A Priority based Resource Allocation Algorithm for LTE-V2X Mode 3

Anit Mangal
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, India
Email: anit.mangal@kgpian.iitkgp.ac.in

Arobinda Gupta
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, India
Email: agupta@cse.iitkgp.ac.in

***Abstract*—Connected vehicles are seen as an important part of future transportation systems, with direct vehicle-to-vehicle (V2V) communication enabling a variety of applications. LTE-V2X Mode 3 has been proposed as a cellular infrastructure based enabler for V2V communication. In Mode 3, a vehicle requests the eNB for radio resources and the eNB allocates resources to a vehicle for interference-free communication for a duration. Efficient allocation of radio resources to requesting vehicles so that the QoS requirements of the applications running in the vehicles are met is both important and challenging. In this paper, we first propose *PT-Sched*, an online resource allocation algorithm that considers minimum reuse distances and past allocation patterns to prioritise those vehicles that have been denied allocation more. We then extend the algorithm to propose *PT-Sched-2C* which works for two classes with different priority. The performance of the two algorithms are evaluated through detailed simulation in a realistic scenario and compared with two existing resource allocation algorithms, the one proposed in [1], and *PEARL* [2], to show that the proposed algorithms achieve significant improvement over both of them in most cases.**

***Index Terms*—LTE-V Mode 3, Resource Allocation, V2V**

I. Introduction

Connected vehicles are seen as an important part of future transportation systems, enabling a variety of applications. The term connected vehicles refers to a broad spectrum of scenarios in which vehicles can communicate with other entities in their environment, such as other vehicles, pedestrians, roadside units, cellular infrastructure etc. Of these, Vehicle-to-vehicle (V2V) communication enables vehicles to connect and exchange real-time data about speed, position, road conditions and other information, enabling a variety of safety, traffic management, and user-convenience applications.

Early implementations of vehicle-to-vehicle communication have relied on Dedicated Short-Range Communication (DSRC), an IEEE 802.11p-based technology. In recent years, Long-Term Evolution Vehicle-To-Everything (LTE-V2X) has emerged as a cellular alternative, offering enhanced reliability through existing cellular infrastructure and improved spectrum efficiency. The 3rd Generation Partnership Project (3GPP) [3] has specified a standard for Vehicle-to-Everything (V2X) communications based on the Long Term Evolution (LTE) radio interface by introducing PC5 sidelink communication for direct vehicle interactions while maintaining backward compatibility with cellular networks. The technology would enable vehicles to connect to various entities in their environment, including other vehicles (V2V), infrastructure (V2I), pedestrians (V2P), and networks (V2N).

LTE-V2X presents two modes of resource allocation: a centralised under-coverage mode, Mode 3, and a distributed out-of-coverage mode, Mode 4. In Mode 3, a vehicle requests for radio resources from the base station or eNB (evolved Node B) in the cellular infrastructure; the eNB then allocates/schedules the radio resources to the vehicle if available. In contrast, Mode 4 employs a distributed scheduling algorithm where vehicles autonomously select the radio resources through the 3GPP standardised sensing-based semi-persistent scheduling (SPS). Unlike Mode 4, no specific resource allocation algorithm has been standardised for Mode 3. While Mode 3 is easier to implement due to the involvement of the eNB, the challenge is in allocating resources to vehicles efficiently so that the Quality of Service (QoS) requirements of the applications running in the vehicles are met. In this paper, we consider resource allocation algorithms for Mode 3 based operations in LTE-V2X for vehicle-to-vehicle communication. As the radio resources are allocated to ensure an interference-free communication channel for a vehicle to use for a particular duration, in the rest of this paper we will use the term *resource* and *channel* interchangeably. Similarly, as the task of allocating a channel to a requesting vehicle is the same as that of scheduling available channels among the set of requesting vehicle at a time, we will also use the terms *allocation* and *scheduling* interchangeably in the rest of this paper.

There exists several works that address channel allocation schemes for LTE-V2X Mode 3. However, most existing works do not properly consider application-specific requirements or past resource allocation history in making efficient channel allocation decisions. In this paper, we address this research gap. In particular, we first propose *PT-Sched*, an online channel allocation algorithm that considers past allocation patterns and reuse distances to prioritise those vehicles that have been denied allocation more. We then extend the algorithm to propose *PT-Sched-2C* which works for two classes with different priority. The performance of the two algorithms are evaluated through detailed simulation in a realistic scenario

978-3-903176-77-5© 2025 IFIP

and compared with two existing resource management algorithms, the one proposed in [1], and *PEARL* [2], to show that the proposed algorithms achieve significant improvement over both of them in most cases. We also show comparison with an optimal approach in a limited scenario to indicate that PT-Sched may have good performance in many cases while taking very little time to make it practically feasible for online channel allocation at fine-grain timesteps in eNBs.

The rest of the paper is organised as follows. Section II briefly describes related works done in the area. A formulation of the problem is presented in Section III. Section IV describes the proposed algorithms, PT-Sched and PT-Sched-2C. The simulation setup and results are presented in Section V. Finally, Section VI concludes the paper.

II. Related Works

Radio resource allocation for LTE-V2X Mode 4 has seen significant work in recent years. The works in [4] [5] [6] optimise the parameters of the standardised sensing-based semi-persistent scheduling (SPS) algorithm. In contrast, the works in [7] [8] modify the sensing-based SPS algorithm itself, and those in [9] [10] propose alternative algorithms for vehicles to select the radio resources autonomously.

