FNs.
- We model the problem of task service as a coalition formation game where the FNs resources are efficiently utilized considering the requirements of the tasks.

II. RELATED WORK

We present the existing works discussing the offloading in fog-enabled IoT networks considering different aspects of task requirements. Liu *et al.* [11] considered a delay-sensitive IoT environment where multiple tasks are handled at the fog layer. They formulated the problem as a non-cooperative game where the tasks compete to get the services of the helper nodes. Considering battery-powered FNs, Bozorgchenani *et al.* [12] proposed a system to stabilize the energy of the FNs by taking the offloading decision based on packet generation time. Ali *et al.* [13] considered the problem of latency incurred in forwarding the computation-intensive tasks to the cloud by the FNs. To solve the problem, they proposed a paradigm where the idle resources of the user are utilized for the service of the task. Similarly, Mukherjee *et al.* [3] proposed a scheduling strategy for deadline-aware user tasks at the FNs. The authors aimed to maximize the number of processed tasks while considering two types of queues at the FNs – high priority and low priority. However, the task delay can be minimized by cooperated service by the FNs since the FNs have limited resources which the aforementioned works did not consider.

The concept of social IoV is introduced in the literature where the authors discuss the social networks formed by the vehicles. Alam *et al.* [7] presented the structure of social IoV and study the relation and interaction of the vehicles in this architecture. A recommendation system based on social IoV network is presented by Muhammad and Zia [14]. The authors considered the concern of information sharing in social IoV architecture for studying the feasibility of recommendation system in such architecture by considering an agent-based model. Kerrache *et al.* [8] discussed the issue of trust among

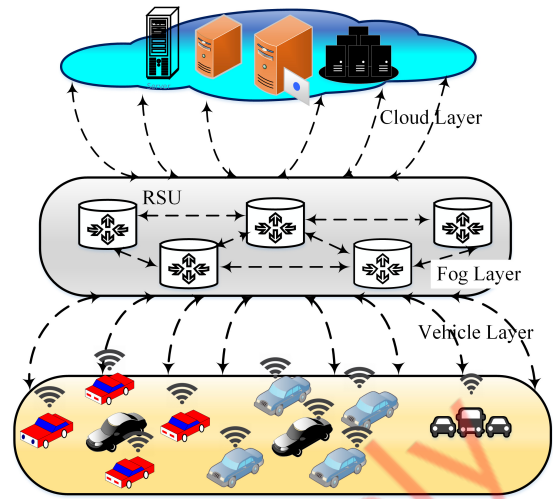


Fig. 1: Social Fog-IoV Architecture

the communicating vehicles and their users in social IoV architecture. The aforementioned works discuss the social IoV network for vehicle interaction but did not consider the task requirements while serving the task in such networks.

Further, few works in the literature [4], [15] focused on the collaborative task service. Chiu *et al.* [15] presented a Fog Radio Access Network (F-RAN) architecture where the master F-RAN node assigns the task to F-RAN nodes which collaboratively serve the task. The authors aim to minimize the latency of task service and utilize the resources of F-RAN nodes. A federated task service approach is presented by Tiwari *et al.* [4] where the authors proposed a fog computing framework focusing on the critical task and providing a collaborative task service by FNs.

Synthesis: The study of the existing works reveals that the authors have considered the time-critical tasks and prioritized their processing. However, time-critical tasks have different priority levels depending on the task type. Therefore, these tasks should be processed accordingly. Moreover, in a social fog-IoV environment where time-critical tasks coexist with best-effort tasks, existing works prioritize only time-critical tasks and do not prevent the starvation of best-effort tasks. Additionally, for association-free social fog-IoV networks, the most important decision is the selection of the FN that serves the request broadcasted by a vehicle. Therefore, in our work, we consider optimal processing of time-critical and best-effort tasks in an association-free social fog-IoV network.

III. SYSTEM MODEL

We consider a social Fog-IoV architecture consisting of vehicles requesting the task services, fog nodes offering the services, and the cloud. Let the set of vehicles be represented as $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$. The vehicles are equipped with computational power in their OBUs. This allows them to provide task services. On the other hand, an FN can be mobile or static and can serve the task requested. Furthermore, the FNs are associated with a roadside unit (RSU) which has the information of FNs. We denote the set of FNs as

$\mathcal{F} = \{f_1, f_2, \dots, f_M\}$. The FNs can communicate with other FN in their vicinity. Let FC_j denote the available processing capacity of $f_j \in \mathcal{F}$.

