Robust Resource Allocation for Delay Sensitive Tasks in V2I Networks

Razin Farhan Hussain
{*razinfarhan.hussain*1}@*louisiana.edu*
The School of Computing and Informatics, at University of Louisiana Lafayette
Anna Kovalenko
{*alisenperson*}@*gmail.com*
The School of Computing and Informatics, at University of Louisiana Lafayette
Mohsen Amini Salehi *Member, IEEE*
{*amini*}@*louisiana.edu*
Assistant Professor at the School of Computing and Informatics, at University of Louisiana Lafayette
Omid Semiari
{*osemiari*}@*georgiasouthern.edu*
Assistant Professor at Electrical and Computer Engineering Department, at Georgia Southern University

***Abstract*—The development of vehicular network leveraged the Vehicle to Infrastructure(V2I) communication in roads and highways which provides different services (e.g., wrong way driver warning, weather information, traffic information) to moving vehicles. In V2I system majority of the services are delay sensitive which need to be served quickly. In this case delay is one of the main constraint to be addressed. So in this paper we identified end-to-end delay while servicing a task in V2I system. There is a certain period of time after which a task has no value of execution. Conventional cloud structure can not serve the purpose as it introduce more delay because of cloud server which is located in distant place. That is why edge computing concept needs to be implement which brings computational power to edge of network to support the delay sensitive tasks. In V2I road side units (Base Station) are edge devices which in real time scenario has different computational power and communication medium(e.g., wifi, wired) as well. This heterogeneity across base stations and oversubscribe situations like peak time of a day and emergency situations(e.g, accident ahead, road block) can cause erratic quality of service (QoS) in V2I system. So in this research work we define a robust V2I system to minimize the end to end delay without compromising QoS in oversubscribe situations. We considered worst case scenario or oversubscribe situations which denotes the system is really busy and sometimes can not meet the deadline of receiving task. So our main goal in this work is how to allocate the tasks to the Base Stations so that the tasks deadline missing rate can be minimized. As the Base Stations have connectivity with other Base Stations, so this concept is used to transfer/allocate the task to other Base Stations if task's deadline can not be met within requested Base Station. Vehicles mobility in the road is also considered while transferring task and provided efficient scheduling technique for executing the delay sensitive tasks in Base Stations.**

***Index Terms*—Road Side Unit, Vehucular Network, End-to-End delay, V2I, V2X.**

I. Introduction

With the advancement in the communication technology and high speed data transfer through the network link many future applications are emerging in Vehicular Network. Vehicles are getting smarter everyday with different sensors and communication units in it, which enables the vehicles to communicate with other vehicles and network infrastructures. The scenario of vehicular communication is termed as V2X which is becoming popular day by day. V2X can be defined as Vehicle to everything communication that consists of two types. They are Vehicle to Vehicle(V2V) and Vehicle to Infrastructure(Road Side Unit)(V2I) . Road Side Units(Base Station) can be assumed as stationary devices that is installed in road side which has computational power as well as communication capability [1].

The field of V2X, specially V2I is getting lots of leverage from organizations like Federal Communications Commission(FCC) that reserved the 5.850 to 5.925 GHz frequency band for V2X. The United States Department of Transportation (USDOT or DOT) defines V2I communication as next generation of Intelligent Transportation Systems (ITS). Through this V2I communication system, many different kind of services(for example Wrong Way Driver Warning, Cooperative Forward Collision Warning, Lane-change Warning, Weather Information, Traffic Information: Traffic Jam Ahead Warning to base station) can be provided to the moving vehicles in roads and highways. The moving vehicles can request the Base Stations for many services and receive response from the Base Stations accordingly. The majority of the services requested by the vehicles are delay sensitive which means they need real time task processing in Base Stations. For performing this process efficiently, end to end delay needs to be considered as it is one of the main constraints in delay sensitive task processing. End to end delay in V2I communication can be defined as accumulation of three delay components. They are - Uplink delay, task processing delay in Base Stations and Downlink delay. One of the main contributions in this paper is to define robustness of V2I service to satisfy the Quality of Service(QoS) to V2I users.