Radio resource allocation for LTE-V2X Mode 3 operation, the focus of this paper, has also seen significant works in recent times. 3GPP standards already specify procedures for vehicles to report their location to the cellular network [3] [11]. Most existing works in Mode 3 use the geographical locations of the vehicles, and focus on mitigating interference and allocating resources based on signal-to-interference-plus-noise ratio (SINR), noise power, distance between vehicle etc. Cecchini et al. [1] propose a resource allocation scheme that first uses vehicle positions and parameters such as required awareness range, SINR, and noise power to calculate minimum reuse distance to avoid interference. It then allocates radio resources, prioritising vehicles that have been denied allocation before. In a similar approach, Zhang et al. [12] use the Euclidean distance to compute reuse distance. Sehla et al. [13] propose a clustering algorithm based on Maximum Inter-Centroids Reuse Distance (MIRD). The main goal is to efficiently allocate resources to vehicles by applying a specific resource reuse mechanism between clusters. Sempere-Garcia et al. [14] propose a context-based scheduling scheme, *aDaptive spatIal Reuse of rAdio resourCes (DIRAC)*, that exploits the geographical location of vehicles and dynamically configures its operation with the objective that all vehicles experience a similar level of interference when resources must be shared. In another work [15], the resource allocation problem is modelled using Game Theory as a Stackelberg Game using a price-penalty mechanism. Zhang et al. [16] propose a fuzzy logic-based dynamic resource scheduling algorithm for Mode 3. They consider multiple classes of applications and consider four different factors in allocating a channel. Other works in Mode 3 focus on providing Quality of Service (QoS). Liang et al. [17] analyse the different QoS requirements for V2X links. Their work studies the optimal power allocation for each reused Cellular User Equipment - Vehicle User Equipment (CUE-VUE) pair and uses the Hungarian algorithm to perform resource assignment. Allouch et al. [2] propose *PEARL*, a resource allocation scheme to handle applications belonging to two classes, safety and non-safety, with a minimum percentage of resources kept reserved for safety applications.

Although the works mentioned above have proposed several resource allocation schemes for Mode 3, they either do not consider application-specific requirements or do not use priority scheduling effectively. Among the works discussed, the two works that are closest to our work proposed in this paper are the ones in [1] and [2]. While [1] considers some past allocation history in assigning channels, it does not differentiate between the degree to which vehicles are denied allocation in the past. In particular, it first considers the set of vehicles that were not allocated a channel in the previous timestep and tries to allocate resources to them randomly; it then considers the remaining vehicles. Similarly, the work in [2] does consider two application classes, but only statically prioritizes one over the other. It reserves a part of the resources for the class with higher priority. At each timestep, it tries to allocate resources for that class in the reserved part first and then uses the resources in the non-reserved part if still needed. Allocations to the lower priority class is considered at the end from the remaining resources in the non-reserved part. It does not prioritise within an application class, and does not have any scope for dynamic priority among applications belonging to different classes based on QoS requirements. We propose algorithms that use a more effective prioritization scheme and can handle multiple application classes.

III. Problem Formulation

We consider a city scenario with vehicles moving along roads in the city. Each vehicle is assumed to be equipped with LTE modules to connect to the cellular infrastructure, and uses LTE-V Mode 3 to directly communicate with other vehicles. There is a single eNB in the system. Let T be the total number of timesteps for which the system runs. Each vehicle runs an application that needs to communicate with other vehicles within its range periodically at fixed timesteps. A vehicle sends a request to the eNB to allocate it a channel for one timestep for this communication. The request also contains the current position of the vehicle. We assume that an application needs not more than one timestep to send application messages to other vehicles. The eNB responds to the request by either accepting the request and allocating a channel to the requesting vehicle, or rejecting the request. Since a vehicle is assumed to run a single application only in this work, we will use the terms *application* and *vehicle* interchangeably while referring to an application running in a vehicle in the rest of this paper.

Let $\mathcal{V}$ be the set of vehicles. For any vehicle $v \in \mathcal{V}$, let st_v and dt_v denote the trip start time and trip end time respectively of v, and let loc_v^t denote the location of the vehicle v at time t. Let $\mathcal{C}$ be the set of all channels. Let d_{reuse} be the minimum reuse distance, which is the minimum distance at which two different vehicles can use the same channel without causing interference. Let $dist(x, y)$ denote the Euclidean distance between locations x and y.

Each vehicle runs an application that sends messages to some other vehicles periodically. An application is characterised by a message request period, $\mathcal{RQ}_{app}$, that denotes the period (in number of timesteps) at which the application wants to send a message, and hence requests for a channel. As noted earlier, a channel may or may not be granted, and hence the application may or may not be able to send its message. Let $\mathcal{TH}_{app}$ denote a threshold number of periods, the number of consecutive message request periods, that is acceptable for the application's channel request to be denied. Messages sent depend on the application, and can be periodic broadcasts of regular information (such as speed and position of the vehicle) to nearby vehicles, or other information such as safety warnings or traffic advisory information. The threshold period indicates the time interval where the application needs to transmit at least once successfully to maintain its QoS requirement. However, since a channel may not be available when it wants to send a message, a vehicle tries to send more number of times within this threshold period, which is identified by the parameter $\mathcal{RQ}_{app}$. An application is said to have missed its threshold if it has not been allocated a channel for $\mathcal{TH}_{app}$ number of consecutive message request periods. Applications can belong to different classes. Two applications that belong to the same class are assumed to have the same values $\mathcal{RQ}_{app}$ and $\mathcal{TH}_{app}$; however applications belonging to different classes can have different values of $\mathcal{RQ}_{app}$ and $\mathcal{TH}_{app}$.

