A Game Theoretic Approach for Fair Channel Access in Cooperative Vehicle Safety Systems

FU-XIN ZHANG, GUO-ZHEN TAN, CHAO YU, NAN DING AND MING-JIAN LIU
School of Computer Science and Technology
Dalian University of Technology
Dalian, 116024, P.R. China
E-mail: zhangsw8881@mail.dlut.edu.cn
{tangz; cy496; dingnan; liumingjian}@dlut.edu.cn

Abstract: Cooperative vehicle safety systems rely on periodic broadcast of each vehicle’s state information to track neighbors’ positions and therefore to predict potential collisions. The most pressing challenge in such systems is to maintain real-time tracking accuracy while avoiding channel congestion. Previous work on congestion control in CVSSs primarily focuses on the effort of individual vehicle to adjust the contention window based on its local channel status, which may cause the available channel resource to be monopolized by a small number of vehicles. As a result, unfairness issue arises. To overcome this problem, in this paper, we present a game-based channel access approach for contention window adjustment in order to achieve fair channel access and high vehicle tracking accuracy for all vehicles. In this approach, a utility function that defines the relationship between vehicle tracking accuracy and throughput is firstly proposed. Then, based on the utility function, a non-cooperative dynamic game model with a penalty function is established. In order to implement the game model on each vehicle, a distributed channel access algorithm is then proposed. This algorithm adjusts the contention window for each vehicle, and hence, guides each vehicle to the Pareto-optimal Nash equilibrium. Simulation results confirm that the game-based channel access approach can guarantee a fair share of channel resources while achieving the optimal tracking performance for all vehicles under various traffic conditions.

Keywords: Cooperative vehicle safety systems, vehicular networking, channel congestion, fairness, channel access, vehicle tracking

1. INTRODUCTION

Vehicular networking, which is based on wireless communications between vehicles and with other infrastructures, enables a variety of new vehicular safety and automation applications. The most critical application deployed over vehicular networking is Cooperative Vehicle Safety Systems (CVSSs) [1]. CVSSs rely on tracking the positions and movements of neighboring vehicles to detect potential threats and provide warnings to the driver. To enable vehicle tracking among neighboring vehicles, each node (vehicle) periodically broadcasts state information (e.g., vehicle position, speed, and heading) over a wireless channel according to the IEEE 802.11p [2] and IEEE 1609 standards [3] defined under the Dedicated Short Range Communications (DSRC) framework. CVSSs

*This work is supported by High-Tech 863 Program (Grant No. 2012AA111902 and No. 2012AA112302) and the National Science Foundation of China (No.61471084 and No. 61502072), the Fundamental Research Funds for the Central Universities of China (No. DUT14RC(3)064), and Postdoctoral Science Foundation of China (No.2014M561229 and No. 2015T80251)
play a critical role in vehicle collision warning, and it has been shown that CVSSs can reduce over 75% of a nation’s crashes [4].

The performance of vehicle tracking accuracy is the basis for CVSSs [5]. The vehicle tracking accuracy, however, is significantly affected by the wireless channel in IEEE 802.11p. It is shown that even a medium vehicle density will bring about a high channel load [6]. The high channel load can cause increased amount of packet collisions in the shared channel. As a consequence, the nodes cannot receive the state information from their neighboring nodes. This results in position inaccuracies, which may adversely affect the performance of collision warning in CVSSs.

Controlling the contention window can reduce the amounts of packets collisions while ensuring the tracking accuracy for CVSSs. When the vehicle density is high, an increased contention window can decrease the probability that all nodes try to access the channel at the same time and therefore decrease the number of packet collisions; On the contrary, when the vehicle density is low, a decreased contention window can allow for a fast access to the channel for transmitting more up-to-date state information and thus gaining better tracking accuracy. Therefore, controlling contention window can be an effective mechanism to achieve a balance between packets collisions and vehicle tracking accuracy [8].

In the literature, several solutions [8] [9] [10] [11] [12] have been proposed to tackle channel congestion and vehicle tracking problems in CVSSs by controlling the contention window size. All these solutions, however, primarily rely on locally measured channel collision ratio to adjust the contention window for each node. Under this circumstance, the increase of contention window at a few nodes can lead to a reduced channel collision ratio observation from the perspective of neighboring nodes that have yet not increased their contention window sizes. As a result, those nodes would decrease their contention window and consequently monopolize the limited channel resource, causing unfairness to nodes that have already increased their contention window sizes. It has been shown in [13] that, for a network containing eight nodes delivering information, if one of the eight nodes selects a smaller contention window size, the throughput of other seven nodes is degraded by as much as fifty percent. As a result, these seven neighboring nodes cannot deliver their state information to their neighbors and consequently the tracking accuracy of these seven nodes will be degraded heavily.

The main cause of the unfairness is that these traditional solutions to vehicle tracking problems all focus on the effort of individual vehicle to adjust the contention window size based on its locally measured channel status. To overcome this disadvantage in current literature, in this paper, we propose a game-based channel access approach (GCAA) in order to realize fair access to the channel and at the same time achieve the optimal tracking accuracy for all nodes. Distinct from all the previous studies that rely on individual vehicle to adjust the contention window size, the proposed channel access approach uses a penalty mechanism and provides an incentive for vehicles to cooperate with each other to adjust the contention window around a fairness equilibrium point. The contributions of this work are as follows:

(1) A utility function that defines the relationship between vehicle tracking accuracy and throughput is proposed. The utility function reveals that the tracking error of each node exponentially decreases with the increase of throughput of each node. The utility function can be used to choose an appropriate network parameter, i.e., contention win-
dow size, to keep the tracking accuracy at a high level.

(2) Based on the utility function of tracking accuracy, a non-cooperative dynamic game model with a penalty function is designed. The penalty function in the dynamic game model guarantees that the nodes, who set their contention window to a lower value, are punished by their neighboring nodes so as to guide all nodes to a fair utilization of the channel resource.

(3) A distributed channel access algorithm that implements the non-cooperative dynamic game model on each node is designed. The distributed channel access algorithm can find the optimal contention window size for each node, and hence, guide each node to the Pareto-optimal Nash equilibrium. So, the optimal tracking accuracy for each node and fair share of channel access resource can be guaranteed.

This paper is organized as follows. Section 2 reviews related work in channel control approaches in vehicular networking. Section 3 describes the vehicle tracking problem in CVSSs and fundamentals in game theory. Section 4 presents a system model that identifies the utility function of tracking accuracy. Section 5 provides a channel access game model, and Section 6 proposes a distributed channel access algorithm. Section 7 gives the evaluation of the performance of the proposed channel access approach via simulation. Finally, Section 8 concludes this paper.

2. RELATED WORK

There have been many studies on congestion control for CVSSs. These studies can be categorized into two main different types according to the techniques they adopted. One technique is to control transmission rate or transmission power, and the other one is to adjust contention window size.

For the transmission rate control technique, C. L. Huang et al. [14] proposed an adaptive rate control approach to attain accurate tracking accuracy in CVSSs. The proposed approach relies on the estimated channel occupancy and the estimated position errors to determine whether to deliver vehicle tracking information or not. J. B. Kenney and G. Banal et al. [15] proposed a message rate adaptation approach based on a binary comparison between measured channel load and a target threshold. This approach uses linear feedback to adapt the message rate to ensure the efficient channel utilization. The work in [17] analyzed the effect of different choices of transmission rate and transmission power on the network performance and designed a feedback strategy for transmission power control in order to improve the vehicle tracking performance. M. Torrent-Moreno et al. [18] proposed a distributed fair transmission power control approach called D-FPAV. D-FPAV restricts the load on the shared channel and provides a fair share of channel resource.

