Blockchain based Model for Nondeterministic Crowdsensing Strategy with Vehicular Team-Cooperation

Jianrong Wang, Xinlei Feng, Tianyi Xu, Huansheng Ning, Senior Member, IEEE, Tie Qiu, Senior Member, IEEE

Abstract—Smart vehicles can cooperate in teams to perform crowdsensing tasks in smart cities. A critical challenge in this regard is to build a secure model for nondeterministic vehicle teams to achieve maximum social welfare. Although several crowdsensing models have been proposed, none of them has focused on real-time vehicle teamwork. In this study, to the best of our knowledge, we propose the first secure model, called Blockchain-based Nondeterministic Teamwork Cooperation (BNTC), for nondeterministic teamwork cooperation in a vehicular crowdsensing system. We model the system as a multi-conditional NP-complete problem by explicitly considering the dynamic features of task issuers and workers. To solve the problem, we propose Winning Teams Selected (WTS) algorithm based on a reverse auction and utilize a knapsack-based method to solve the models. We consider credit of teams for determining the payment. Thus, we propose a Credit-based Team Payment (CTP) algorithm for BNTC to maximize the welfare of the system. We also propose a general blockchain-based framework to address trust issues and security concerns about the leakage of private information when sharing data. Thus, an effective incentive strategy and safe platform are important for MCS.

Typical crowdsensing schemes function well for tasks that require a single worker; however, tasks that require real-time cooperative teamwork by multiple workers are likely to fail because of subtask failures by some workers or time-outs due to a lack of sufficient workers. In addition, multiple vehicles can form a pool of powerful computing resources because of the increasing computing capacity of vehicle computers. The computing power of the massive number of vehicles on roads can significantly support computational tasks in smart cities due to the development of distributed computing. However, efficiently utilizing the computational power of numerous vehicles on the roads in a smart city remains to be an unresolved problem. Thus, building a secure MCS mechanism for IoV with suitable incentives based on teamwork is an important issue for smart cities. What’s more, for determination of payment for workers, some works pay a worker according to highest utility value based on external pricing [7], without considering credit and fairness of overall utility, which are meaningful evaluation metrics to indicate the performance of the worker and MCS system.

In the present study, to address the problems described above, we design a model called Blockchain-based Nondeterministic Teamwork Cooperation (BNTC) in a MCS system with IoV to exploit the real-time collaborative computing power of numerous vehicles to complete MCS tasks which require teamwork. The main contributions of the present study are as follows.

1) We design a novel model based on nondeterministic crowdsensing for vehicular teams. A real-time MCS task can be distributed by the MCS platform to one or more vehicular teams directly rather than to some single vehicles. Vehicles equipped with smart devices form teams to complete tasks. The model based on teamwork can improve the completion ratio for tasks by selecting appropriate teams according to their capabilities.

2) To determine the distribution of tasks, we propose an optimized reverse auction mechanism based on the knapsack algorithm called WTS algorithm for BNTC, where it considers
task completion ratio, social cost, and number of teams selected. We consider the impact of credit factors on the critical payment and propose a CTP method for BNTC, which is inspired by the payment determination algorithm described by [7]. The two algorithms can ensure that the model maximizes the social welfare.

3) We propose a blockchain-based framework for the non-deterministic MCS with IoV model. All members in the system are in a blockchain. Key services are handled by smart contracts. The framework can ensure the privacy of data for users and the data integrity for the system. The workers can work and receive payments anonymously. The fairness of the payments is guaranteed and the framework makes the model suitable for practical applications.

II. RELATE WORK
A. Incentive Mechanisms for Mobile Crowdsensing

Incentive mechanisms are employed to select appropriate workers and to determine suitable payments for workers in different application models, and to ensure that the social welfare is maximized [8][9][10]. Wang et al. [11] proposed a graph-based solution to transform Minimum-Delay-Maximum-Coverage and Minimum-Overhead-Maximum-Coverage to a connection routing search problem. And Greedy-based recursive optimization approaches were proposed to address the two problems with a divide-and-conquer mode. Gao et al. [12] considered different vehicle trajectories and the uncertainty of driving routes to establish a MCS task system where each task could be performed jointly by multiple vehicles. Chen et al. [13] studied location-aware and location diversity based dynamic MCS system. However, these algorithms could not deal with tasks that needed to be completed by multiple workers within a specific period of time. For works on real-time cooperation MCS, Yin et al.[14] studied a time-window based method to manage the emergency task. This method selected idle vehicles when emergency task happened. Chen et al. [15] proposed a crowd tracking system that people can collaboratively keep track of the moving vehicle by taking photographs. [16] proposed a quality-driven auction based incentive mechanism with EM algorithm to guarantee trust. However, candidate vehicle is selected in sequence in those methods to form a vehicle set. They lack attention to big scale joint computing vehicle resources for MCS.