The solution for the channel allocation problem is a $|\mathcal{V}| \times T$ matrix, $\mathcal{A}$, where $\mathcal{A}[v, t] = b$ if and only if the vehicle v is allocated channel $b \in \mathcal{C}$ at time t, 0 otherwise. The channel allocation problem is specified as allocating channels to the vehicles while minimizing the total number of threshold misses by the applications running in the vehicles. For any vehicle $v \in \mathcal{V}$, let $\mathcal{T}_{req}$ denote the set of timestamps at which v requests for a channel, i.e., $\mathcal{T}_{req} = \{st_v, st_v + \mathcal{RQ}_{app}, \ldots, st_v + \lfloor((dt_v - st_v)/\mathcal{RQ}_{app})\rfloor \times \mathcal{RQ}_{app}\}$. Then the problem is to minimize

$$\sum_{v\in\mathcal{V}}\left(\sum_{t\in\mathcal{T}_{req}}\left(t - \max_{\substack{t'\in[st_v, st_v+1,\ldots,t]:\\ \mathcal{A}[v,t']\neq 0}}(t') > \mathcal{TH}_{app} \times \mathcal{RQ}_{app}\right)\right)$$

subject to the following constraints.

1) At any timestep, a vehicle can either be allocated a channel from $\mathcal{C}$ or denied allocation. i.e., $\forall v \in \mathcal{V},\ \forall t \in \mathcal{T} : \mathcal{A}[v, t] \in \mathcal{C} \bigcup \{0\}$
2) All vehicles allocated the same channel simultaneously have to be at least d_{reuse} distance away from each other, i.e., $\forall t \in \mathcal{T},\ \forall v, v' \in \mathcal{V} : (\mathcal{A}[v, t] = \mathcal{A}[v', t]) \wedge A[v, t] \neq 0 \implies dist(loc_v^t, loc_{v'}^t) \geq d_{\text{reuse}}$
3) A vehicle can only be allocated a channel during its trip at multiples of message request periods from the trip start time, i.e., $\forall v \in \mathcal{V},\ \forall t \in \mathcal{T} : \mathcal{A}[v, t] \neq 0 \implies [(t - st_v) \equiv 0\ (mod\ \mathcal{RQ}_{app})]$

IV. Channel Allocation Algorithms

This section describes two online heuristic algorithms (*PT-Sched* and *PT-Sched-2C*) that are run by the eNB to allocate channels to requesting vehicles.

A. Priority and Threshold based Channel Scheduling (PT-Sched)

The main idea of PT-Sched is as follows. At each timestep t, PT-Sched considers the set of vehicles that are requesting a channel at that timestep, and tries to allocate channels to them. For each such vehicle v, the algorithm finds the last timestep $alloc_v$ at which v was allocated a channel. The available channels are allocated in ascending order of $alloc_v$ values, i.e., the ones that have not been allocated a channel for the longer time (i.e., lower $alloc_v$ value) in the past are given higher priority. For two vehicles with the same $alloc_v$ values, tie is broken by giving higher priority to the one which has missed its threshold (if any), specifically to the one that has not been allocated a channel for a larger number of timesteps since missing the threshold $\mathcal{TH}_{app}$. This ensures that vehicles that have not been allocated a channel for more than the threshold number of periods are given the highest priority in allocation, with vehicles that have not missed the threshold given priority in order of their last allocation time. While allocating channels, we ensure that the channel assigned to a vehicle is not allocated to any other vehicle within the minimum reuse distance d_{reuse} to avoid interference. Algorithm 1 shows the pseudocode of PT-Sched.

Algorithm 1 PT-Sched Algorithm

1: **for all** $v \in \mathcal{V}_{t_i}$ **do**
2: $\quad alloc_v = st_v$
3: $\quad N_v = 0$
4: **end for**
5: **for all** $t_i \in T$ **do**
6: $\quad \mathcal{V}_{t_i}$ = set of vehicles requesting channels at t_i
7: $\quad A_b^{t_i} = \emptyset$
8: $\quad$ sort $\mathcal{V}_{t_i}$ in ascending order of $alloc_v$, then in descending order of N_v for conflicts
9: $\quad$ **for all** $v \in \mathcal{V}_{t_i}$ in order **do**
10: $\quad\quad$ **if** $t_i - alloc_v > \mathcal{TH}_{app}.\mathcal{RQ}_{app}$ **then**
11: $\quad\quad\quad N_v = N_v + 1$
12: $\quad\quad$ **end if**
13: $\quad\quad$ **for all** $b \in \mathcal{C}$ **do**
14: $\quad\quad\quad$ **if** $\exists v' \in A_b^{t_i}$: $dist(v, v') < d_{\text{reuse}}$ **then** continue
15: $\quad\quad\quad$ **else**
16: $\quad\quad\quad\quad \mathcal{A}[v, t_i] = b$
17: $\quad\quad\quad\quad alloc_v = t_i$
18: $\quad\quad\quad\quad A_b^{t_i} = A_b^{t_i} \bigcup \{b\}$
19: $\quad\quad\quad$ **end if**
20: $\quad\quad$ **end for**
21: $\quad$ **end for**
22: **end for**

Algorithm 1 iterates over all timesteps, showing how channels are allocated at every timestep t_i by the eNB. In line 8, we sort the array of requesting vehicles ($\mathcal{V}_{t_i}$) in increasing order of $alloc_v$ values as described earlier. Between lines 9-20, we iterate over all vehicles in $\mathcal{V}_{t_i}$. If the vehicle has not been allocated a channel for $\mathcal{TH}_{app}$ number of message request periods, we increment N_v. Then, we iterate over all channels for that vehicle. We deem a channel eligible for allocation if the candidate vehicle is far enough from other vehicles allocated the same channel simultaneously. If an eligible channel is found, it is allocated to the vehicle and appropriate variables are updated.