For the contention window adjustment technique, one representative work is the one
by R. Stanica and E. Chaput [8], in which a new decremental back off mechanism for the MAC layer of 802.11p was designed. This back off mechanism controls the channel load by adjusting the contention window size dynamically, in order to find an equilibrium between collisions and expired packets. The proposed mechanism gains better quality of communications than the one used in the current version of the protocol, especially in scenarios with high vehicle density. To deal with the detection problem of packet collision, H. C. Jang and W. C. Feng [9] proposed a detection-based channel access approach. This approach adjusts contention window size based on channel status and the number of competing nodes. But the detection mechanism in [9] can cause delay, and therefore, is not suitable for real-time tracking information delivery in CVSSs. In order to improve the successful reception rate of vehicle tracking messages. J. Rezgui and S. Cherkaoui [10] proposed an adaptive channel access approach. This approach relies on the channel collisions ratio to adjust the contention window size. D. B. Rawat and D. C. Popescu et al. [11] proposed a feedback channel access approach to reduce the channel congestion. This approach uses the instantaneous collision rate in the shared channel to adjust the contention window. Another work in [12] designed a mathematical framework for three channel access schemes: probabilistic, deterministic and combined channel access schemes. The three channel access schemes are used for transmission control to improve the estimation accuracy of tracking application. The unified framework provides guidelines on how to design channel access schemes for CVSSs. The above approaches, however, all focus on individual vehicle to adjust the contention window based on locally observed channel information. Under this circumstance, a few nodes would increase their contention window sizes, leading to a reduced channel occupancy. Other nodes, however, may decrease their contention window sizes once they have found the channel occupancy is below a predefined threshold. As a result, the nodes that decrease their contention window sizes are exploiting the nodes that have already increased their contention window sizes. Consequently, an unfair situation is brought about between these two kinds of nodes.

3. PRELIMINARIES

3.1. Vehicle tracking problem

The vehicle tracking function blocks inside each vehicle are shown in Fig. 1. For
\( n \geq 2 \) nodes, each node \( i, \ i \in \{1, 2, ..., n\} \), has a plant and a bank of neighbor estimators. The plant describes the state of node \( i \) and the neighbor estimators operate simple kinematics models for estimating the state of the neighbors. Let \( \tilde{x}_p(t) \) be the receiver \( j \)'s estimation of sender \( i \)'s position and \( \tilde{v}_p(t) \) be the receiver \( j \)'s estimation of sender \( i \)'s speed. The kinematics model is formulated as a simple discrete equation, and can switch between the following two modes [4]:

1. If no state information of node \( i \) is received at time \( t \) by receiver \( j \), the estimated state of node \( i \) at time \( t-1 \) is used to estimate the state of node \( i \) at time \( t \),
   \[
   \tilde{x}_p(t) = \tilde{x}_p(t-1) + \tilde{v}_p(t-1) \Delta t,
   \]
   \[
   \tilde{v}_p(t) = \tilde{v}_p(t-1)
   \]
   where \( \Delta t \) is the time interval in-between message transmission.

2. Else if the state information of node \( i \) is received at \( t \), the new information is used to reset estimation state of node \( i \),
   \[
   \tilde{x}_p(t) = x_i(t),
   \]
   \[
   \tilde{v}_p(t) = v_i(t)
   \]
   where \( x_i(t) \) is the position of node \( i \) and \( v_i(t) \) is the speed of node \( i \). Thus, the position error between the actual position and the estimated position of sender \( i \) at receiver \( j \) is calculated as follows:
   \[
   \epsilon_p(t) = x_i(t) - \tilde{x}_p(t).
   \]

The objective of our channel access approach is to minimize vehicle tracking error \( \epsilon_p(t) \) (i.e., tracking accuracy) for each node.

### 3.2 Game theory

A game consists of three components: a set of players, a set of strategies, and a set of utility functions that map strategies into the real numbers. In a game, each player selects a strategy with the objective of maximizing its utility [19] [20] [21].

1. **The formulation of game model**

   A game can be modeled as \( G = (P, A, U, R) \), where:
   - \( P = \{1, ..., n\} \) denotes the set of players.
   - \( S = S_1 \times S_2 \times ... \times S_n \) denotes the strategies space of all players, where \( S_i \) denotes strategy space of player \( i \). In the strategies space of all players, \( s \in S \) denotes a strategy profile of all players.
   - \( U = \{u_1, u_2, ..., u_n\} \) denotes the set of utility functions, where \( u_i : S \rightarrow R \) denotes player \( i \)'s utility function.
   - \( R \) denotes the set of payoff, which is in the form of real numbers.
(2) Nash equilibrium

A Nash equilibrium is a strategy profile at which no player has anything to gain by changing only its own strategy unilaterally. A Nash equilibrium is a stable point because no player can benefit by changing his strategy. Hence, no users have incentive to change his strategy at Nash equilibrium point. Formally, a Nash equilibrium is a strategy profile \( \mathbf{s} \) such that for all \( s_i \in S_i \),

\[
    u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})
\]

where \( s_i \in S_i \) denotes the player \( i \)'s strategy, \( s_{-i} \) denotes the strategies of the other \((n-1)\) players, \( u_i(s_i, s_{-i}) \) is the payoff of player \( i \) when player \( i \) selects an strategy \( s_i \) from its strategy space \( S_i \), and all the players except \( i \) select strategies \( s_{-i} \) from their own strategies space.

(3) Pareto-optimal point

Pareto-optimal point is another important concept for game theory. The Pareto-optimal point is a measure that it is impossible to make the utility of at least one node better off without making the utility of others worse off. Moreover, the Pareto-optimal point is the point in which the utility of each player is maximized simultaneously. Formally, a strategy profile \( s^* \in S \) is said to be Pareto-optimal if there is no strategy profile \( s \in S \) such that for all player \( i \),

\[
    u_i(s) \geq u_i(s^*)
\]

where \( u_i(s) \) is the payoff of player \( i \) when all players select a strategy profile \( s \), and \( u_i(s^*) \) is the payoff of player \( i \) when all players select a strategy profile \( s^* \). This means that a strategy profile is Pareto-optimal if it is impossible to improve the payoff of any player without harming other players.

4. SYSTEM MODEL

This section first presents the utility function of tracking accuracy with respect to throughput, and then, formulates the utility function of vehicle tracking accuracy regarding channel access probability.

4.1. Throughput-dependent utility function for vehicle tracking accuracy

The main goal of CVSSs is to guarantee an accurate tracking accuracy for each node. The vehicle tracking accuracy is the application metric of CVSSs, which can be seen as a utility of a vehicle. In CVSSs, the tracking accuracy is described by the tracking error. The lower tracking error stands for an improved utility of a vehicle. Based on the utility-based approach [22], vehicle tracking accuracy can be described by a utility function, which expresses the relationship between tracking accuracy and network parameters.
However, as the utility is a more application-specific or subjective measure, it is hard to provide an accurate quantitative relationship between an application-specific parameter and a network-level parameter [23]. The exponential relationship hypothesis is a generic formula that describes the interdependency of application-specific parameters and network-level parameters [24]. In the exponential formula, the change of application parameters depends on the current level of application performance, given the same account of change of network-level parameters. When the application performance is high, a small change of network-level parameters will heavily affect the application parameters. On the other hand, when the application performance is already low, a further improvement of network-level parameter is not perceived significantly. That is, the exponential formula relates changes of application parameters with respect to network-level parameters to the current level of application performance [24].

In CVSSs, the required tracking accuracy from the vehicle tracking process mainly relies on how many packets of a sending node are successfully received by receiving nodes per second (i.e., throughput) in order to make an accurate vehicle state estimation [1]. As a consequence, the utility of vehicle tracking accuracy can be defined as the function of throughput parameters. The tracking error drops quickly as throughput increases when the tracking error is high. Moreover, the performance of tracking accuracy does not improve significantly after a certain throughput, at which the tracking error is already very low [1]. In other words, the change of vehicle tracking accuracy highly depends on the current level of tracking accuracy, given the same amount of change of the throughput value. Therefore, the exponential relationship hypothesis is well appropriate for modeling the relationship between the tracking accuracy and the throughput in CVSSs. Against this background, based on the exponential relationship hypothesis, the equation that depicts the relationship between the change of tracking accuracy and the change of throughput can be defined as follows:

$$ \frac{\partial U_i(r_i)}{\partial r_i} = -\alpha \cdot U_i(r_i) $$

(4)

where $U_i(r_i), \; i \in \{1, 2, ..., n\}$, is the utility function of tracking accuracy of node $i$ with respect to throughput, $r_i$ is the throughput of node $i$, and $\alpha$ is a constant.