B. Blockchain for IoV

Vehicles can be connected with RSU via advanced wireless network device[17]. Many methods have been proposed to solve the identity privacy problems associated with the IoV when it is combined with a blockchain. Lu et al. [18] used two types of blockchains to hide the connections between real identities and public keys. Yao et al. [19] implemented cross-data center authentication and allowed users to request changes to their pseudonyms to protect personal privacy. To address the problem of data security, Zhang et al. [21] addressed the challenge of combining the mobile features of the IoV with a blockchain. Kang et al. [20] used a combination of traditional cloud storage and a blockchain to ensure the reliability of data in VANETs. Yue et al. [22] used a consortium blockchain to develop a credit-based data sharing scheme and proposed the use of a three-rights subjective logic model to monitor the sources of data to improve the data reliability. A research[25] has proved entering the blockchain system anonymously guaranteed the user’s unconnectability and security to a certain extent.

In summary, existing researches lack a MCS framework which can utilize real-time joint computing power and ensure security protection at the same time. Most of the existing MCS considers to allocate the subtask to a specific vehicle without considering teams, which leads to massive time consuming and cost. In addition, information integration can not be secured in a central MCS service sometime.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In the following, we describe the entities in the model, and explain the design of the functions and services associated with the entities. We then illustrate the crowdsensing process in our model.

A. Entity Definition

The model is designed on a blockchain. The network structure of the model is based on the Internet and a VANET[23]. The model has the following nodes: task issuer node, vehicle node, RSU node, and Personal Certificate Authority(PCA) node.

1) Task Issuer Node: Task issuer nodes are generally organizations or companies with the need for crowdsensing. They register in the blockchain as full nodes to maintain the whole ledger. Task issuers can upload task information to the system, such as completion conditions. Finally, task issuers submit payments to workers via the system.

2) RSU Node: RSU can identify vehicle nodes that are capable of participating in crowdsensing tasks in real time. Each RSU node maintains a table for storing information about teams formed by vehicles in the current time period. When a task is crowdsensed, a smart contract collects information from the RSU nodes to determine the team or teams having the ability to complete the task. When a team is selected, an RSU node receives task information and distributes it to the vehicles in the team, as well as collecting and uploading the results submitted by the vehicles. RSU nodes are also full nodes in the blockchain system and they maintain the blockchain.

3) Vehicle Node: Vehicle nodes are vehicles with a vehicle computer, smart sensors, and network interfaces. All vehicle nodes can communicate with RSU nodes with wireless network[24]. Vehicle nodes can submit their status, willingness to complete tasks, and bidding information to RSUs. They can receive task information from RSUs and submit results to RSUs. Each vehicle node has credit value. We define the node with high credit as formal nodes and the nodes with low credit as informal nodes. In the blockchain system used in our model, vehicle nodes are light nodes registered in the blockchain system because of their poor storing capacity and they have no sufficiently power to generate blocks.
4) **PCA:** PCA is a management agency responsible for identity verification and registering the nodes in the blockchain system in our model. All the nodes must register with the PCA to be added to the blockchain. And PCA has access to cloud server which can provide complicate computing task and store massive data secretly.

**B. BNCTC Process**

BNCTC system shows in Fig. 1.

1) **Init System:** Nodes with different roles must register in the blockchain system to use the service. To ensure the anonymity and security of all the worker(vehicle node), a certificate is issued by the PCA[25] based on Pointcheval and Sanders(PS-signatures)[26]. The types of nodes in our system are different and the blockchain platform must indicate the roles of the nodes in BNCTC by issuing different types of certificates.

2) **Crowdsensing Process:** The crowdsening process of BNCTC is described as follows.

Step 1. An RSU node located at a traffic light sets up network linking the vehicles that enter its road section. The RSU collects information about the vehicles. The information is maintained and refreshed within a certain time period by the RSU.