Thinking of implementing the traditional cloud architecture where the cloud server is located in a distant place is not

preferable in this case because of increased network delay. The heterogeneity of Base Stations also need to be considered. In V2I, heterogeneity is represented in terms of computational power and communication medium of Base Stations to the core network (e.g., wireless, fiber optic). Furthermore in emergency situations (e.g., disaster, accidents or peak time of day) there is a surge of requests from vehicles to Base Stations which is termed as oversubscribed situation. So the heterogeneity and oversubscription can cause erratic quality of service (in terms of end to end delay) for V2I users which is not acceptable in any robust system. For this reason we define robustness of a V2I service as to provide consistent real time service to V2I users in the presence of heterogeneity and fluctuating (burst) request arrival to Base Stations. Our main goal in this research is to offer a robust V2I service to users so that users can have QoS while moving in the roads and can concentrate only on driving.

II. Scenario

In V2I communication an ideal scenario of serving moving vehicles from road side Base Stations would be as follows: while moving through the roads, vehicles would be requesting tasks to the road-side Base Stations. As the request is generated, it would travel from the vehicle to the Base Station and after processing the task the result would be sent to the requesting vehicle. Majority of these tasks requested would be delay-sensitive which means the response of each request needs to reach the vehicle within a time frame of certain delay time. As the vehicles would be on the move, their locations would change with time and so, the direction of each vehicle is also crucial in delivering the response in time.

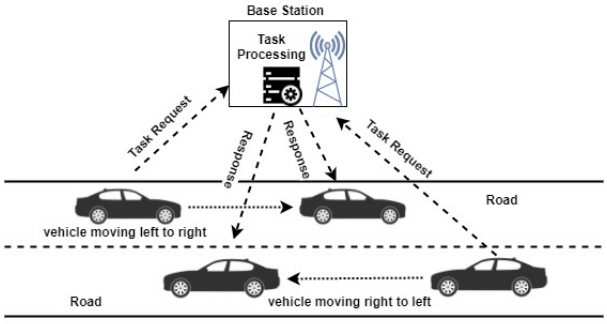


Fig: V2I scenario

III. Problem Statement and System Model

A. PROBLEM STATEMENT

The main goal of this research is to offer a robust Quality of Service in a V2I system in the presence of heterogeneity across Base Stations and oversubscription. In order to strengthen the support of the real time task processing in Base Stations, delay needs to be reduced with respect to time constratint. To specify a limited end-to-end delay for each task, we assign an individual deadline to each arriving task.

So the the problem is defined as: *how to allocate arriving tasks to a base station in a V2I network so that the number of tasks missing their deadline is minimized?*

B. SYSTEM MODEL

1) Formulation: According to V2I scenario, we assume a set of tasks will be generated by vehicles and send to the Base Station. Every task has its own deadline within which it has to be completed. So our system will allocate the tasks to Base Stations considering individual deadlines so that number of task missing their deadline can be minimized or number of task meeting their deadline can be maximized.

According to problem definition, there is a set of tasks "T" where T = $\{t_1, t_2, t_3, t_4 \ldots, t_n\}$ and a set of Base Stations "BS" where BS = $\{bs_1, bs_2, bs_3, bs_4 \ldots, bs_m\}$. The set of tasks that meet their deadline can be denoted as, $T_s \subseteq T$.

In this case the random variable is the number of tasks which is an integer. So the problem can be defined as Integer Programming Problem.(IPP). We assume the number of task is τ and their individual deadline is δ. Then the problem can be formulated as below :

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \sum_{i=1}^{n} f(x_i), where\ x_i \in \tau_i \\ \text{subject to} \quad & t_i \leq \delta_i,\ where\ t_i \in \tau_i \qquad i = 1, \ldots, m. \end{aligned}$$

Here idicating function is , $t_i = \begin{cases} 1, & if \quad t_i \leq \delta_i \\ 0, & Otherwise \end{cases}$

2) Assumptions: For the mentioned problem, there are some assumptions that need to be considered for the system model.