The solution to Eq. (4) is an exponential function, which depicts the relationship between tracking accuracy and throughput as follows:

$$ U_i(r_i) = v_i \cdot e^{-\lambda_i r_i} + \chi $$

(5)

where $U_i(r_i)$ is the utility function of tracking accuracy of node $i$, $r_i$ is the throughput of node $i$, $\lambda_i$ represents how sensitive the tracking accuracy is to the amount of the throughput, and $v_i$ and $\chi$ are the coefficient.

To validate the utility function of tracking accuracy of Eq. (5), and retrieve the corresponding optimal fitting utility function, an empirical method [25] is used to build an empirical model of tracking accuracy with respect to throughput. Based on the empirical method, we first conduct a large number of simulations to obtain a large set of simulation traces, and then, we apply the technique of linear least squares curve fitting to each trace.
to derive the optimal empirical model of tracking accuracy with respect to throughput.

We use traffic simulator VISSIM [26] and network simulator NS-3 [27] to conduct the simulations. The simulation scenario in VISSIM is a straight 1-km of a 4-lane highway. The generated vehicle trajectories are then fed to NS3 for network simulations. In each simulation, the speed of vehicles follow a uniform distribution between $v_{\text{min}}$ and $v_{\text{max}}$ with mean $\mu=(v_{\text{min}}+v_{\text{max}})/2$ and variance $\sigma^2=(v_{\text{max}}-v_{\text{min}})^2/2$. We set $v_{\text{min}}=20\text{m/s}$ and $v_{\text{max}}=30\text{m/s}$, that is, the vehicles’ speed ranges from 20m/s–30m/s, which is typical for highways [28]. According to [13], the vehicles broadcast their state messages to a distance of 150m, which can satisfy most safety applications. Therefore, the neighboring nodes of a vehicle are defined as the nodes in the circular area of a 150m radius. We calculate the position errors over all neighboring nodes according to Eq. (3).

Experiments have been repeated by varying the packet transmission rate of each node. All the nodes use the same transmission rate and adjust it in synchronization with other nodes. We use the measure of average tracking error as the main performance metric for CVSSs simulations. At the same time, the average throughput of each node is calculated and recorded. The results are averaged over 10 simulation runs.

Fig. 2 plots the tracking errors obtained by each node with the throughput of each node. Each node represents a single tracking error for a given throughput. We observe that after a certain throughput of 20pkts/s, the tracking error is already very low and close to its minimum value. To retrieve the optimal fitting theoretical function of Eq. (5), we use the linear least squares curve fitting to each trace, and find an empirical model such that the normal error of tracking error $E$ is minimized. According to [24], the normal error $E_i$ of node $i$ ($i \in \{1, 2, ..., n\}$) is defined as the sum of the residuals $s_i$ between tracking accuracy from theoretical function and actual tracking accuracy from the simulation for all measured throughput $r_i$. The normal error $E_i$ of node $i$ is given as follows:

$$E = \sum_{i=1}^{n} s_i, s_i = U_i(r_i) - y_i$$

(6)

where $r_i$ is the measured throughput $r_i$, $U_i(r_i)$ is the tracking error of node $i$ from theoretical function (5), and $y_i$ is the tracking error of node $i$ from the simulation.

Based on the linear least squares curve fitting technique, we obtain the following empirical model of tracking accuracy with respect to throughput:

$$U_i(r_i) = 10t \cdot e^{-0.1045r_i}$$

(7)

where $U_i(r_i), i \in \{1, 2, ..., n\}$, is the tracking accuracy of node $i$ with respect to throughput, $(0, t)$ is the vehicle’s travel time, $10t$ is the maximum tracking error of node $i$ during $(0, t)$, 0.1045 represents how sensitive the tracking accuracy is to the amount of the throughput, and $r_i$ is the throughput of node $i$. 
The fitness of the empirical model expressed by Eq. (7) can be measured by the coefficient of correlation $R$, the coefficient of determination $R^2$, the mean square error ($MSE$) and the normalized mean squared error ($NMSE$). Both $R$ and $R^2$ should approach one if the empirical model and the measured data match perfectly. Meanwhile, the $MSE$ and $NMSE$ should approach zero [24]. For the fitness of Eq. (7), $R=0.998$, $R^2=0.996$, $MSE=0.03028$ and $NMSE=0.005$. All these metrics indicate an almost perfect match between the fitting empirical model of tracking accuracy and actual measured data.

Therefore, we can see that the utility function, i.e., Eq. (5), is a perfect match to the actual measured data given by Eq. (7), with $\nu = 10r$, $\lambda = 0.1045$ and $\chi = 0$. This proves that Eq. (7) is an optimal theoretical function that can represent the relationship between vehicle tracking accuracy and throughput.

In addition, as can be seen from Fig. 2, when the tracking accuracy is low, improving throughput further is not helpful to achieve a higher tracking accuracy. That is, allocating more channel resources to a node is useless to improve its tracking accuracy. In this case, it would be useful to allocate the channel resources to other nodes with low tracking accuracy so as to realize the fair share of channel resources.

4.2 The utility function of vehicle tracking accuracy regarding channel access probability

In CVSSs, vehicles employ a CSMA/CA MAC protocol to broadcast their state information. In CSMA/CA MAC protocol, the channel access probability $\tau_i$ of node $i$ in a time slot is determined by its contention window size $W_i$ as follows [29]:

$$
\tau_i = \frac{2}{1 + W_i}.
$$

(8)
The probability that node \( i \), \( i \in \{1, 2, ..., n\} \), successfully transmits a packet during a slot time is [29]:

\[
P_i^s = \tau_i \prod_{j \neq i} (1 - \tau_j).
\]  

(9)

The probability of the channel being idle is [29]:

\[
P_i = \prod_k (1 - \tau_k)
\]  

(10)

where \( k \in \{1, 2, ..., n\} \).

The throughput, which is the average number of packets transmitted in a slot time by node \( i \), can be computed as follows [29]:

\[
r_i = \frac{P_i^s}{P_i^s + P_i^c + P_i^c_T}
\]  

(11)

where \( P_i^s = \sum_k P_k^s \) is the probability that all nodes within communication range successfully transmit packets and \( P_k^s \) is the probability that node \( k \) successfully transmits a packet from Eq. (9), \( P_i^c = 1 - P_i - P_i^s \) is the probability of packets collision, \( T_s \) is the duration that the channel is busy for transmitting a packet, \( T_i \) is the duration of the idle period, and \( T_c \) is the average time that the channel is busy during a collision.

After some algebraic manipulations of Eq. (11), the throughput of node \( i \) can be expressed as follows:

\[
r_i(\tau_i) = \frac{\tau_i c_i^l}{\tau_i c_i^l + c_i^c}
\]  

(12)

where \( c_i^l = \alpha_i c_i^l \), \( c_i^c = \alpha_i (T_s - T_i) - s_i (T_i - T_s) \), \( c_i^l = (1 - \alpha_i - s_i) T_i + s_i T_s + \alpha_i T_c \), in which \( \alpha_i = \prod_j (1 - \tau_j) \) and \( s_i = \sum_{j \neq i} \tau_j \prod_k (1 - \tau_k) \).

Combing Eq. (5), (7) and (12), the utility function that describes the relationship between vehicle tracking accuracy and channel access probability is given by:

\[
U_i(\tau_i) = v_i \cdot e^{-\lambda_i(\tau_i)}
\]  

(13)

where \( v_i = 10 r \), in which \( (0, r) \) is the vehicle's travel time in a road, and \( \lambda_i = 0.1045 \).