Note that while the pseudocode shows the algorithm iterating over all timesteps to show the use of past allocation history, the eNB will execute the algorithm at each timestep considering all requests at that timestep only, and does not use any future information. Hence the algorithm is fully online.

B. Priority and Threshold based channel Scheduling for 2-class scenario (PT-Sched-2C)

Next, we look at the case when there are two classes of applications, Class 1 (for ex. safety applications) having higher priority than Class 2 (for ex. non-safety applications). The message request periods of both the classes are assumed to be the same, $\mathcal{RQ}_{app}$. Let $\mathcal{TH}^1_{app}$ and $\mathcal{TH}^2_{app}$ denote the threshold number of periods for Class 1 and Class 2 applications respectively. Unlike previous works, we do not reserve a fixed proportion of the channels for the messages belonging to the higher priority class. Instead, we prioritize a Class 1 message only if it is close to missing its threshold period value ($\mathcal{TH}^1_{app}$), or has already missed it. Otherwise, we treat both the classes as the same and prioritize them based on the same parameter as in PT-Sched. To this end, we introduce another threshold (number of timesteps) D^1 to indicate how close a Class 1 application is to missing the threshold $\mathcal{TH}^1_{app}$, with D^1 being positive if the application has not missed the threshold period yet and negative if it has already missed it.

The pseudocode for PT-Sched-2c is very similar to that of PT-Sched, and is not shown again. The main modification is in line 8 of Algorithm 1, where after sorting the vehicles $\mathcal{V}_{t_i}$ as shown there, PT-Sched-2C further rearranges the sorted $\mathcal{V}_{t_i}$ such that Class 1 vehicles with $alloc_v + \mathcal{TH}^1_{app} - t_i \leq D^1$ are brought to the front in the same order. This prioritises those vehicles that have Class 1 applications running and have already missed the threshold or are close to missing it. Also, line 10 is modified slightly to use the class-specific $\mathcal{TH}_{app}$ values ($\mathcal{TH}^1_{app}$ and $\mathcal{TH}^2_{app}$) depending on the application class running in the vehicle. Rest of the pseudocode remains the same.

V. Simulation Results

The performances of PT-Sched and PT-Sched-2C are evaluated using detailed simulations in different scenarios. We compare the proposed algorithms with two existing algorithms, the algorithm in [1] (referred to as *Benchmark* in the rest of the paper) and the algorithm PEARL [2]. The details of this algorithms are already described in Section **??**. For comparing with PT-Sched, we restrict PEARL to one class only for fair comparison. Finally, we also compare the performance of PT-Sched in a specific scenario with an optimal approach to solving the problem using Integer Linear Programming to get an idea about the practical advantage of using PT-Sched with respect to time.

Fig. 1. Road Network of Lower Manhattan

(a) *OpenStreetMap* of the area (b) *Polyconverted* Road Network

Fig. 2. Road Network of Lower Manhattan

TABLE I
Parameter Values Used for PT-Sched

Parameter	Value
Number of Vehicles	500, 1000, 2000
Vehicle Spawn End Time	50 sec
Message Request Period $\mathcal{RQ}_{app}$	1 sec, 2 sec, 3 sec
Threshold Number of Periods $\mathcal{TH}_{app}$	2, 4
Minimum Trip Distance	750 m
Number of LTE-V channels	12
d_{reuse}	300m

A. Simulation Setup

We use a map of the Lower Manhattan region in New York, USA obtained from *OpenStreetMap*. The area considered spans around 10 square kilometers. The region considered and the road network derived from it are shown in Fig. 1. We use the open source traffic simulator Eclipse SUMO (Simulation of Urban MObility) to generate the vehicular traffic on the roads. A vehicle trip is approximately 3 kilometres long and lasts 15 minutes on average. All vehicles are started randomly within the first 50 seconds. All results reported are the average of five runs.

The parameters used for evaluating PT-Sched are summarized in Table I. If the *value* column contains only one value, the parameter is fixed for the entire simulation; otherwise, results are generated on the combinations of all the values mentioned.

The parameters for evaluating PT-Sched-2C are listed in Table II. In this case, as noted earlier, the message request periods of the two classes are kept the same, but the threshold number of periods is different. For PEARL, we also use the same value for the percentage of the resource block exclusively reserved for the higher priority class as specified in [2].

TABLE II
PARAMETER VALUES USED FOR THE PT-SCHED-2C

Parameter	Value
Number of Vehicles	500, 1000, 2000
Vehicle Spawn End Time	50 sec
Message Request Period $\mathcal{RQ}_{app}$	1 sec
Class 1 Threshold Number of Periods $\mathcal{TH}^1_{app}$	5
Class 2 Threshold Number of Periods $\mathcal{TH}^2_{app}$	5, 8, 10
D^1 (Number of Periods)	2
% of vehicles running Class 1 application	30%
% of vehicles running Class 2 application	70%
PEARL Class 1 Guaranteed Resources	30%
Minimum Trip Distance	750 m
Number of LTE-V channels	12
d_{reuse}	300m

B. Evaluation Metrics

The following metrics are used to evaluate the performance of the algorithms:

1) **The average number of threshold misses:** This is defined as the average number of times a vehicle misses its threshold. The average is taken over all vehicles. This metric tries to capture how many times an application failed its basic QoS requirement of sending a message at least once during a maximum threshold period.
2) **The maximum number of periods a vehicle is not allocated a channel after missing its threshold:** This is defined as the maximum number of message request periods a vehicle is not allocated a channel after missing the threshold number of periods. The maximum is taken over all threshold misses in all vehicles. This shows how applications whose needs have become critical (missed their threshold period) are prioritised by the algorithms.
3) **The percentage of thresholds missed per vehicle:** This metric is computed as follows. For each vehicle, we divide the total trip duration into intervals spanning its threshold periods, and then count the percentage of those intervals in which the vehicle was not able to send any message (not allocated resource). This is then averaged over all vehicles. This metric shows the extent to which a vehicle is missing its threshold over its entire trip duration.
4) **The percentage of request denials per class:** For PT-Sched-2C, this metric tells the degree to which message requests were denied for each class separately.