Step 2. When the task issuer uploads a task for MCS, a deposit is placed in the blockchain to calculate the candidate teams and payment. The platform scans all the RSUs to collect information about the teams and to calculate the most suitable team set for the task by using WTS algorithm for BNCTC. The payment for winning set is also calculated in this step based on the credit and bidding price from each vehicle in the team based on the CTP algorithm for BNCTC.

Step 3. The winner team set and its payment information are submitted to the task issuers by the system. If the task issuer agrees with the result, the result is signed and the payment is submitted to the system, otherwise the result will be dropped. The blockchain system records the information for the teams, vehicles, RSU, task, task distribution, task issuer, and payment after the result is signed by the task issuer. Each winning RSU distributes subtask to the vehicles in the selected team. After the vehicles complete their task, the result is collected by the RSU. The RSU records the results in the blockchain and sends them to the task issuers at the same time. Smart contracts guarantee that the workers are paid correctly.

**C. Problem Formulation**

The social welfare should be maximized for the proposed model. Therefore, a satisfaction formula must be defined to measure the welfare for society, and thus, two problems need to be formulated. The first problem involves the manner by which to select one or several appropriate teams for a task according to the task requirements, bidding information, and capability of each vehicle to achieve a high task completion ratio at a low cost. The second problem involves determining an appropriate payment for the selected team to ensure that the incentive is adequate for workers. These two problems both contribute to the social welfare of the system.

**IV. Algorithm**

To solve the problems raised above, we introduced two algorithms for the BNCTC system: Winning Teams Selected algorithm and Credit of Teams Payment algorithm for the BNCTC model. The main notations used in this system are shown in Table I.

**TABLE I: Notations**

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>Existing total vehicular teams</td>
</tr>
<tr>
<td>V_v</td>
<td>Existing total vehicular teams except team v</td>
</tr>
<tr>
<td>S</td>
<td>Scale of task</td>
</tr>
<tr>
<td>N</td>
<td>Total number of vehicular teams</td>
</tr>
<tr>
<td>v_k</td>
<td>The k\textsuperscript{th} vehicular team</td>
</tr>
<tr>
<td>y_k_i</td>
<td>The i\textsuperscript{th} vehicle in the k\textsuperscript{th} vehicular team</td>
</tr>
<tr>
<td>t_c_k</td>
<td>Number of sub-tasks which vehicle team v_k can complete</td>
</tr>
<tr>
<td>q_k_i</td>
<td>Completion Probability of vehicle y_k_i</td>
</tr>
<tr>
<td>P_k_i</td>
<td>Completion Probability of vehicular team v_k</td>
</tr>
<tr>
<td>r_k_i</td>
<td>Credit value of vehicle y_k_i</td>
</tr>
<tr>
<td>R_k_i</td>
<td>Credit value of vehicular team v_k</td>
</tr>
<tr>
<td>B_k</td>
<td>The average bid of vehicular team v_k</td>
</tr>
<tr>
<td>(\chi)</td>
<td>Winning team set of the task</td>
</tr>
<tr>
<td>Cost_(\chi)</td>
<td>Cost set of winning team set</td>
</tr>
<tr>
<td>Payment_(\chi)</td>
<td>Payment set to winning team set</td>
</tr>
<tr>
<td>(\chi_v)</td>
<td>Winning team set of a task excluding team v</td>
</tr>
<tr>
<td>A</td>
<td>Number of winning teams</td>
</tr>
</tbody>
</table>

**A. Problem Description**

The following definitions are required to explain the algorithm. In the entire network, N vehicular teams form a whole team set \(V = \{v_1, v_2, ..., v_N\}\). The formal vehicles included in the team \(v_k\) are \(\{y_{k1}, y_{k2}, ..., y_{kn}\}\) with a total number of \(m_k\). The informal vehicles are not considered because the low completion rate in their history would lower the predicted probability of completing a task by the team. The completion capacity of a team is defined as \(tc_k = m_k \times c\), where \(c\) is a constant value which denotes the number of subtask one vehicle can complete within a limited time. One task can be divided into A parts, \(\{s_1, s_2, ..., s_A\}\). A vehicular teams can complete a task together.
The social payment for a task issuer is given by

\[ W_{Social} = \frac{S}{\alpha_1 \ast Cost + \alpha_2 \ast OPR \ast Cost + \alpha_3 \ast A}. \]  

(10)

\( S \) is the scale of a aim task that needs to be completed. \( A \) is the number of winning teams. \( \alpha_1, \alpha_2 \) and \( \alpha_3 \) are adjustable parameters to ensure the progressive of the three conditions.