In this work, we consider two types of tasks – time-critical tasks and best-effort tasks. The time-critical tasks are stringent in their delay requirement. On the other hand, the best-effort tasks do not provide any QoS guarantee. Let the task set be represented as $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_P\}$. Each task τ_i is associated with a maximum allowable task service delay $\delta_{i,max}^{serv}$. Additionally, we consider that each task τ_i is associated with a maximum allowable waiting time ($\delta_{i,max}^{wait}$) before it is assigned to the OBU or an FN. We consider that the maximum allowable waiting time of best-effort tasks is greater than that of time-critical tasks.

We consider that the vehicles are not associated with a particular FN. Therefore, the vehicles broadcast a task service request when the corresponding OBU is unable to process the task. The request comprises of the task parameters, viz., task priority, and task size. The priority of a task signifies whether it is a time-critical task or a best-effort task. For example, a highly time-critical task has the highest priority and a best-effort task has the lowest priority. Let the request for task τ_k is denoted as $R_k = (t_k^{prio}, s_k, TC_k)$, where t_k^{prio} defines the priority of the task, s_k is the size of the task that is to be served, and TC_k is the processing capacity required for the task. This request is received by all the FNs in its vicinity. Subsequently, the FNs need to decide who will serve the task based on the requirement of the task.

A. Delay Model

The delay incurred in the service of any task i includes the delay in uploading the task, the propagation delay, the queuing delay, the waiting time, the processing delay, and the downloading delay. The upload delay is incurred when the task of size s_i is transmitted to the FN which has agreed to serve the task. It is formulated as $\delta_i^{up} = \frac{s_i}{r_j}$, where r_j is transmission rate. The propagation delay δ_i^{pr} depends on the distance from the vehicle to the FN. At each FN, the selected tasks for serving can be considered as the Poisson process [16]. Therefore, the delay induced due to the tasks queued at the FNs is formulated as an $M/M/1$ model, $\delta_i^{qu} = \frac{1}{\Lambda_j - \psi_j}$, where Λ_j is the task arrival rate and ψ_j is the service rate of the FN j . When the task is served at the FN, the execution delay is incurred during the processing of the task which depends on the amount of computation c_i of the task and the CPU frequency μ_j of the FN. Mathematically, $\delta_i^{exe} = \frac{c_i}{\mu_j}$. Moreover, every task is associated with a fixed allowable waiting time. Notably, when the task is broadcasted by the vehicle, the FNs collaboratively decide who will serve the task. This decision takes some time. Therefore, we define the waiting time as the interval between the generation of the task and the time when the task is allocated to a fog queue. This can be written as, $\delta_i^{wait} = \delta_i^{allo} - \delta_i^{gen}$, where δ_i^{allo} and δ_i^{gen} are the task allocation and generation time, respectively. Therefore, the estimated value of the total delay incurred in the service of the task is expressed as $\delta_i^{tot} =$

$$\delta_i^{up} + \delta_i^{pr} + \delta_i^{wait} + \delta_i^{qu} + \delta_i^{exe} + \delta_i^{dow}.$$

B. Problem Description

When the vehicles need a task service and their OBU is not capable to serve the task, they broadcast a task request. The task request is received by all the FNs in its vicinity. The FNs that receive the task request collaboratively decide who will serve the task. The decision of task service is accepted by the FNs who have enough resources to serve the task according to the task requirement. Therefore, the problem is to determine the target FN for task offloading for the scenario where time-critical tasks of heterogeneous priority levels and best-effort tasks co-exist.

To solve the problem of how the FN should collaborate to serve any task while fulfilling the task requirements, we formulate the problem as a coalition formation game [17]. A coalition formation game involves multiple rational players who cooperatively form coalitions. Each player decides which coalition to join based on payoff or utility. A coalition structure is a set of coalitions that includes all players. The solution of the coalition formation game is a stable coalition structure that maximizes the utility of the players [18].

The coalition formation game ensures that the coalition of tasks formed can be optimally served by the FNs and the FNs collaborating in the service meet the requirements of the tasks. The tasks are the players of the coalition game. Each coalition is served by a designated FN. We denote the coalition structure as $\mathcal{Z} = \{\zeta_1, \zeta_2, \dots, \zeta_c\}$ where each ζ_k denote a coalition. The coalition formation algorithm continues until the coalitions are stable. Once the coalitions are stable, the respective coalitions are executed by the corresponding FNs. Let A_l denote the FN associated with coalition $\zeta_l \in \mathcal{Z}$.