C. Results for PT-Sched

In this section, we show the results obtained by varying the *number of vehicles*, the *message request period* and the *threshold number of periods*. The inputs to PT-Sched, Benchmark and PEARL are kept the same in each run.

1) Average number of threshold misses: Fig. 3 shows the variation of the average number of threshold misses with the number of vehicles, the message request period, and the threshold number of periods. From the figure, it can be seen that as we increase the number of vehicles, irrespective of the other parameters, the number of threshold misses tends to increase. This is because increasing the number of vehicles leads to an increase in the density of vehicles at a particular time, causing an increase in the rate of requests which overloads the system.

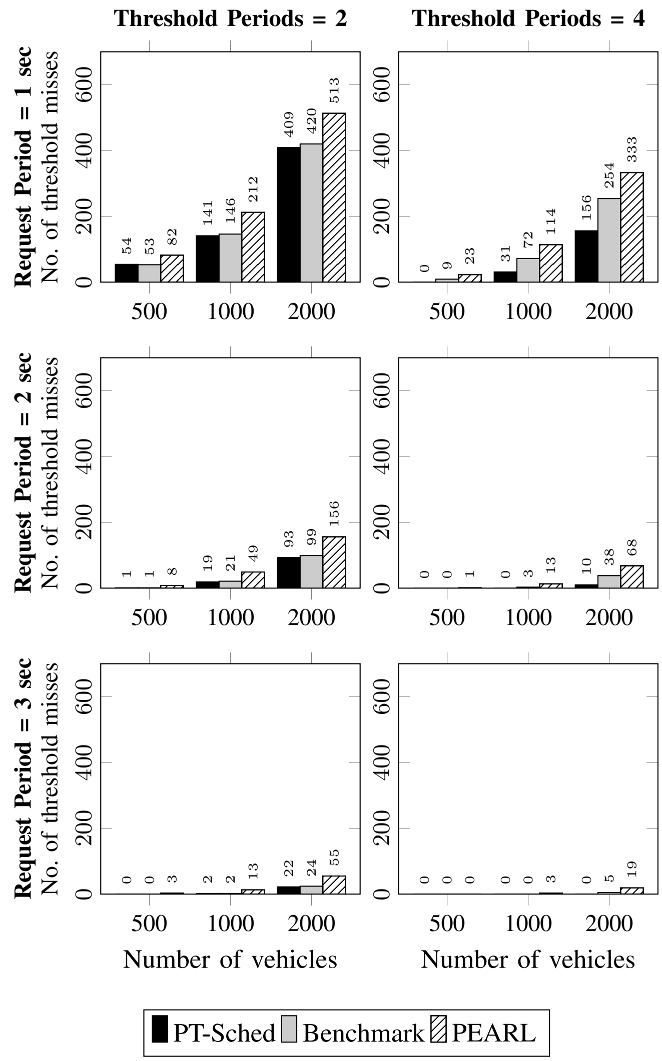

Fig. 3. Average number of threshold misses

As expected, if we increase the threshold number of periods, fewer threshold misses are seen as the algorithm can make more attempts to allocate channels within the threshold period. An increase in the message request period decreases this number as the rate of requests is lower.

As seen, PT-Sched outperforms both Benchmark and PEARL. Benchmark gives higher priority to vehicles that have been denied allocation; however, it allocates resources to the prioritised vehicles randomly. So, a vehicle can be denied multiple times if the number of channels is insufficient, leading to potential starvation. However, PT-Sched uses more fine-grain prioritisation that increases the priority of a vehicle dynamically as it is denied a channel more and more in consecutive timesteps. Similarly, PEARL allocates channels