From utility function (13), we can see that the only parameter that a node can control is its own channel access probability. Moreover, utility function (13) is a strictly decreasing function with respect to the channel access probability. A higher channel access probability of a node can cause channel resource monopolized by this node, leading to an unfair utilization of channel recourses and hence, degrading the tracking accuracy of other nodes. In next section, we propose a game-theory-based channel access approach to resolve this problem.
5. CHANNEL ACCESS GAME MODEL

In this section, a channel access game model for vehicle tracking problem is developed in Subsection 5.1. Then, in Subsection 5.2, the existence of the Nash equilibrium of the game model is proved.

5.1. Channel access game formulation

Based on the non-cooperative dynamic game model introduced in [30], we propose a new game model in which nodes make their decisions based on vehicle tracking errors. In the non-cooperative dynamic game model, each node maximizes its own benefit and makes their decisions based on previous actions and system states [19]. We assume that each node is rational and aims to maximize its own tracking accuracy.

Let \( \Phi \) denote the set of all nodes within communication range of each other. In the proposed game model, the strategy of each node \( i \) in the game model is to set a channel access probability such that the utility function of vehicle tracking accuracy is maximized. In fact, the channel access probability and contention window size are interchangeable based on Eq. (8). By varying contention window size \( W_i \), node \( i \) can change his channel access probability. We assume that the nodes can directly adjust the channel access probability in the game model. A node’s utility function \( J_i \) is redefined as:

\[
J_i = U_i(\tau_i) + P_i(\tau_i)
\]

where \( U_i(\tau_i) = v_i e^{-\lambda_i(\tau_i)} \) is defined in Eq. (13), and \( P_i(\tau_i) \) denotes a penalty function defined as follows.

\[
P_i(\tau_i) = k_i (\tau_i - \overline{\tau}), k_i \geq 0, 0 < \overline{\tau} < 1
\]

where \( k_i \) is a constant and \( \overline{\tau} \) is the Nash equilibrium point. The Nash equilibrium point \( \overline{\tau} \) is the point at which each node achieves the optimal tracking accuracy. Although the penalty function defined in Eq. (15) has the same form with the one defined in [30], they are different in essence. The penalty function given by Eq. (14) is based on vehicle tracking accuracy, rather than network performance, i.e., throughput, in the penalty function defined in [30]. The penalty function guarantees that deviating nodes, which have lower tracking errors than the optimal threshold value, are punished by other neighbors with larger tracking errors. This will leave the channel resources monopolized by deviating nodes to other nodes that have higher tracking errors. In this way, each node participating in the game will cooperate and eventually reach Nash equilibrium, in which each node achieves its optimal tracking accuracy while realizing a fair share of channel resources.

For tracking applications in CVSSs, a lower tracking error will bring about a higher utility. Combining Eq. (13), (14) and (15), a dynamic game model is defined as follows:

\[
\min_{0 < \tau_i \leq 1} J_i = v_i e^{-\lambda_i(\tau_i)} + k_i (\tau_i - \overline{\tau}) \quad \forall i \in \Phi.
\]
In order to solve the minimization problem (16), we define a Lagrangian function $Z(\tau, \lambda)$ for the minimization problem (16) and transform the constrained minimization problem into an unconstrained minimization optimization problem. The Lagrangian function for the minimization problem (16) is given as follows:

$$Z(\tau, \eta) = J_i + \eta_i (1 - \tau_i)$$

where $\eta_i \geq 0$ is a Lagrangian multiplier.

The utility function $J_i$ is quasi-concave with respect to the channel access probability and the proof will be given in Subsection 5.2. Therefore, the Karush-Kuhn-Tucker condition [31] is met. According to the Karush-Kuhn-Tucker condition, the sufficient and necessary conditions for $J_i$ to be minimized is:

$$\begin{cases}
\frac{\partial Z(\tau_i, \eta_i)}{\partial \tau_i} - \eta_i \geq 0, \tau_i \geq 0 \quad \text{and} \quad \tau_i \frac{\partial Z(\tau_i, \eta_i)}{\partial \tau_i} - \eta_i = 0 \\
\frac{\partial Z(\tau_i, \eta_i)}{\partial \eta_i} \leq 0, \eta_i \geq 0 \quad \text{and} \quad \eta_i (\tau - \tau_i) = 0
\end{cases}$$

Solving Eq. (18), we obtain the following optimality conditions for node $i$:

(a) $\tau_i = 1$ if $\frac{\partial J_i}{\partial \tau_i} \leq 0$; (b) $\tau_i \in (0, 1)$ if $\frac{\partial J_i}{\partial \tau_i} = 0$; (c) $\tau_i = 0$ if $\frac{\partial J_i}{\partial \tau_i} \geq 0$.

Let $\tau_i^*$ denote the solution of the Equation $\frac{\partial J_i}{\partial \tau_i} = 0$. Substituting $\tau_i^*$ into equation $\frac{\partial J_i}{\partial \tau_i} = 0$, we can obtain:

$$-\nu_i \lambda e^{-\lambda(\tau_i^*)} \frac{c_i' c_3'}{(\tau_i^* c_2' + c_3')^2} - k_i = 0.$$  

Thus, the node’s utility function in the game model (16) becomes as follows:

$$k_i = -\nu_i \lambda e^{-\lambda(\tau_i^*)} \frac{c_i' c_3'}{(\tau_i^* c_2' + c_3')^2}.$$
A GAME THEORETIC APPROACH FOR FAIR CHANNEL ACCESS IN COOPERATIVE VEHICLE SAFETY SYSTEMS

\[ J_i = v_i e^{-\lambda_i(t_i)} + v_\lambda e^{-\lambda_i(t)} \frac{c_i^j c_i^j}{(\tau c_i^j + c_i^j)^2} (\tau_i - \tau). \]

Based on Eq. (21), the first issue is how to lead the nodes to the Nash equilibrium \( \tau_i = \tau \), \( \forall i \in \Phi \). In the remaining section, we first prove the existence of the Nash equilibrium point, and then propose a method to guide the nodes to the Nash equilibrium.

5.2. Derivation of the Nash equilibrium

We suppose that the utility function of node \( i \) in the game model is \( J_i(\tau_i, \tau_{-i}) \), where \( \tau_{-i} \) is the channel access probability of all the other nodes excluding node \( i \). The Nash equilibrium means that a node would not get a higher utility if it deviates from the equilibrium point, i.e., the channel access probability should satisfy:

\[ J_i(\tau^*_i, \tau_{-i}) \leq J_i(\tau_i, \tau_{-i}) \]

where \( \tau^*_i \) is Nash equilibrium point.

We want to find the Nash equilibrium point for the game. The existence of a Nash equilibrium should meet the following two conditions [32]: (1) The strategy space is a non-empty, closed and bounded convex subset of the Euclidean space; and (2) The \( J_i(\tau_i, \tau_{-i}) \) is continuous and quasi-concave with respect to the channel access probability.

**Theorem 1** There exists a Nash equilibrium in the channel access game

**Proof.** The strategy space \( 0 < \tau_i \leq 1 \) is non-empty, closed and bounded convex. Moreover, from Eq. (14) and (15), it can be concluded that the utility function of tracking accuracy and penalty function \( P_\tau(\tau_i) \) are all continuous with the node’s strategy space \( \tau_i \). Thus the utility function \( J_i(\tau_i, \tau_{-i}) \) redefined in game (16) is continuous. To verify the exit of the Nash equilibrium, the first partial derivative of \( J_i(\tau_i, \tau_{-i}) \) is given by:

\[ \frac{\partial J_i(\tau_i, \tau_{-i})}{\partial \tau_i} = -v_i \lambda e^{-\lambda_i(\tau_i)} \frac{c_i^j c_i^j}{(\tau c_i^j + c_i^j)^2} - k_i(t) \]

where \( k_i(\tau_i) = \frac{\tau_i c_i^j}{\tau_i c_i^j + c_i^j} \).