- Different services provided by a vehicular network are offered through processing vehicular tasks. As such, we can categorize arriving tasks to a vehicular network under different *task types*.
  - Task types represent different types of requests users can submit (e.g., hazard around an area, gas stations nearby, weather forecast).
- Due to heterogeneity across Base Stations, one task type has different expected completion times in different Base Stations.
- Upon arrival of a task to a base station, it is assigned an individual deadline based on its arrival time and the end to end delay it can tolerate.
  - Communication delay(uplink,downlink delay) is significant and must be considered in calculating end to end delay.
- For arriving task i , deadline δ_i can be defined as :
  - $\delta_i = t_i + E_i + \varepsilon + \beta$, where E_i is the estimated task execution time of task t_i , ε is a constant value defined by the system. (slack time) and β is the communication delay.
- The arrival time and deadline (denoted as i) are available with the task. So absolute deadline can be defined as $\Delta i = \delta_i + \psi_i$, where ψ_i is arrival time.
- Task arrival rate to the base station assumed to be unknown.
- The receiving Base Station is assumed to be oversubscribed:

  - Oversubscription is defined as a Base Station that it cannot handle all tasks before their deadlines.
- It is assumed that arriving tasks are sequential meaning each task needs one cpu/core for processing.
- If a task cannot meet its deadline and the task is delay sensitive then it is dropped
- A cloud server is connected to a Base Stations (Fog/Edge devices) for failover or compute intensive tasks that can not be processed in Base Stations.

3) Delay Estimation: In a V2I system, three distinct factors contribute to the definition of end-to-end delay (D_{V2I}). They are d_U, d_{BS} and d_D. So we can define V2I end-to-end delay as,

$$D_{V2I} = d_U + d_{BS} + d_D \tag{1}$$

Where d_U = uplink delay, d_{BS} = delay in base station and d_D = downlink delay. From equation (1) d_{UP} and d_D can be defined as,

For a task t_i requested from vehicle i to Base Station m, uplink delay from i to m is

$$d_U = \frac{L_i}{g(i,m)} \tag{2}$$

and for t_i travelling back from m to i downlink delay is

$$d_D = \frac{L_i}{g(m,i)} \tag{3}$$

where L_i is the task data size, g(i,m) and g(m,i) is the effective transmission data rate for the link i to m(uplink) and from m to i(downlink) respectively.

4) System Model Scenario: At first arriving tasks goes through a load balancer to a Base Station. The load balancer allocates task to a suitable Base Station and it serves the process in immediate mode. So whenever a task enters into a Base Station's load balancer, it is immediately gets the decision of allocation.

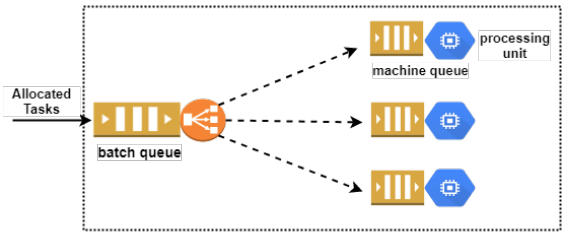


Fig : System Model Scenario

When a task is allocated to a Base Station, it enters into the batch queue of that Base Station for Processing. From arrival, to the end of task processing impose a delay which can be defined as "computational delay"(d_c).

Therefore d_{BS} can be defined as : d_{BS} = d_c, where d_c = average computational delay

- Therefore d_{BS} can be defined as : $d_{BS} = d_{TQ} + d_{TP}$, where d_{TQ} = average task queuing delay and d_{TP} = average task processing delay.
- So we need to consider the 3 components(d_U, d_{BS} and d_D) of D_{V2I} delay for reducing end-to-end delay.

IV. Approach

When vehicles move along the roads they may generate tasks according to different situations. Base Stations are stationary edge/fog devices which are situated along road side. These Base Stations have computational capacity for processing vehicle's requested tasks. So moving vehicles request task to Base Station for processing and after completion of task processing, the result is sent back to requested vehicle. The requested task arrives at the nearest Base Station and then go through a load balancer which is the component for allocating tasks to appropriate Base Stations. The load balancer takes the decision of task allocation with consideration of maximizing task robustness. So the load balancer allocates arriving task to Base Station that offers highest task robustness.

Every Base Station has two matrices. One is Estimated Task Completion(ETC) time matrix and other one is Estimated Task Transfer (ETT) time matrix. These two matrices help the load balancer to work efficiently.