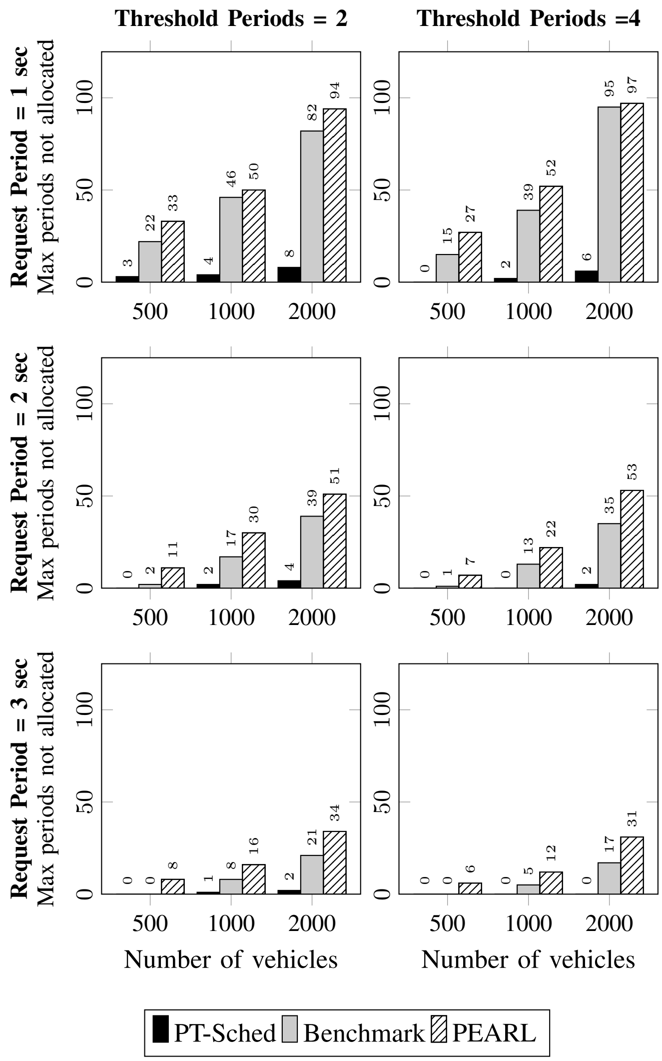

Fig. 4. Maximum number of periods not allocated after threshold

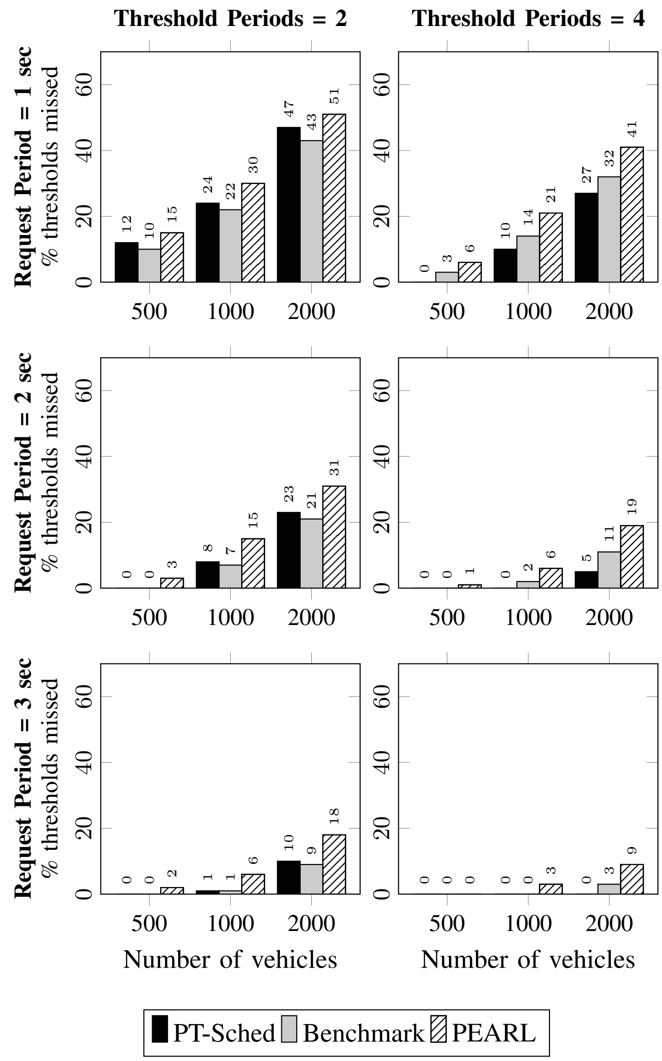

Fig. 5. Percentage of per-vehicle thresholds missed

within a class randomly, and does not use any prioritization among vehicles in the same class. PT-Sched prioritises vehicles whose requests have been denied more in the past, causing the priority of a vehicle to increase as it approaches its threshold, which in turn causes less threshold misses.

2) Maximum number of periods a vehicle is not allocated after threshold: Fig 4 shows the maximum number of periods a vehicle is not allocated any channel after missing its threshold. It can be seen that increasing the number of vehicles increases this number. This is because with more requesting vehicles, more vehicles request a channel simultaneously, and as the number of channels is constant, more vehicles are denied a channel, causing more threshold misses in turn. Allocating a channel to those vehicles after they have missed the threshold also takes more time for the same reason. Increasing the threshold number of periods or increasing the message request period reduces the rate of requests and threshold misses, enabling more prompt serving of a request even if a vehicle misses its threshold.

As seen from the figure, PT-Sched performs better than both Benchmark and PEARL. This is because it orders the set of candidate vehicles while allocating channels, such that the vehicle that has not been allocated a channel the longest is considered first. This causes vehicles that have missed their thresholds to have very high priority in allocation. Benchmark and PEARL randomly pick from the set of candidate vehicles and do not prioritise those critical vehicles that have already missed their basic QoS requirement of sending at least one message every threshold number of periods.

3) Percentage of thresholds missed per vehicle: Fig. 5 shows the percentage of thresholds missed per vehicle during their entire trip. Typically, increasing the number of vehicles increases this percentage. The algorithms can allocate a higher percentage of requests within the threshold if the threshold valuer is increased, thereby decreasing this percentage. The effect is similar if we vary the request period.

It can be seen that PT-Sched has better results here than PEARL since it classifies and prioritises vehicles based on