Notably, the aim is to maximize the utility of the coalition by meeting the task requirements. The coalitions are formed according to the task requirement and the tasks are queued by the FNs. Further, as mentioned, each task has a maximum allowable waiting time before which the task should be assigned to a coalition. The coalitions are formed in a way that the FNs satisfy the priority of the task. The FNs that are assigned the coalition should finish the task service within the maximum allowable delay of all the tasks in the coalition. Therefore, the utility of the coalition depends on the efficient service of the task fulfilling the task requirements. Mathematically, the utility of coalition ζ_l is defined as,

$$U_l = \sum_{\tau_i \in \mathcal{T}_l} \left(\frac{\delta_{i,max}^{serv} - \delta_i^{tot}}{\delta_{i,max}^{serv}} + \frac{\delta_{i,max}^{wait} - \delta_i^{wait}}{\delta_{i,max}^{wait}} \right) t_i^{prio}, \quad (1)$$

where \mathcal{T}_l is the set of tasks served by A_l .

C. Objective

The objective of the task coalitions is to get efficient task service by utilizing the FN resources that would maximize their utility. The decision of which tasks should form a coalition according to the availability of FNs and the requirements of the tasks depends on the utility achieved. Therefore, the

objective is to maximize the utility of the coalitions which in turn maximizes the utility of the tasks. The objective is defined as,

$$\max(\mathcal{U}_{\uparrow}) \quad \forall \zeta_l \in \mathcal{Z} \quad (2)$$

subject to

$$\begin{aligned} \delta_i^{tot} &\leq \delta_{i,max}^{serv} & \forall \tau_i \in \mathcal{T}_l, \\ \delta_i^{wait} &\leq \delta_{i,max}^{wait} & \forall \tau_i \in \mathcal{T}_l, \\ \sum_{\tau_i \in \mathcal{T}_l} TC_i &\leq FC_j, \text{ where } f_j = A_l \end{aligned}$$

The first constraint ensures that the task service delay should not exceed the maximum allowable service delay. The second constraint is to assure that the waiting time of the task does not surpass the maximum waiting time of a particular task. The third constraint expresses that the selected FN should have sufficient capacity to serve all tasks in the coalition.

IV. PROPOSED SOLUTION

We present the solution approach, named LOAN, adopted to solve the coalition formation game where the players are the tasks. When the vehicle j broadcast the task request R_j , it is received by the FNs in its vicinity. According to the service requirement of the task, an associated RSU takes the decision of coalition formation. We consider that the FNs should remain in the vicinity of the vehicle for which they are serving the task. The coalitions of the tasks are formed and it is assigned to the best FN capable of serving the tasks of the coalition. Algorithm 1 shows the coalition formation process.

The coalitions are formed in a way that the utility is maximized while efficiently utilizing FN resources. Also, we present a hedonic coalition game where the players themselves decide their preference of joining a coalition. However, the preference of coalition is decided by the preference rule.

Definition 1 (Preference Rule). *The order in which a coalition structure of a set is preferred over the other coalition structure is defined by the Preference rule. The relation $\mathcal{Z}_1 \prec \mathcal{Z}_2$ denotes that the partitioning \mathcal{Z}_1 of the set \mathcal{T} is preferred over \mathcal{Z}_2 partitioning, where \mathcal{Z}_1 and \mathcal{Z}_2 are the two partitions of the set \mathcal{T} .*

The players decide which coalition to join based on the utility value. A coalition can merge with another coalition based on the merge rule.

Definition 2 (Merge Rule). *The coalition set represented as $\{\zeta_1, \zeta_2, \dots, \zeta_B\}$ can be merged into a single coalition denoted as $Q = \bigcup_{\beta=1}^B \zeta_\beta$ when $Q \prec \{\zeta_1, \zeta_2, \dots, \zeta_B\}$.*

Similarly, the players of a coalition may prefer to split from the current coalition and form another. However, the splitting from one coalition is governed by the split rule.