ETC Matrix contains estimated task completion time distribution($X \sim \mathcal{N}(\mu, \sigma^2)$) for different task type in different Base Stations. Within a task type there are different data size tasks which have different completion time in a Base Station.

The value of ETC matrix cell is μ(mean) and σ(standard deviation) which represent normal distribution of different task type in different Base Stations which is found from the historical execution values of tasks. In ETC matrix,every column define the Base Stations and every row define task type.

To capture the differences in the estimated completion time of a task type, we consider the worst- case analysis of completion time estimation, that is the sum of mean historic completion time plus it's standard deviation.

The Estimated Task Transfer(ETT) time matrix in every Base Station represent the historic task transfer time from received Base Station to other neighbouring Base Stations for different task types. The ETT matrix is created from running a batch of tasks from one Base Station to another Base Station and plotting them in a normal distribution($Y \sim \mathcal{N}(\mu, \sigma^2)$) with respect to their task type.

The cloud server send periodic pulse which updates the ETC and ETT matrix values according to Base Stations current status.

When load balancer get the task "t_i" of task type "i", it calculates the probability(P_i^j) of meeting the task's deadline δ_i across the Base Stations.

For received Base Station "j", the probability can be defined as $P_i^j(\gamma_i^j < \delta_i) = P_i^j(Z < z)$ where "z" is $(\delta_i - \mu_i^j) / \sigma_i^j$.

For neighbouring Base Station before calculating the probability we convolve ETC matrix normal distribution with ETT matrix normal distribution and resulting distribution is also a normal distribution which can be defined as $W \sim \mathcal{N}(\mu, \sigma^2) = X \sim \mathcal{N}(\mu, \sigma^2) \circledast Y \sim \mathcal{N}(\mu, \sigma^2)$

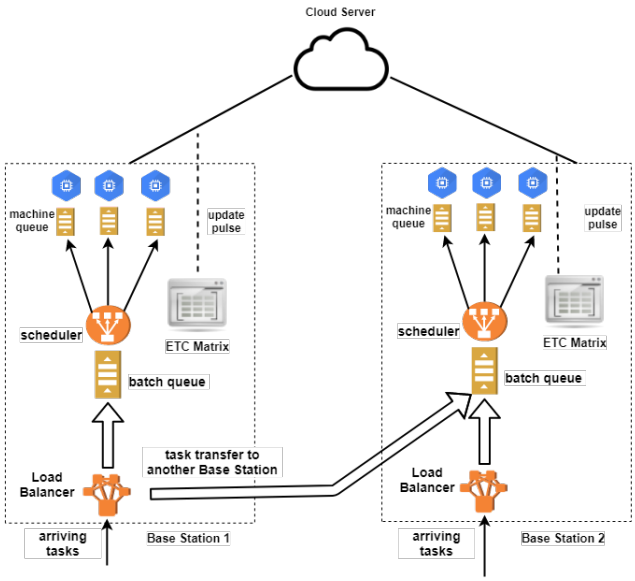


Fig : Approach Overview

The resulting distribution (W$\sim \mathcal{N}(\mu, \sigma^2)$) is used to calculate the probability of the task in that specific Base Station. If neighbouring Base Station is "k" and task type "i" then the probability can be defined as "P_i^k" where z = (δ_i - μ_i^k) / σ_i^k.

After calculating the probability of received task in all local Base Stations (received and neighbouring), It is transferred to highest probability Base Station. When task's probability of meeting deadline is zero(0), then the task is dropped. Dropped task does not get allocated to any Base Stations. Therefor it does not enter into the batch queue of any Base Station which does not increase the historic mean of executing task.

V. Heuristics

A. Load Balancer Heuristics

Inside a Base Station the Load Balancer works in immediate mode. It means each task is allocated to Base Stations immediately upon its arrival to load balancer. There is a buffer queue in the load balancer which stores tasks if there are many tasks coming to the load balancer. For simplicity the buffer limit is assumed as infinite. Whenever a task is assigned to a Base Station for processing there is no re-allocation process due to the overhead of task transfer latency.