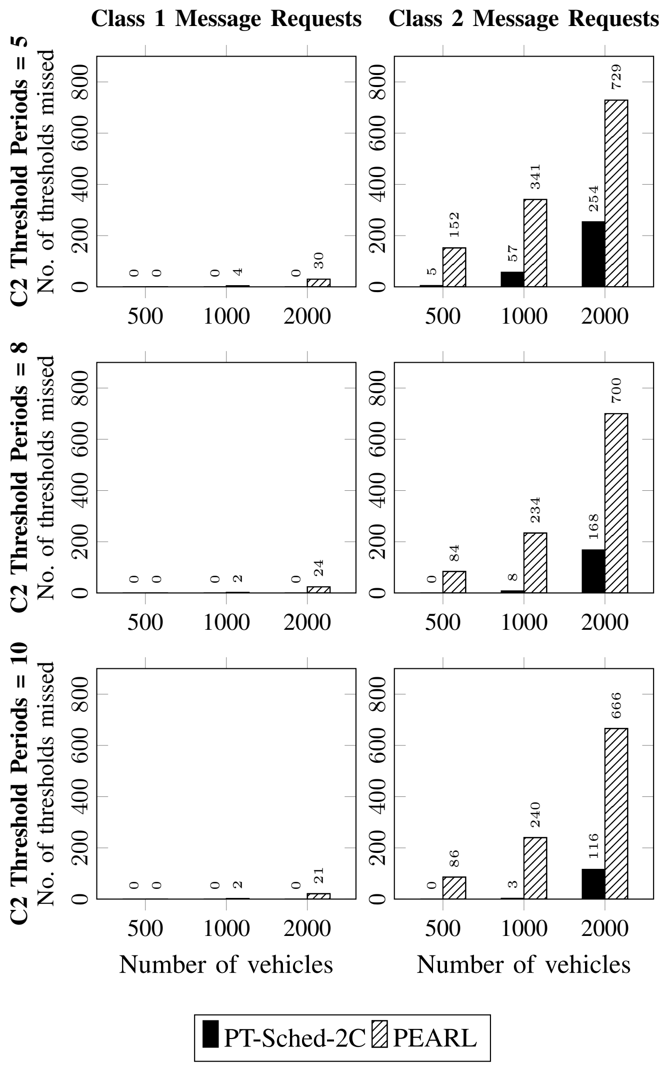

Fig. 6. Average no. of thresholds missed per class

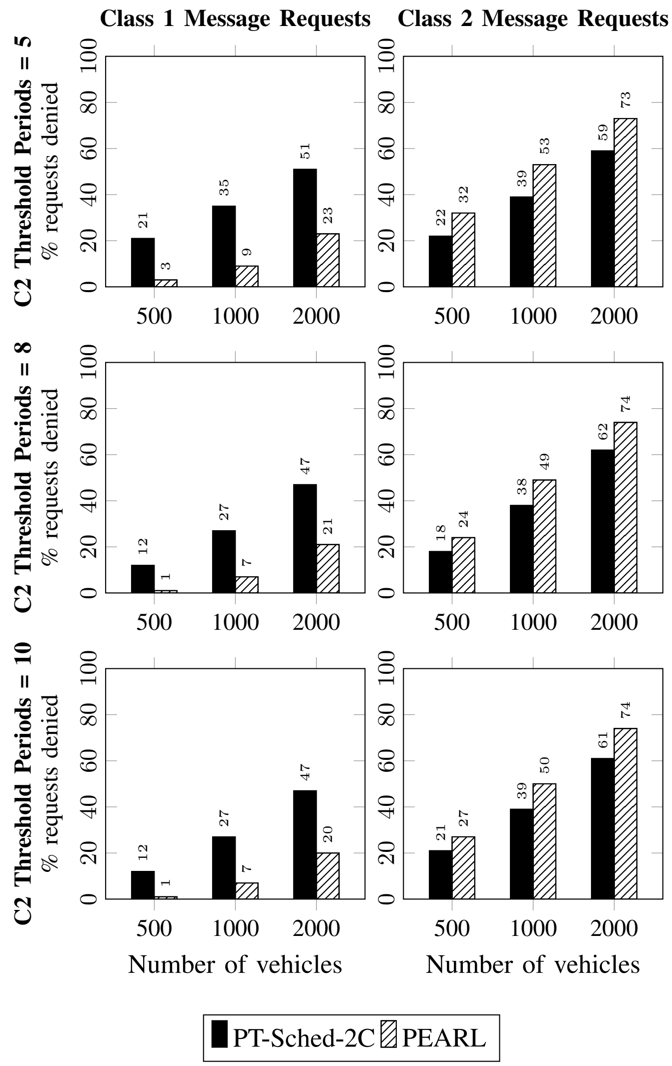

Fig. 7. Percentage of denials per class

the time of last allocation of a channel. PT-Sched performs almost the same as Benchmark since both algorithms consider the vehicles that have been denied until the current timestep first, though they prioritise them differently.

D. Results for PT-Sched-2C

We next show the results for PT-Sched-2C. In this case, the vehicles run two classes of applications, having a distribution of 30% vehicles running Class 1 (higher priority) applications, and 70% vehicles running Class 2 (lower priority) applications. We compare PT-Sched-2C only with PEARL as Benchmark is not designed to work for more than one class.

1) Average number of threshold misses per class: Fig. 6 shows the variation of the average number of threshold misses per class with the number of vehicles and the Class 2 threshold number of periods. The general trends remain the same as those seen for PT-Sched. For both algorithms, the threshold misses are higher for Class 2 messages, which could be acceptable in a real-world scenario where Class 2 may be used for non-critical applications, while using Class 1 for more critical safety-related applications.

Although both algorithms perform satisfactorily for Class 1 requests, PT-Sched-2C has a much lower number of threshold misses than PEARL for Class 2. This is because after serving the close-to-deadline Class 1 requests, PT-Sched-2C does not particularly prioritise Class 1 over Class 2 as PEARL does. So, both Class 1 and Class 2 message requests get a chance for a channel allocated within their thresholds.