The winner team set is given:

\[ \chi = \{x_1, x_2, ..., x_A\}. \]  

(11)

In this model, we need to determined \( \chi \) to maximize \( W_{Social} \). The following assumptions are required to solve \( \chi \).

Assumption 1: The existing vehicular network must be able to satisfy the demand in terms of the number of sub-tasks uploaded by the task issuers. Thus, a task can be crowdsensed by one or several teams.

Assumption 2: The team formed to complete a partial task must remain stable throughout the whole process, so the completion of the work depends only on the task completion probability for each vehicle.

Assumption 3: The bid of a vehicle is its real cost for a sub-task.

Assumption 1 and assumption 2 are easy to meet in practical. Assumption 3 is proved in [16].

C. Winning Teams Selected algorithm for BNTC

Teams or joint teams with a high probability of task completion are selected as the candidate set for a task, thereby improving the task completion rate. Selecting appropriate teams for a task from the candidate set involves minimizing the number of teams to reduce the time consumption by the system and minimizing the total cost for the vehicular teams to satisfy the task issuer. Thus, a satisfaction formula that satisfies the multi-conditional problem is defined and our aim is to select a candidate team based on the satisfaction formula. We define the satisfaction(SAT) formula as follows:

\[ SAT = -(a_{min} - a_{max} - 1) \sum_{\chi} B_i - A) \]  

(12)

\[ = (1 + a_{max} - a_{min}) \sum_{\chi} B_i + A. \]  

(13)

where, \( a_{min} \) and \( a_{max} \) is the boundary value that divides the number of teams.

**Proof:** First, we consider that the bidding \( B \) is constant. A low team number \( A \) is better. We define the formula:

\[ f(A) = k_1A \quad s.t. \quad A_1 > A_2, f_1 < f_2 \]

Thus, \( k_1 < 0 \). We simply consider \( k_1 = -1 \). Similarly, for bidding \( B \), we extend the formula above to the following:

\[ g(B, A) = k_2B - A \]
We consider that the satisfaction is lower when the cost is higher. When \( B_1 > B_2, \forall A_1, A_2 \in [a_{min}, a_{max}] \), we can obtain \( g_1 < g_2 \). Thus,

\[
k_2B_1 - A_1 < k_2B_2 - A_2
\]

\[
k_2 < \frac{A_1 - A_2}{B_1 - B_2}
\]

So, we first use the inequality scaling method to obtain

\[
k_2 < \min\left(\frac{A_1 - A_2}{B_1 - B_2}\right)
\]

We second use the inequality scaling method to get

\[
k_2 < a_{min} - a_{max}
\]

Simply, we make

\[
k_2 = a_{min} - a_{max} - 1
\]

We obtain the formula:

\[
g(B, A) = k_2B - A = (a_{min} - a_{max} - 1) \times \sum \chi B_i - A. \quad (14)
\]

\[
\xi = \frac{1}{\sum v \in \chi} B_v. \quad (15)
\]

We can conclude that the SAT result is always a negative number. Therefore, we reverse it and turn (15) into a positive number in the calculation process as (13). Thus, the satisfaction is greater when SAT is smaller. To solve the problem of selecting teams \( \chi \) for tasks, we propose a knapsack-based algorithm called winning teams for the BNTC algorithm. In BNTC algorithm, the dynamic state transfer function is to get the minimum value which denotes the maximum satisfaction at that aim scale task. We use array \( SA \) to denote the status of \( SAT \) in the procedure of dynamic transfer,

\[
SA[j] = \min\{SAT(SA[j - TC[i]], TC[i]), SA[j]\} \quad (16)
\]

where, \( j \) denotes the task scale. When the task scale is \( j - TC[i] \) and team \( i \) is selected, the task scale is \( j = (j - TC[i]) + TC[i], \) the satisfaction function of scale \( j \) is \( SAT(SA[j - TC[i]], TC[i]) \). We update \( SA[j] \) with the minimum value between \( SAT(SA[j - TC[i]], TC[i]) \) and previous \( SA[j] \).

The pseudo code is shown as Algorithm 1. Task scale \( S \), candidate teams count \( N \), bids set \( BV[N] \) and team capacity \( TC[N] \) are input while winning team set \( \chi \) is output. We need traverse from 0 to \( N \). We calculate from scale \( S \) to the scale \( TC[i] \) of team \( i \). The satisfaction of every team scale is calculated until the minimum value which denotes maximum satisfaction of that scale is achieved.