1) Maximum Robustness(MR): As the name goes, this heuristic works on maximizing the robustness of each task. We defined "robustness" of a task is the probability of that task meeting it's deadline. So when a task comes to the load balancer, it calculates the probability of that task meeting it's deadline in receiving Base Station as well as neighbouring Base Stations. From that probabilities, the Base Station which gives highest probability or maximum robustness of that task, is being transferred by the load balancer.

2) No Redirection(NR): This heuristic works on serving the task from the Base Station, the task enters first. So whenever tasks enters a Base Station, the load balancer of that Base Station calculates the probability of that task meeting it's deadline in the received Base Station. If the task has probability greater than a certain value than it is allocated to the Base Station. On the other hand if the task does not satisfy the probability constrain than it is dropped. There is no redirection or transferring of tasks in this heuristic.

3) Minimum Expected Completion Time(MECT): The Minimum Expected Completion Time(MECT) heuristic works on minimizing the expected task completion time. This heuristic use the average (μ) completion time from ETC matrix for taking decision of task allocation. So the load balancer checks the arriving task's average completion time in receiving Base Station and other neighbouring Base Stations. It allocates the task to the Base Station, where the task has minimum completion time.

B. Scheduler Heuristics

1) FCFS: The scheduler use First Come First Serve policy for scheduling the tasks in batch queue. So the task which has arrived first, stays in queue head and late arriving task stays in queue tail. Scheduler schedule task from queue head to VM's local queue whenever free slot appears.

2) SDF: In this heuristic, the scheduler leverage the task's with soonest deadline. It means that tasks in the batch queue is arranged according to their deadline and soonest deadline task gets the highest priority. So task with soonest deadline stays in the queue head and whenever there is a free slot appears in VM's local queue, the task with soonest deadline assigned to the queue to be processed.

VI. Simulation

For simulation of the system model "cloudsim" is used. Cloudsim is a java library for simulating cloud computing models. In our implementation we used "Data Centers" as Base Stations. Within a Base Station there is one machine with multiple virtual machines(VM's). We used 5 VMs in each Base Station with same computational capacity which represents homogeneous nature within a Base Station. VM's computational power is represented in MIPS(Million Instructions Per Second). For creating heterogeneity across Base Stations different Base Station is defined with different VM MIPS. For simulating the scenario 3 Base Stations are used with different computational power.

Vehicular tasks are represented as "cloudlets" with different configurations. Different cloudlets have different MIs(Million Instruction). The requested tasks have different parameters (e.g., taskType, taskID, length, deadline, fileSize etc).

VII. Experiments

1) Impact of Load Balancer:

a) Increasing over subscription level: In this experiment the impact of load balancer is tested with two different settings. In first setting we use 50 tasks to create over subscription in each Base Station. And in second setting over subscription is created with 100 tasks.

The simulation is run with starting workload of 100 tasks and incrementing 100 tasks in every trial. So simulation is run for 30 trial with different seeds using Load balancer and

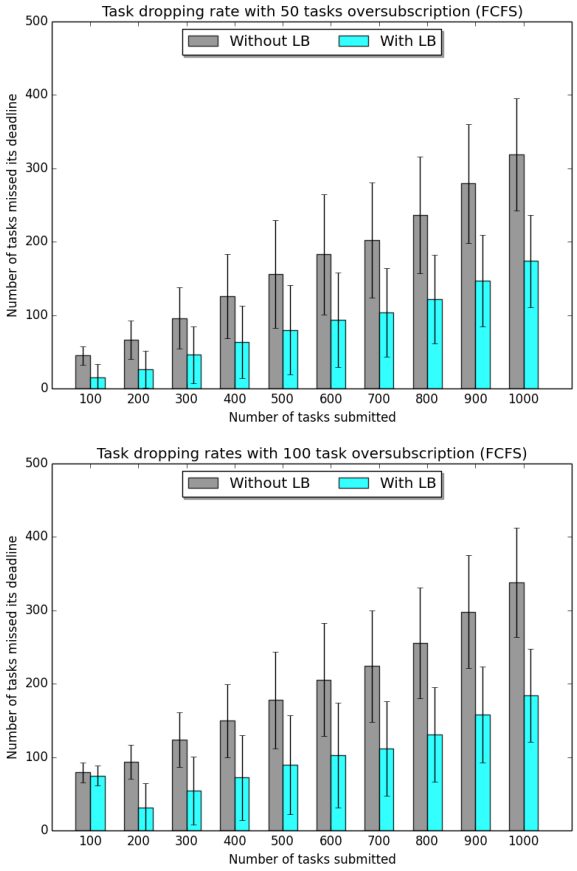
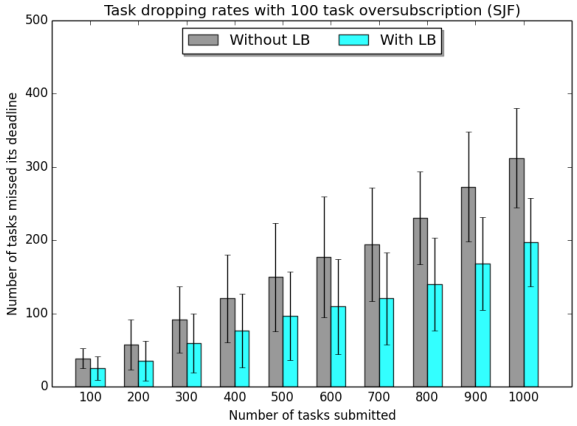