Definition 3 (Split Rule). *According to the split rule, the coalition $Q = \bigcup_{\beta=1}^B \zeta_\beta$ can be split into smaller coalitions $\{\zeta_1, \zeta_2, \dots, \zeta_B\}$ if $\{\zeta_1, \zeta_2, \dots, \zeta_B\} \prec Q$ by the players of the coalition Q .*

Algorithm 1: Coalition Formation Game

```

1 Inputs:  $\mathcal{F}, \mathcal{T}$ .
2 Output:  $\mathcal{Z}$ .
3 for  $f_i \in \mathcal{F}$  do
4   Create empty coalition  $\zeta_l$ , where  $A_l = f_i$ 
5   Update  $\mathcal{Z}$ 
6 while Merge and Split do not converge do
7   Coalitions in  $\mathcal{Z}$  attempt to merge based on
      Definition 2
8   Coalitions in  $\mathcal{Z}$  attempt to split based on Definition 3
9 Update  $\mathcal{Z}$  by removing empty coalitions return  $\mathcal{Z}$ 

```

A. Prioritized Task Service

In this section, we present the proposed prioritized task service algorithm for LOAN. Algorithm 2 shows the steps for prioritized task service. The vehicles which are unable to process their own tasks broadcast task requests to all the FNs in their vicinity. The FNs collaboratively decide the willingness to serve the tasks and informs the same to the RSU. RSU then executes the coalition formation game in which the coalitions are formed for the tasks with similar QoS requirements. As discussed, we consider the time-critical tasks and the best-effort tasks from the vehicles. Also, from the information received from the FN, RSU checks the mobility range of the vehicles in the vicinity of the requesting vehicles. According to the resource possession of the FNs, the coalitions of the tasks are assigned to them. Corresponding RSU saves the details about ongoing tasks. In each round, a vehicle can resend a task service request if the previous request is incomplete. We update the task priorities for each new request. If the waiting time $\delta_{i,max}^{wait}$ of a particular task τ_i in the coalition ζ_k assigned to the FN f_j is over by a certain percentage q , the priority level of the task is increased by one level. Nevertheless, the time-critical tasks have different priority levels based on which they form the coalitions and the FNs are assigned. However, the best-effort tasks have the lowest priority level. The increment in the priority level ensures that the best-effort tasks do not suffer starvation. However, we restrict the number of times a task priority can be increased to \mathcal{M} to ensure fairness in task processing. Therefore, the priority of a task cannot be increased further when it has already increased its priority for \mathcal{M} times. Thereafter, the task is offloaded to the cloud.

V. PERFORMANCE EVALUATION

We analyze the performance of the proposed scheme in this section. We consider an area of $1000 \times 1000 m^2$ where the vehicles are randomly deployed in the network. The RSU

Algorithm 2: Prioritized Task Service

```

1 Input:  $\mathcal{V}$ .
2 Output:  $\mathcal{G}$ .
3 Initialize  $\mathcal{G} \leftarrow \emptyset$ .
4 for  $v_k \in \mathcal{V}$  do
5   if  $v_k$  is willing to offload a task then
6     if  $\delta_k^{wait} \geq \frac{p_{k,max}^{d_{wait}}}{100}$  then
7       if The number of times when the task
           priority was increased is less than  $\mathcal{M}$  then
8          $t_k^{prio} \leftarrow t_k^{prio} + 1$ 
9         Broadcast task request  $\mathcal{R}_k$ 
10        else
11          Offload the task  $\tau_k$  to the cloud
12 Form the coalition structure  $\mathcal{Z}$  using Algorithm 1 for
     $\zeta_l \in \mathcal{Z}$  do
13   for  $v_k \in \mathcal{V}$  do
14     if  $v_k$  broadcasted a request and  $\tau_k \in \zeta_l$  then
15       Task  $\tau_k$  is added to the service queue of FN
            $A_t$ 
16       Update  $\delta_i^{wait}$ 
17    $\mathcal{G} \leftarrow \mathcal{G} \cup \{\tau_k\}$ 
18 return  $\mathcal{G}$ 

```

are deployed in a Barabasi-Albert topology [19] which is a scale-free topology ensuring that the system performance is compliant with the change in the network size. The vehicles move in the network following the Gaussian Markov mobility. We discuss the simulation settings in Table I. Further, we plot the results using the confidence interval of 95%.