The second partial derivative of \( J_i \) is given by:

\[ \frac{\partial^2 J_i(\tau_i, \tau_{-i})}{\partial \tau_i^2} = v_i \lambda^2 e^{-\lambda_i(\tau_i)} \left( \frac{\partial k_i(\tau_i)}{\partial \tau_i} \right)^2 - v_i \lambda^2 e^{-\lambda_i(\tau_i)} \frac{\partial^2 k_i(\tau_i)}{\partial \tau_i^2}. \]

Since \( \frac{\partial^2 k_i(\tau_i)}{\partial \tau_i^2} = \frac{-2c_i^j c_i^j c_i^j}{(\tau_i c_i^j + c_i^j)^3} < 0 \), the second partial derivative of the utility function
\[
\frac{\partial^2 J_i(\tau_i, \tau_{-i})}{\partial \tau_i^2} > 0 \quad \text{for } \tau_i \in [0, 1].
\]
Thus, the utility function \( J_i(\tau_i, \tau_{-i}) \) is quasi-concave with respect to the channel access probability, which proves the exit of the Nash equilibrium together with convex strategy space \( 0 < \tau < 1 \).

In CVSSs, the limited channel source is shared by multiple nodes within the communication range. A node obtaining a high utility would be at the expense of preventing other vehicles from obtaining high tracking accuracy. Thus, the Nash equilibrium point \( \tau \) in the game model (16) should be Pareto-optimal. To guide each node to the Nash equilibrium, and also to a Pareto-optimal point, in the following section, we first design a method that makes each node converge to the Nash equilibrium point, and then propose a method that makes the Nash equilibrium Pareto-optimal.

### 6. DISTRIBUTED CHANNEL ACCESS ALGORITHM

In this section, a method that leads the nodes to the Nash equilibrium point is designed. Then, a distributed channel access algorithm (DCAA) is proposed to guide the nodes to reach the Pareto-optimal point.

#### 6.1. Reaching a Nash equilibrium point

From Eq. (14), we can see that at the Nash equilibrium point \( \tau_i = \tau \), the following holds: \( J_i = U_i(\tau_i) = v_i e^{-\lambda_i(\tau_i)} \), which indicates that the node achieves the optimal tracking accuracy when \( \tau_i = \tau \). However, when \( P_i(\tau_i) < 0 \), i.e., \( \tau_i < \tau \), there is no need to inflict penalty on node \( i \). Node \( i \) will continue to increase the channel access probability to achieve the optimal tracking accuracy until \( \tau_i = \tau \). Thus, the redefinition of the node \( i \)'s utility function in the game model can be:

\[
J_i = v_i (1 - e^{-\lambda_i(\tau_i)}) + v_i \lambda_i e^{-\lambda_i(\tau_i)} \frac{c_i' c_j'}{c_i' c_j' + c_j' c_i'} \times \begin{cases} 
\tau_i - \tau, & \tau_i > \tau \\
0, & \tau_i < \tau
\end{cases}.
\tag{25}
\]

Given the game model (25), a method is designed to lead each node to the Nash equilibrium. According to [30], we use the node that has the lowest channel access probability to guide other nodes to the Nash equilibrium. Therefore, we define:

\[
\tau \triangleq \min_{\tau_i \in \Theta_i} \tau_i
\tag{26}
\]
in which the penalty function \( P_i(t) \) in utility function (25) can be seen as the penalty that the node with the lowest channel access probability inflicts on node \( i \).

Let \( k \) denote a virtual node such that \( \tau_k = \tau \), where \( \tau \) is defined by Eq. (26). As \( \frac{\partial (U_i)}{\partial \tau_k} < 0 \) and \( \frac{\partial (r_k(\tau_k))}{\partial \tau_j} < 0 \), we can obtain:
\[
\frac{\partial U_k(\tau_k)}{\partial \tau_i} = \frac{\partial (U_k)}{\partial r_k} \frac{\partial (r_k)}{\partial \tau_i} > 0. \tag{27}
\]

Based on the property of \( \frac{\partial U_k(\tau_k)}{\partial \tau_i} > 0 \), we can derive the following optimization problem with respect to \( \tau_i (\tau \leq \tau_i \leq 1) \) for each node \( i \neq k \):

\[
\begin{align*}
\min & (U_k(\tau_i, \tau_{-k})) \\
\Omega_1: & \; s.t. \; \tau \leq \tau_i \leq 1, \forall i \in \Phi \setminus \{k\} \\
& \; \tau_k = \bar{\tau}.
\end{align*} \tag{28}
\]

Since \( \frac{\partial U_k(\tau_k)}{\partial \tau_i} > 0 \), only by decreasing the channel access probability of node \( i \), the minimum tracking performance of node \( k \) can be obtained. Thus, the optimization problem \( \Omega_1 \) guarantees that the node who has a higher channel access probability would be punished by the node with a lowest channel access probability, so as to guide all nodes to reach a Nash equilibrium point.

Next, we show how the node \( k \) would guide the other nodes to the Nash equilibrium point.

It can be found that the optimization problem \( \min(U_k(\tau_i, \tau_{-k})) \) with respect to \( \tau_i (\tau \leq \tau_i \leq 1) \) in Eq. (28) is equivalent to \( \max(\tau_i (\tau_i, \tau_{-k})) \). This is because \( U_k(\tau_i, \tau_{-k}) \) and \( r_k(\tau_i, \tau_{-k}) \) exhibits opposite behaviors in \( \tau_i (i \neq k) \), i.e., \( U_k(\tau_i, \tau_{-k}) \) is monotone increasing in \( \tau_i (\tau_i < 1, i \neq k) \) while \( r_k(\tau_i, \tau_{-k}) \) is monotone decreasing in \( \tau_i (\tau_i < 1, i \neq k) \). Thus, the optimal channel access probability \( \tau_i \) for the optimization problem \( \min(U_k(\tau_i, \tau_{-k})) \) is also optimal for the optimization problem \( \max(r_k(\tau_i, \tau_{-k})) \).

Moreover, since \( \tau_k \) is equal to \( \bar{\tau} \) and \( c_2^k \), \( c_3^k \) are constant,

\[
\begin{align*}
r_k(\tau_i, \tau_{-k}) &= \frac{\tau_i}{
\prod_{\nu \in \Phi \setminus \{k\}} (1 - \tau_{\nu})
\} \frac{1}{\tau_k c_2^k + c_3^k}
\end{align*}
\]

is equivalent to \( \prod_{\nu \in \Phi \setminus \{k\}} (1 - \tau_{\nu}) \). As a consequence, \( \min(U_k(\tau_i, \tau_{-k})) \) is equivalent to \( \max \prod_{\nu \in \Phi \setminus \{k\}} (1 - \tau_{\nu}) \). By taking the logarithm of \( \prod_{\nu \in \Phi \setminus \{k\}} (1 - \tau_{\nu}) \), the optimization problem \( \Omega_1 \) can be transformed into the following equivalent optimization problem:

\[
\begin{align*}
\max & \sum_{i \in \Phi \setminus \{k\}} \log(1 - \tau_i) \\
\Omega_2: & \; s.t. \; \tau \leq \tau_i \leq 1, \forall i \in \Phi \setminus \{k\}.
\end{align*} \tag{29}
\]

The Lagrangian formulation of \( \Omega_2 \) is given as follows:

\[
\begin{align*}
Z(\tau_{-k}, \sigma_j, \omega_k) &= \sum_{i \in \Phi \setminus \{k\}} z(\tau, \sigma_j, \omega_k) \\
&= \sum_{i \in \Phi \setminus \{k\}} \log(1 - \tau_i) + \sigma_j (\bar{\tau} - \tau_i) + \omega_k (\bar{\tau} - 1) \tag{30}
\end{align*}
\]
where $\sigma_i > 0$ and $\omega_i > 0$ are the Lagrangian multipliers.