Fig. 1: With and without Load Balancer with 50 and 100 task over subscription



without Load Balancer. The average of every 30 trial is used for plotting barchart.

For this experiment the load balancer assigned the tasks to Base Station with highest probability. Here the probability condition of task dropping is zero which means that the highest probability value needs to be greater than zero. If there is no such Base Station then task is dropped or not allocated. This experiment is the implementation of MR heuristic which is mentioned in heuristic section. For this experiment two different scheduling heuristics are used. They are FCFS and SJF.The result plotting is given below :

From the above result it is visible that using load balancer leads to better performance. For smaller number of tasks difference between them is negligible. With the increased number of tasks load balancer works better than conventional scenario. From the two scheduling policy FCFS works better than SJF.

b) Task Dropping with given probability: In earlier experiment we used highest probability Base Station with task dropping for zero probability. But in this experiment we used task dropping condition with given probability. So we used a certain probability value below which we will not allocate tasks to Base Stations.

There are 4 types of experiments are considered in our system.

- X axis is no of tasks Y-axis is no of task meeting deadline. (for 3 L.B heuristics -MR, NR,MCCT with probability of task dropping is set to zero)
- X axis no of tasks Y-axis no of task meeting deadline. (for 3 L.B heuristics -MR, NR,MCCT with probability of task dropping is set to a certain value after running simulation)
- X axis is no of arriving tasks Y-axis is no of task meeting deadline.(Using different scheduler policy FCFS,SDF)
- Behavior of system deadline tightness.

VIII. Conclusions

In this paper, we have proposed a robust resource allocation technique for delay sensitive applications or tasks in vehicular network under low latency constraints. In the proposed model we have a load balancer module which take efficient decision for selecting best possible Base Station considering task's robustness and improvement of reliability of whole system.

Acknowledgments

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000097.

References

[1] GG Md Nawaz Ali and Edward Chan. Co-operative data access in multiple road side units (rsus)-based vehicular ad hoc networks (vanets). In *Telecommunication Networks and Applications Conference (ATNAC), 2011 Australasian*, pages 1–6. IEEE, 2011.

[2] Kyoungsoo Bok, Seungwan Hong, Jongtae Lim, and Jaesoo Yoo. A multiple rsu scheduling for v2i-based data services. In *Big Data and Smart Computing (BigComp), 2016 International Conference on*, pages 163–168. IEEE, 2016.

[3] Jun Li, Carlos Natalino, Dung Pham Van, Lena Wosinska, and Jiajia Chen. Resource management in fog-enhanced radio access network to support real-time vehicular services. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 68–74. IEEE, 2017.

[4] Petri Luoto, Mehdi Bennis, Pekka Pirinen, Sumudu Samarakoon, Kari Horneman, and Matti Latva-aho. System level performance evaluation of lte-v2x network. In *European Wireless 2016; 22th European Wireless Conference; Proceedings of*, pages 1–5. VDE, 2016.