TABLE I: Simulation Parameters

Parameters	Values
Number of RSUs	30-100
Number of tasks	100-500
Network topology	Barabasi-Albert [19]
Mobility model	Gauss Markov
CPU frequency (vehicle)	1-3 GHz
CPU frequency (FN)	2.9-4.2 GHz [20]
Task computation amount	1500 and 2500 Megacycle [20]
Task size	450 KB [20]
FN transmission power	20 dBm [20]
Bandwidth	20 MHz [20]

Moreover, to assess the efficiency of LOAN, the results are compared with heuristic-based offloading (HOV) [2] where the vehicles for offloading are chosen based on the heuristic method. HOV was selected to show the consideration of priority levels of the tasks for executing and offloading in an IoV environment. Moreover, we took two other traditional methods for comparison, namely, Greedy and random offloading (ROV). Greedy method chooses the nearest FN for offloading the task and ROV chooses an FN randomly for offloading. The two methods highlight the reason for using a

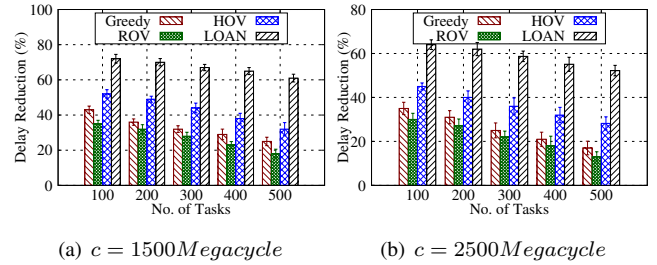


Fig. 2: Delay Reduction for Time-Critical Tasks

proper algorithm for the service of the task.

A. Results

1) *Delay Reduction:* We analyze the reduction in delay of the time-critical tasks using LOAN by comparing it with the baselines. Fig. 2 shows the comparison of delay reduction using all the methods for different computation amounts of the tasks. It is evident from the figure that the delay reduction decreases with the increase in the number of the tasks. It is obvious as more number of tasks are offloaded to FNs with the increase in number of task which increases the queuing delay of the tasks. Further, ROV does not perform well as compared to the HOV and proposed scheme. Since ROV does not consider the QoS requirement of the tasks and randomly chooses the FN without knowing their resource availability. Similarly Greedy does not improve the delay reduction due to its greedy nature of FN selection without considering the task requirement for offloading. On the contrary, HOV suffers the delay of considering a task service at a time by selecting an FN without considering the different priority levels of the tasks. LOAN outperforms all the baseline schemes by considering the task requirement, resource availability of FNs, and priority levels of the tasks.

The delay reduction is also compared for the best-effort tasks in Fig. 3. Although, the best-effort tasks do not have any QoS requirement but suffer starvation without using the proper method. ROV cannot improve the delay reduction of best-effort tasks by considering all the tasks of similar priority. Greedy, in the same way, adopt the greedy method for all the tasks for offloading it to FN, and thus cannot improve the delay. In contrast, HOV prioritizes the time-critical tasks and thus, the delay of the best-effort tasks is not reduced. LOAN performs the best in delay reduction of the best-effort tasks by considering the priority levels of the tasks.

2) *QoS Violation:* We also evaluated the performance of the proposed scheme for the violation of QoS of the tasks. The comparison of the QoS violation of the schemes is shown in Fig. 4. The figure shows that the percentage of QoS violation increases when the number of tasks increases. LOAN performs the best whereas ROV performs the worst. ROV violates the QoS requirement of most of the tasks due to its random selection of the FN. Likewise, Greedy due to its selection of the nearest FN without considering their resources violates the QoS requirement. In contrast, HOV maintains the QoS requirement for some of the tasks but

due to its non-consideration of the priority levels suffers the QoS violation. However, LOAN, due to its consideration of different priority levels and QoS requirement of tasks before offloading achieves high QoS.

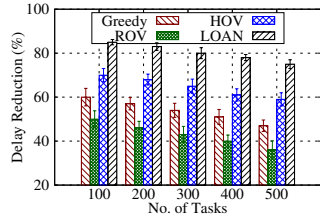


Fig. 3: Delay Reduction of Best-Effort Tasks

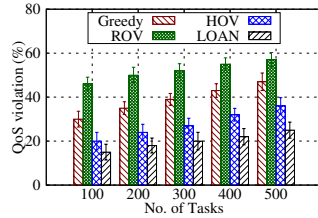


Fig. 4: QoS Violation of Time-Critical Tasks

VI. CONCLUSION

In this paper, we presented a latency-aware task offloading framework for association-free social fog-IoV networks, named LOAN, while considering the co-existence of time-critical and best-effort tasks. LOAN aims to minimize the latency of time-critical tasks while saving the starvation of best-effort tasks. We considered different priority levels of the tasks which privileged the service of tasks with more strict delay-requirement. To efficiently utilize the resources of vehicles, we consider that the vehicles are capable of serving their tasks and handling some of the offloaded tasks. We modeled the problem of task service with similar QoS requirements as a coalition game which allows efficient cooperation of FNs to serve the task. This allows the FNs to cooperatively decide and serve the task service in an association-free environment where the vehicles are not associated with a particular FN. The results depicted that LOAN reduces the delay of the time-critical tasks by 27%, 36.5%, and 44% compared to HOV, Greedy, and Random schemes.