According to the Kuhn-Tucker first order necessary conditions, we have $\omega_i (\tau_i - 1) = 0$. Since $\tau_i < 1$ and $\omega_i \geq 0$, $\omega_i (\tau_i - 1) = 0$ implies $\omega_i = 0$. Thus, the Lagrangian formulation of $\Omega_2$ in Eq. (31) can be simplified as follows:

$$Z(\tau_{-i}, \sigma_i) = \sum_{i \in \Phi \setminus \{k\}} z(\tau_i, \sigma_i)$$

s.t. $z(\tau_i, \sigma_i) = \log(1 - \tau_i) + \sigma_i(\bar{z} - \tau_i)$. (31)

Since $z(\tau_i, \sigma_i)$ in Eq. (31) is strictly concave and twice continuously differentiable, the Kuhn-Tucker first order necessary condition is also the first order sufficient condition for $z(\tau_i, \sigma_i)$, i.e., $\frac{\partial z(\tau_i, \sigma_i)}{\partial \tau_i} = 0$. Let $\tau_i^{opt}$ denote the solution of the Equation

$$\frac{\partial z(\tau_i, \sigma_i)}{\partial \tau_i} = 0.$$ 

Thus, we have:

$$\tau_i^{opt} = 1 - \frac{1}{\tau_i}.$$ (32)

In order to design a method that leads the nodes to the Nash equilibrium, we transform the optimization problem $\Omega_2$ into a dual problem based on the Lagrangian formulation of $\Omega_2$. The dual problem of $\Omega_2$ is given as follows:

$$\Omega_3: \min_{\lambda_r \geq 0} Z(\tau_i^{r}, \sigma_i).$$ (33)

Based on the steepest descent method, the optimization problem $\Omega_3$ is solved by using the following iterative algorithm:

$$\sigma_i(r+1) = \sigma_i(r) - \rho \frac{\partial Z(\tau_i^{r}, \sigma_i)}{\partial \sigma_i} = \sigma_i(r) - \rho(\tau_i^{opt}(r) - \bar{z}), \forall i \in \Phi \setminus \{k\}$$ (34)

where $\sigma_i(r)$ is the size of Lagrangian multipliers at iteration $r$, $\tau_i^{opt}(r)$ is the channel access probability of node $i$ at iteration $r$, and $\rho > 0$ is a step size.

The iterative algorithm works as follows. Whenever $\tau_i^{opt}(r) > \bar{z}$ or $\tau_i^{opt}(r) < \bar{z}$ at iteration $r$, $\sigma_i(r)$ is updated to $\sigma_i(r+1)$ by Eq. (34) and then, $\sigma_i(r+1)$ is informed to node $i$. Having received $\sigma_i(r+1)$, node $i$ recalculates its optimal channel access probability $\tau_i^{opt}(r+1)$ and then moves to the next iterative computation.

Using Eq. (32) and (34), each node is guided to the Nash equilibrium point $\bar{z}$. This Nash equilibrium, however, may not be the optimal point, i.e., the Pareto-optimal point. At Pareto-optimal point, the utility of each node is maximized simultaneously. In order to achieve the high tracking accuracy for each node, the Nash equilibrium point must be a Pareto-optimal point. In next section, we design a distributed channel access algorithm,
in order to guide each node to the Pareto-optimal Nash equilibrium point.

6.2. Achieving the Pareto-optimal Nash point.

The penalty function guides each node to the Nash equilibrium point \( \tau \), by running Eq. (32) and Eq. (34). In order to make the Nash equilibrium Pareto-optimal, a distributed channel access algorithm (DCAA) is designed. This algorithm is implemented on each node to guide each node to the Pareto-optimal Nash equilibrium. The pseudo-code of DCAA is shown in Algorithm 1. The general idea is that, initially, each node \( i \in \Phi \), sets its contention window \( W_i(0) = 15 \). \( W_i(0) = 15 \) is the minimum contention window size defined in IEEE 802.11p for periodical broadcasting of state information. Node \( k \) increases its contention window size by a small value \( \delta \). This will in turn trigger the penalizing mechanism of node \( k \) and guide the other nodes to a new Nash equilibrium point, as they also run Eq. (32) and Eq. (34). At the new equilibrium point, each node will compare its current utility (i.e., the value of utility function in the game model (16)) to the utility that is achieved at the last Nash equilibrium point. If the current utility is very close to the last utility, the iteration process of algorithm will be ended and all nodes reach the Pareto-optimal Nash point. Otherwise, the nodes will repeat the above iterative process until they find the Pareto-optimal Nash point.

**Algorithm 1:** Distributed channel access algorithm

1. **Initialize** the contention window size for each node \( i \in \Phi \), \( W_i(0) = 15 \), and set iteration counters \( s \) and \( r \) to 1, set computing accuracy \( \epsilon_1 < 0.001 \), \( \epsilon_2 < 0.01 \);
2. **for** each iteration \( s \) **do**
3.  **Select** a node \( k \in \Phi \) and define a new contention window size for node \( k \), \( W_k(s) = W_k(s-1) + \delta \)
4.  **for** each node \( i \in \Phi \backslash \{k\} \) **do**
5.   **Update** \( \sigma_i(s, r) \) according to Eq. (34), i.e.,
   \[ \sigma_i(s, r) = \sigma_i(s, r-1) - \rho (\tau_i^{\text{opt}}(s, r-1) - \tau_i) \]
   where \( \sigma_i(s, r) \) is the size of Lagrangian multipliers at iteration \( s \) and \( r \), and \( \tau_i^{\text{opt}}(s, r-1) \) is the channel access probability of node \( i \) at iteration \( s \) and \( r-1 \);
6.   **Communicate** \( \sigma_i(s, r) \) to node \( i \) and calculate the optimal channel access portability \( \tau_i^{\text{opt}} \) for each node \( i \) according to Eq. (32), i.e.,
   \[ \tau_i^{\text{opt}}(s, r) = 1 - \frac{1}{\sigma_i(s, r)} \]
7.  **Move** to the next iteration \( r+1 \) and continues with step 5, unless
   \[ |\tau_i^{\text{opt}}(s, r) - \tau_i^{\text{opt}}(s, r-1)| < \epsilon_1 ; \]
8. **end for**
9. **Compute** the contention window size \( W_i(s) \) for each node \( i \) according to Eq. (8);
10. **Move** to the next iteration \( s+1 \) and continues with step 3, unless
\[ |j_i(s) - j_i(s-1)| < \epsilon_i, \] where \( j_i(s) \) is the value of the utility function in the game model (16) at iteration \( s \).

11 end for

7. PERFORMANCE EVALUATION

This section first introduces simulation settings, and then presents simulation results.

7.1. Simulation settings

To test the performance of our proposed channel access approach, we conduct traffic/network simulation experiments for CVSSs using VISSIM and NS3 simulators for road traffic and network simulations, respectively. The main simulation scenario in VISSIM is a straight 1-km of a 4-lane highway. In each simulation, the speed of each vehicle follows a uniform distribution between \( v_{\text{min}} \) and \( v_{\text{max}} \) with mean \( \mu = (v_{\text{min}} + v_{\text{max}})/2 \) and variance \( \sigma^2 = (v_{\text{min}} - v_{\text{max}})^2/2 \). We set \( v_{\text{min}} = 20 \text{m/s} \) and \( v_{\text{max}} = 30 \text{m/s} \), which is typical for highways [28]. Vehicle trajectories are then fed to NS3 in which vehicles communicate over a DSRC channel by following the DSRC Rayleigh distributions model in [33]. Table 1 lists the simulation parameters used in the experiments. We use a packet generation rate of 20 packets/s, which is an acceptable value to provide accurate enough state information of vehicles to CVSSs [17]. Therefore, during this simulation, we sample at 20Hz to collect the vehicle state information, such as vehicle position, speed and heading. At each 50ms time step, each node generates a packet and broadcasts it to its neighbors. Upon receiving the vehicle state information from the shared channel, each vehicle resets its estimation of vehicles within its communication range.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>5.9GHz</td>
</tr>
<tr>
<td>Modulation and Data rate</td>
<td>BPSK, 3Mbps</td>
</tr>
<tr>
<td>Transmission power</td>
<td>25dBm</td>
</tr>
<tr>
<td>Slot time</td>
<td>13(\mu)s</td>
</tr>
<tr>
<td>Exponent factor (\alpha)</td>
<td>2.00</td>
</tr>
<tr>
<td>Received power threshold</td>
<td>3.162e-13W</td>
</tr>
<tr>
<td>Communication range</td>
<td>150m</td>
</tr>
<tr>
<td>State packets rate</td>
<td>20 packets/s</td>
</tr>
<tr>
<td>Vehicle speed</td>
<td>80-120 km/h</td>
</tr>
<tr>
<td>Propagation delay</td>
<td>1(\mu)s</td>
</tr>
<tr>
<td>DIFS</td>
<td>64(\mu)s</td>
</tr>
<tr>
<td>Message and Header Sizes</td>
<td>512.64Byters</td>
</tr>
<tr>
<td>Number of lane</td>
<td>4</td>
</tr>
</tbody>
</table>