The time complexity of the algorithm is \( O(SN) \). Using the idea of dynamic programming, the knapsack algorithm can directly determine the optimal result that satisfies the completion of a certain number of tasks.

### Algorithm 1 Winning-Bid Selection Algorithm

**Input:** \( S, N, BV[N], TC[N] \)

**Output:** \( value, nums, \chi \)

1. \( WinningTeamSet \leftarrow [] \)
2. \( nums \leftarrow 0 \)
3. for \( i = 0 \) to \( N \) do
4.  for \( j = S \) to \( TC[i] \) do
5.    \( bs \leftarrow SAT(SA[j - TC[i]], TC[i]) \)
6.    if \( SA[j] > bs \) then
7.      \( SA[j] \leftarrow bs \)
8.    WinningTeamSet[j] \leftarrow WinningTeamSet[j - TC[i]] \cup i
9.  end if
10. end for
11. \( i \leftarrow i + 1 \)
12. end for
13. for \( i \in WinningTeamSet[S] \) do
14.   \( value \leftarrow value + B[i] \)
15. end for
16. \( nums \leftarrow nums + 1 \)
17. \( \chi \leftarrow WinningTeamSet[S] \)
18. return \( value \) and \( nums \) and \( \chi \)

### D. Credit-based Team Payment algorithm for BNTC

In this section, we discuss how to determine the rewards for selected teams with CTP algorithm for BNTC. To guarantee the fairness of each team, we pay team \( x \) according to its credit and external pricing which is calculated by utility and overall utility of another best candidate team set with \( x \) excluded. When a task can’t be completed without team \( x \), team \( x \) is regarded as “Critical Team”. When a task can be completed without team \( x \), team \( x \) is regarded as “Ordinary Team”. First, we define the utility of a team \( x \) for the sub-task as \( Utilv \):

\[
Utilv = TC_x / B_x. \quad (17)
\]

We define average utility value of a new winning set without team \( x \) as \( eUtilv \):

\[
eUtilv = S/ \sum_{v \in \chi - x} B_v. \quad (18)
\]

Thus, we determine the payment to winner team \( x \) as \( Pay_x \). If \( x \) is an “Ordinary Team”, we pay \( x \) as (19). If it’s a “Critical Team”, we pay \( x \) as (20).

\[
Pay_x = tc_x((1/eUtilv + 1/Utilv_{min})/2 + (R_0 - R)^2) \cdot \varphi \quad (19)
\]

\[
Pay_x = tc_x((1/Utilv_{min} + (R_0 - R)^2) \quad (20)
\]

where, \( R_0 \) is the benchmark value of the reward. \( \varphi \) is an adjustable parameter for reward. The description of this algorithm is listed in Algorithm 2.

We need to give a price to each vehicle in the network team and each price must be calculated by recalculating the allocation. Thus, the time complexity is: \( O(AN^2) \).
Algorithm 2 Rewards Payment Algorithm

Input: $S$, $C$, $B[V[N]]$, $T[C[N]]$, $\chi$
Output: Reward $\text{Payment}_x$
1: for all $x \epsilon \chi$ do
2: $TC \leftarrow TC_{-x}$
3: $BV \leftarrow BV_{-x}$
4: if CriticalTeam then
5: $payment_x \leftarrow eq.(20)$
6: end if
7: recalculate winning set result without $x$ as $\chi_{-x}$, $value_{-x}$, $num_{-x}$ by Algorithm 1
8: select $Utilv_{min} = \text{minimum}(Utilv)$ from $\chi_{-x}$
9: $payment_x \leftarrow eq.(19)$
10: end for
11: return $\text{Payment}_x$

V. EVALUATION

In the following, we introduce the experimental environment and the dataset employed. We first experimentally evaluate WTS algorithm and CTP algorithm for BNTC, as described in the previous section, and analyze the experimental results. Then we evaluate resource consumption of BNTC in blockchain.

A. Dataset and Experimental Design

We mainly conduct simulation experiments. We simulate the status of different users for bidding tasks. We use the dataset of [27] in Bologna, Italy. This dataset collected road vehicle information from 8 am to 9 am a day. The road map of the dataset is shown in Fig. 2. When a congestion occurs at an intersection, a vehicular network team is formed by RSU (more than 10 vehicles). We assume all the vehicles on road are able and willing to participate in conducting crowdsensing tasks. And all data submitted by vehicles are reliable.