2) Percentage of request denials per class: Fig. 7 shows the variation of the percentage of message request denials per class with the number of vehicles and the Class 2 threshold number of periods. For Class 1, it is seen that PEARL accepts significantly more requests than PT-Sched-2C. However, the application's QoS criterion for the algorithms is considered to be the ability of the algorithms to allocate requests within the threshold number of periods for an application to be able to send at least one message within the threshold. PT-Sched-2C performs better if we look at this QoS criterion as seen from

TABLE III
COMPARISON OF PT-SCHED AND HIGHS

Method	# of threshold misses	Max periods vehicle is denied	% of per-vehicle thresholds missed	Max time per timestep (ms)	Average time per timestep (ms)
PT-Sched	35.37	2.2	7.753	78.12	9.737
HiGHSa	35.06	2.2	7.745	31059.4	523.961
HiGHSb	35.32	2.2	7.868	32537.5	532.498

the earlier results. For Class 2, PT-Sched-2C denies a lower percentage of requests than PEARL, since it does not entirely prioritise Class 1 over Class 2 during allocation.

E. Comparison with Optimal Result

The problem formulated can also be modeled as an equivalent Integer Linear Programming (ILP) problem. We implemented the ILP problem using HiGHS [18], an open-source solver, which gave the best results among several other open-source solvers explored. The motivation for the implementation was to get an idea of the trade-off between the time required and the quality of solution obtained if the problem is solved optimally and if a heurstic algorithm like PT-Sched is used.

Table III compares an average of 5 solver runs for 500 vehicles, with a message request period of 1 second and threshold of 2 periods. The methods are compared based on the evaluation metrics for comparing the quality of the solutions, and additionally, the maximum and average time taken by the program to allocate resources per timestep. HiGHSa shows the result of the optimal solver when it is started with the solution obtained from PT-Sched as initial input to reduce time. HiGHSb shows the result when the optimal solver is started with a random input. The solver was run on a system with an AMD Ryzen 5600H processor (6 cores, 3.3 GHz base clock, 4.2 GHz boost) and 16GB of DDR4-3200 MHz memory operating in single-channel mode.

It can be seen from the table that at least for this specific scenario, the performance of PT-Sched is almost as good as that of the solution provided by the optimal solver. However, the time taken by PT-Sched is very significantly lower. More specifically, the ILP solver when run at the eNB will most likely not be able to allocate resources at very fine-grain timesteps at scale, even with better hardware at the eNB. However, with the same basic hardware, PT-Sched takes a maximum of 78 milliseconds during the whole duration of the simulation, which is sufficient for making allocation decisions at very low timesteps. This indicates that the algorithm may be highly efficient while providing results close to the optimal solution.

VI. CONCLUSION

In this paper, we proposed two algorithms for LTE-V2X Mode 3 resource allocation, PT-Sched for a single class of applications, and PT-Sched-2C which is an extension of PT-Sched to handle two classes. Detailed simulation results are presented to show that the proposed algorithms perform significantly better than two other existing algorithms. As future extensions of this work, it would be interesting to explore if future trajectory information of the vehicles available from online navigation systems such as google maps can be helpful in making better resource allocation decisions.

REFERENCES

[1] G. Cecchini, A. Bazzi, B. M. Masini, and A. Zanella, "Localization-based resource selection schemes for network-controlled lte-v2v," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, 2017, pp. 396–401.

[2] M. Allouch, S. Kallel, A. Soua, and S. Tohme, "Pearl: A novel priority and guaranteed-based resource allocation approach for v2v communications in lte-v mode 3," in *2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2020, pp. 1–6.

[3] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.300, 01 2021, version 14.13.0.

[4] A. Bazzi, G. Cecchini, A. Zanella, and B. M. Masini, "Study of the impact of phy and mac parameters in 3gpp c-v2v mode 4," *IEEE Access*, vol. 6, pp. 71 685–71 698, 2018.

[5] R. Molina-Masegosa, J. Gozalvez, and M. Sepulcre, "Configuration of the c-v2x mode 4 sidelink pc5 interface for vehicular communication," in *2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2018, pp. 43–48.

[6] A. Bazzi, A. Zanella, and B. M. Masini, "Optimizing the resource allocation of periodic messages with different sizes in lte-v2v," *IEEE Access*, vol. 7, pp. 43 820–43 830, 2019.

[7] L. F. Abanto-Leon, A. Koppelaar, and S. H. de Groot, "Enhanced c-v2x mode-4 subchannel selection," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

[8] R. Molina-Masegosa and J. Gozalvez, "Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.

[9] R. Molina-Masegosa, M. Sepulcre, and J. Gozalvez, "Geo-based scheduling for c-v2x networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8397–8407, 2019.

[10] T. Sahin and M. Boban, "Radio resource allocation for reliable out-of-coverage v2v communications," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.

[11] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.331, 09 2024, version 18.3.1.

[12] X. Zhang, Y. Shang, X. Li, and J. Fang, "Research on overlay d2d resource scheduling algorithms for v2v broadcast service," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–5.

[13] K. Sehla, T. M. T. Nguyen, G. Pujolle, and P. B. Velloso, "A new clustering-based radio resource allocation scheme for c-v2x," in *2021 Wireless Days (WD)*, 2021, pp. 1–8.

[14] D. Sempere-García, M. Sepulcre, and J. Gozalvez, "Lte-v2x mode 3 scheduling based on adaptive spatial reuse of radio resources," *Ad Hoc Networks*, vol. 113, p. 102351, 2021.

[15] Z. Liu, J. Su, Y.-a. Xie, K. Ma, Y. Yang, and X. Guan, "Resource allocation in d2d enabled vehicular communications: A robust stackelberg game approach based on price-penalty mechanism," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8186–8200, 2021.

[16] M. Zhang, Y. Dou, P. H. J. Chong, H. C. B. Chan, and B.-C. Seet, "Fuzzy logic-based resource allocation algorithm for v2x communications in 5g cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2501–2513, 2021.

[17] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for d2d-enabled vehicular communications," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 3186–3197, 2017.

[18] "HiGHS - High-performance parallel linear optimization software - highs.dev," https://highs.dev/, [Accessed 30-04-2025].