In this simulation, we set the communication range to 150m based on VSC report [34]. We calculate the position error over all neighboring nodes within 150m radius to
explore the performance of our proposed channel access approach. We use the measure of average tracking error as the main performance metric for CVSSs simulations. The average tracking error metric is referred to as 
\[ \frac{1}{nT} \sum_{j=1}^{n} \sum_{t=1}^{T} e_j(t), \]
where \( n \) is number of neighboring nodes, \( T \) is the total time steps and \( t \) is the time step. One of the most important parameters that determine the channel access probability of node \( i \) is the contention window size \( W_i \). By varying \( W_i \), node \( i \) can change its channel access probability. In the simulation, we mainly adjust the contention windows size to analyze and verify the performance of our proposed channel access approach.

7.2. Simulation results

The vehicle density has been selected to cover scenarios between congested traffic and free-flow traffic on a four-lane highway. According to [17], the vehicles’ speed in a scenario of free-flow traffic is 32.9\( m/s \) and the vehicles’ speed in a scenario of congested traffic is 6.1\( m/s \). The time-headway between vehicles includes the driver’s perception response time and the braking time. The driver’s perception response time is set to 0.8s and the maximum braking deceleration is 8\( m/s^2 \) [35]. Under the scenario of free-flow traffic, the maximum braking distance is
\[ S_{fb} = (32.9/2) / 2 \times 8/67.7 \approx 22 \text{m}. \]
Under the scenario of congested traffic, the maximum braking distance is
\[ S_{cb} = (6.1/2) / 2 \times 8/2.25 \approx 2.25 \text{m}. \]
The critical inter-vehicle spacing in a scenario of free-flow traffic is
\[ S_f = 32.9 \times 0.8 + 67.7 = 93.92 \text{m}. \]
The critical inter-vehicle spacing in a scenario of congested traffic is
\[ S_c = 0.8 \times 6.1 + 2.25 = 7.13 \text{m}. \]
Assuming 4m of vehicle length, the vehicle density on a four-lane congested highway is
\[ \rho_c = 4\text{lanes} \times 1/(7.13+4/2) \approx 0.4 \text{vehicles/m}. \]
The vehicle density in a four-lane free-flow road is
\[ \rho_f = 4\text{lanes} \times 1/(93.92+4/2) \approx 0.04 \text{vehicles/m}. \]
In this section, we verify the performance of the proposed game-based channel access approach under the vehicle density ranging from 0.05vehicle/m to 0.4vehicles/m.

To validate the proposed utility function of vehicle tracking accuracy, we compare the computed tracking errors from the proposed utility function with the simulation results. The vehicle density is set to 0.2vehicle/m in this simulation. In this simulation, all the nodes use the same contention window size and adjust it in synchronization with other nodes. The results are averaged over 10 simulation runs. Fig. 3 plots the average tracking error obtained by each node with different contention window sizes using the simulation and the theoretical model. As can be seen, the theoretical model of vehicle tracking accuracy matches well with the simulation results and hence, the vehicle tracking model is highly accurate. Fig. 3 also verifies that there exists an optimal point of contention window size such that the minimum tracking error (which is approximately equal to 0.43m) of each node can be achieved. We can see that the average tracking accuracy of each node is maximized at optimal point of \( W *=515 \). When \( W < W * \), each node simultaneously tries to access the channel all the time, which results in repeated packet collisions. As a consequence, the nodes fail to receive the state information from their neighboring nodes and cannot tracking their neighboring nodes in real-time. When \( W > W * \), nodes spend some time in the back-off process before accessing to the channel for broadcasting messages, which causes time delay and leads to expired packets accor-
Fig. 3. Tracking errors vs. contention window size.  

Fig. 4. Unilateral deviation from the optimal point.

We next analyze the stability of the optimal point \( W^* \). We want to verify whether \( W^* \) is the Pareto-optimal point or not. We also set the vehicle density to 0.2 vehicle/m in this simulation. We randomly select a node, labeled as Node X, and change its contention window size from 0 to 1000. The corresponding contention window value of the other nodes except Node X (denoted as Normal nodes) is set to the optimal value \( W^*=515 \). Fig. 4 plots the obtained tracking errors of Node X and Normal nodes in CVSSs. The results are averaged over 10 simulation runs. As can be seen from Fig. 4, when Node X deviates towards the values higher than \( W^* \) (\( W_X>515 \)), it suffers an increase in its tracking error. When it deviates towards the values lower than \( W^* \) (\( W_X<515 \)), it obtains a reduced tracking error. This is because that Node X attains the most of the channel resource and has more opportunity to obtain access to the channel for broadcasting its messages when it deviates towards a lower value of contention window. Thus, the neighboring nodes can receive more state information from Node X and hence, provide high tracking accuracy for Node X. However, the increase of the tracking accuracy of Node X is at the expense of other Normal nodes. The tracking accuracy of Normal nodes is decreased heavily. This is because that deviating Node X monopolizes channel resources, which cause other nodes to have little chance to access to the channel for broadcasting messages. From this experiment, we can see that the deviation of Node X from the optimal point produces a higher utility. Thus, the optimal point \( W^*=515 \) is not the Nash equilibrium point. However, Node X achieves its high track accuracy at the cost of other nodes when it deviates from the the optimal point \( W^*=515 \) and hence, this point of \( W^* \) is a Pareto-optimal point.

To verify whether the proposed GCAA can guide each node to reach a Nash Equilibrium Point or not, we randomly pick up three nodes (labeled as Node X, Y, Z) and set their contention window values to be 15, 63 and 127, respectively. The contention window values of the other nodes are fixed at \( W^*=515 \). The vehicle density in this scenario is \( \rho=0.2 \) vehicle/m. Fig. 5 shows the corresponding evolution of contention windows sizes of Node X, Y and Z, respectively. As can be seen from Fig. 5 that Node X, Y and Z are penalized for their behaviors and hence, their contention window sizes are guided to the optimal point \( W^* \). Table 2 shows the average tracking accuracy of the three nodes and other nodes, which is obtained by GCAA and the intervehicle transmission rate control.
strategy (denoted as ITRCS) [14] respectively. ITRCS is a well-known transmission control strategy that adapts the packets generation rate to realize robust tracking. We can see that the tracking errors of the three nodes are all high in ITRCS. This is because all the three nodes in ITRCS operate at low contention window sizes point and consequently, they simultaneously try to access the channel all the time, which results in packets collisions. GCAA adapt its contention windows size and can achieve the fair access to the channel for all nodes. Therefore, our approach is adaptive to the scenario where there are multiple nodes with lower contention window sizes and hence, can provide the optimal tracking performance for each node.