In a social fog-IoV environment, vehicles can directly communicate with other vehicles to request task service. Therefore, we plan to design a recommendation system for the selection of static or mobile fog nodes as a future extension of LOAN.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the funding received from the INAE Abdul Kalam Technology Innovation National Fellowship, Sanction No. INAE/121/AKF, Dt. 13-02-2019.

REFERENCES

- [1] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1159–1166, 2019.
- [2] J. Huang, Y. Qian, and R. Q. Hu, "A Vehicle-Assisted Data Offloading in Mobile Edge Computing Enabled Vehicular Networks," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [3] M. Mukherjee, M. Guo, J. Lloret, R. Iqbal, and Q. Zhang, "Deadline-aware fair scheduling for offloaded tasks in fog computing with inter-fog dependency," *IEEE Communications Letters*, vol. 24, no. 2, pp. 307–311, 2019.

- [4] M. Tiwari, S. Misra, P. K. Bishoyi, and L. T. Yang, "Devote: Criticality-aware federated service provisioning in fog-based iot environments," *IEEE Internet of Things Journal*, 2021.
- [5] F. S. Abkenar, Y. Zeng, and A. Jamalipour, "Energy consumption tradeoff for association-free fog-iot," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
- [6] F. S. Abkenar and A. Jamalipour, "Energy optimization in association-free fog-iot networks," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 404–412, 2019.
- [7] K. M. Alam, M. Saini, and A. El Saddik, "Toward social internet of vehicles: Concept, architecture, and applications," *IEEE access*, vol. 3, pp. 343–357, 2015.
- [8] C. A. Kerrache, N. Lagraa, R. Hussain, S. H. Ahmed, A. Benslimane, C. T. Calafate, J.-C. Cano, and A. M. Vegni, "Tacashi: Trust-aware communication architecture for social internet of vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5870–5877, 2018.
- [9] C. Liu, K. Liu, S. Guo, R. Xie, V. C. Lee, and S. H. Son, "Adaptive offloading for time-critical tasks in heterogeneous internet of vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7999–8011, 2020.
- [10] U. K. Saba, S. ul Islam, H. Ijaz, J. J. Rodrigues, A. Gani, and K. Munir, "Planning fog networks for time-critical iot requests," *Computer Communications*, 2021.
- [11] Z. Liu, X. Yang, K. Wang, Y. Yang, and Z. Shao, "Computation offloading game for multi-task multi-helper fog networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [12] A. Bozorghchenani, D. Tarchi, and G. E. Corazza, "Computation offloading decision bounds in swipt-based fog networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [13] B. Ali, M. A. Pasha, S. ul Islam, H. Song, and R. Buyya, "A volunteer-supported fog computing environment for delay-sensitive iot applications," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3822–3830, 2020.
- [14] A. Muhammad and K. Zia, "Feasibility of recommendation system mediated by social internet of vehicles," in *International Symposium on Consumer Technologies (ISCT)*, 2019, pp. 41–46.
- [15] T.-C. Chiu, A.-C. Pang, W.-H. Chung, and J. Zhang, "Latency-driven fog cooperation approach in fog radio access networks," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 698–711, 2018.
- [16] I. Maity, R. Dhiman, and S. Misra, "Mobiplace: Mobility-aware controller placement in software-defined vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 957–966, 2021.
- [17] W. Saad, Z. Han, and A. Hjørungnes, "Coalitional games for cooperative cellular wireless networks," in *Cooperative Cellular Wireless Networks*, 2011, pp. 347–381.
- [18] D. Niyato, P. Wang, W. Saad, and A. Hjørungnes, "Coalition Formation Games for Bandwidth Sharing in Vehicle-To-Roadside Communications," in *IEEE Wireless Communication and Networking Conference*, 2010, pp. 1–5.
- [19] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [20] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2018.