![Fig.5. The evolution of the contention windows of three nodes with lower contention window sizes.](image)

<table>
<thead>
<tr>
<th>Table 2 Tracking errors obtained by different nodes (m).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Node X</td>
</tr>
<tr>
<td>Node Y</td>
</tr>
<tr>
<td>Node Z</td>
</tr>
<tr>
<td>Normal nodes</td>
</tr>
</tbody>
</table>

To verify whether the proposed distributed channel access algorithm (DCAA) can lead each node to reach the Pareto-optimal point, we implement DCAA in NS3. The vehicle density in this simulation is \( \rho = 0.2 \text{vehicle/m} \). All nodes initialize their contention window sizes to 15. The nodes continue their search for Pareto-optimal point only if they see a decrease of 0.05 (\( \varepsilon = 0.05 \)) in their tracking errors from the last Nash equilibrium point. Fig. 6 plots the average tracking errors obtained by all the nodes. As can be seen from Fig. 6, the average tracking error is minimized at contention window \( W = 515 \), which is the Pareto-optimal point when \( \rho = 0.2 \text{vehicle/m} \). All nodes will stop at this point \( W = 515 \) and the system will continue operate at the Pareto-optimal point. For completeness, we obtain the dotted curve in Fig. 6 by deliberately forcing the nodes to go beyond \( W = 515 \). As a result, we can see an increase of the tracking errors of each node when the contention window size is higher than 515.
To verify whether the proposed GCAA can guide the deviating node to Pareto-optimal point, we compare the tracking errors obtained using GCAA with those obtained using beaconing strategy (denoted as BS) and ITRCS. BS is a solution proposed by vehicle safety communication consortium (VSCC). In BS, vehicles send state information every 100ms to its neighboring nodes [34]. The simulation setup also sets vehicle density \( \rho = 0.2 \text{vehicle/m} \) and each node initializes their contention window size to the Pareto-optimal point \( (W^* = 515) \). We randomly select a deviating node from these nodes, labeled as Node X, that unilaterally deviates from the Pareto-optimal point of the operation \( (W^* = 515) \). According to [9], we characterize the magnitude of deviation of nodes with the parameter of “percentage of deviation” (PD). A deviating nodes with PD=x percent means that it transmits a packet after a \((100-x)\) percent of the assigned contention window size. Larger value of PD indicates a greater deviation of a node. The result is averaged over 10 runs of the simulation.

These comparisons are shown in Fig. 7 and Table 3, respectively. Fig. 7 plots the average tracking errors of Node X and other nodes except X (denoted as Normal nodes) when using BS, ITRCS and GCAA. Table 3 shows the corresponding statistics of the average tracking errors of Node X and Normal nodes when using the above three approaches. As can be observed from Fig. 7 and Table 3, the obtained tracking error of Node X is restricted to the Pareto-optimal point using the proposed GCAA, while the tracking error of Node X drops quickly as the percentage of deviation increases using BS.
and ITRCS. The reduced tracking errors of Node X using BS and ITRCS, however, are at the expense of preventing Normal nodes from broadcasting state information and consequently, the tracking accuracy of Normal nodes is degraded heavily when the extent of deviation of Node X increases gradually. When using the proposed GCAA, the tracking errors of Normal nodes are not affected and continually maintain the Pare-to-optimal tracking accuracy throughout the change of the extent of deviation. Hence, the proposed channel access approach is successful in ensuring the optimal tracking accuracy for all nodes. In addition, the tracking errors obtained using BS are higher than the tracking errors obtained using ITRCS. This is because that the packets generation rate is dynamically adjusted based on the estimated tracking errors in ITRCS, but restricted to fixed value in BS. Thus, BS would suffer increased amount of packet collisions, leading to higher tracking errors.

![Tracking error vs vehicle density](image.png)

**Fig. 8.** Tracking errors versus vehicle density.

To test the capacity and effectiveness of the proposed GCAA, we compare tracking errors of GCAA with those of BS and ITRCS. From Fig. 8, we can see that during each stage of change of the vehicle density, the proposed GCAA performs well and the tracking errors produced by GCAA all reach the minimum values. The tracking errors using BS, however, are quite high, especially when there is a high vehicle density. The contention window size of each node in BS is restricted to a fixed and relatively smaller value and consequently, more nodes would simultaneously try to access the channel all the time, which results in repeated packet collisions. Thus, BS suffers from higher tracking errors when there is a high vehicle density. In ITRCS, the packets generation rate is dynamically adjusted based on the estimated tracking errors. When a collision happens in ITRCS, at least two nodes will send tracking information at the same time. This results in consecutive collisions and consequently, brings about higher tracking errors. In the scenario where the vehicle density is high, our proposed GCAA increases the contention window sizes to reduce the opportunity to access the channel all the time for all nodes, which reduces the packets collisions ratio and hence, improves the tracking accuracy. Therefore, our proposed GCAA outcomes ITRCS and BS, and is robust to various traffic conditions.
8. CONCLUSION AND FUTURE WORK

In this paper, we present a game-based channel access approach to ensure a fair share of channel resources while achieving the optimal tracking accuracy for each node. We first propose a utility function of tracking accuracy to capture the relationship between tracking accuracy and throughput. The utility function can be used to maintain high tracking accuracy for each vehicle by adjusting the contention window. Based on the utility function, a non-cooperative channel access game model is proposed. The channel access game model uses the penalty mechanism to punish those nodes who try to monopolize the channel resources, so as to let all nodes work around a fair Nash equilibrium point. In order to implement the game model, a distributed channel access algorithm is then proposed to guide each node to the Pareto-optimal Nash point. Simulation results have shown that the proposed channel access approach is robust to variations of vehicle density and can ensure the fairness of channel access by all vehicles while achieving high tracking performance for each vehicle.

Our future work will include the implementation of the proposed game-based fair channel access approach on more complicated traffic scenarios, particular in interactions scenarios with high vehicle density. Furthermore, power and channel access control should be jointly designed in an adaptive fashion for the optimum performance of vehicle tracking.

REFERENCES


L. S Tan, Z. X. Zhu, F. Ge and N. X. Xiong, “Utility Maximization Resource Alloca-


Fuxin Zhang received his B.S. and M.S. degree in Department of Computer Science and Technology from Shandong University of Science and Technology, P.R. China, in 2006 and 2010. He is currently working towards the Ph.D. degree in School of School of Computer Science and Technology at Dalian University of Technology, P.R.C. His research interests include vehicular networking, wireless communication, power control, mobility modeling and cooperative vehicle safety systems.
Guozhen Tan received the B.S. degree from Shenyang University of Technology, Shenyang, M.S. degree from Harbin Institute of Technology, Harbin, and the Ph.D. degree from Dalian University of Technology, Dalian, P.R. China. He was a visiting scholar with the Department of Electrical and Computer Engineering of University of Illinois at Urbana-Champaign, IL, U.S., from Jan 2007 to Jan 2008. He has been a Professor with the School of Computer Science and Technology, Dalian University of Technology, Dalian, P.R.C. and Director of Engineering and Technology Research Center for the Internet of things and Collaborative Sensing, Liaoning province, and currently is Dean of the School of Computer Science and Technology, Dalian University of Technology, Dalian. His research interests include the Internet of things (IOT), cyber-physical systems (CPS), vehicular networking, mobility modeling, wireless communication, cooperative vehicle safety systems, network optimization algorithms, etc.

Chao Yu received his Ph.D. degree in computer science from the University of Wollongong, Australia, in 2013. Now, he is a lecturer in the School of Computer Science and Technology at the Dalian University of Technology, Dalian, China. His research interests include vehicular networking, multi-agent systems and internet of things.

Nan Ding received the B.S., M.S., and Ph.D. degrees in Computer Science and Engineering from Dalian University of Technology, Dalian, P.R. China, in 2001, 2005, and 2011, respectively. He has been an Assistant Professor in Computer Science and Engineering from Dalian University of Technology, Dalian, P.R. China since 2008. His research interests include vehicular networks, wireless security, wireless sensor networks, cache management, and distributed fault tolerant computing. (Email: dingnan@dlut.edu.cn)

Mingjian Liu, born in 1984. Master candidate at the Department of Computer Science and Technology, Dalian University of Technology. His main research interests include vehicle safety early warning technology, vehicular communications.