TABLE II: Initialization of Major Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit of a vehicle</td>
<td>1.0</td>
</tr>
<tr>
<td>Initial probability of a vehicle</td>
<td>0.8</td>
</tr>
<tr>
<td>Require probability $\delta$</td>
<td>0.5</td>
</tr>
<tr>
<td>Cost coefficient range</td>
<td>[1,1.5]</td>
</tr>
<tr>
<td>$\omega$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\alpha_1$, $\alpha_2$, $\alpha_3$</td>
<td>0.7,0.2,0.1</td>
</tr>
<tr>
<td>$R_0$, $\varphi$</td>
<td>0.5,10</td>
</tr>
</tbody>
</table>

B. Experiment Result

We use the following metrics to evaluate the performance of our system: social welfare, social cost of team workers [16], social cost of task issuers (payment), the overpayment ratio, and number of winning teams. The baseline algorithms in this experiment were MCBS [12] and the DQDA algorithm [16] in the EM algorithm for evaluating the data quality.

The initial settings of the simulation are shown in Table II. $\alpha_1$, $\alpha_2$ and $\alpha_3$ are parameters representing the weight of costs, overpayment and account of winning teams in our model. We init them based on the concerns of maximize social welfare. $R_0$ is base credit set by PCA. $\omega$ and $\varphi$ are parameters to adjust payment to each vehicle and can be initialized by experience.

1) Social Welfare: The social welfare results are shown in Fig. 3. The social welfare is measured using the satisfaction formula. The social welfare is largest with BNTC among the three algorithms.

2) Social Cost: In the case of different workloads, the costs for all workers with each algorithm are shown in Fig. 4. The results indicate that the cost increases and the task issuer needs to pay more as the task scale increases. As shown in Fig. 3, the cost of the BNTC algorithm is always the smallest under the same condition and the knapsack algorithm could obtain the optimal solution in a distribution.

3) Social Payment: The task issuer pay rewards to the selected team. The results indicate that the total payment is highest with BNTC among the three algorithms under the same condition. The payment increases with task scale. The mechanism employed by MCBS is similar to that used by BNTC. However, MCBS considers the utility ratio rather than scale. The results are shown in Fig. 5.

4) Overpayment Ratio: The experiment on the overpayment ratio demonstrates that the overpayment ratio is not correlated with the scale of tasks. MCBS still lead to high...
overpayment ratio due to its external prices mechanism. In some case(task scale 210), the over ratio of DQDA is the best. The payment are more approximative with real cost. But it suffers high cost and payment problems. The results are shown in Fig. 6.

5) Winning Team Count: The numbers of teams selected for a task are shown in Fig. 7. BNCT performs better than the baseline methods in most situations. However, BNCT is not a greedy algorithm. Thus, it might not have performed as well as MCBS according to this indicator.

The experiments show that BNCT performs better than the baseline methods in terms of most of the indicators. The satisfaction formula shows that BNCT could achieve the minimum payment and cost, and the maximum social welfare. Compared with the other two methods, BNCT has the lowest social cost and lowest payment. Since overpayment ratio is inversely correlated with the social cost, so BNCT tends to yield a high overpayment ratio in some cases. Above all, BNCT is more effective than baselines.

6) Credit For BNCT: Simulations for overpayment ratio of a teamset is shown in Fig.8. With a larger credit value, a team is likely to get a higher pay for its work.

A vehicle will get a heavy penalty for failures(e.g. malicious actions). It will take more successful actions to get back to its previous normal credit value. Simulation results is shown in Fig.9.

C. Resource Consumption Evaluation

1) Registration Performance: PCA generates an anonymous certificate for each vehicle node and key pair for RSU node and task issuer node. We evaluate the performance for certificate and key pair generation in Golang off-line. Results show the time consumption is acceptable, as shown in Table III.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Entities</th>
<th>Time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate</td>
<td>Vehicle Node</td>
<td>166</td>
</tr>
<tr>
<td>Key Pair</td>
<td>RSU node and Task Issuer Node</td>
<td>27</td>
</tr>
</tbody>
</table>

2) Smart Contract Performance: We build an Ethereum private chain with a PC and five Raspberry Pies for simulation. PC is regarded as task issuer. RSU services and data are loaded into Raspberry Pies. In order to show the performance of WTS algorithm and CTP algorithm for BNCT in smart contract, we use gas model of Ethereum to measure the cost of each function of the algorithms. We set the gas price as 0.000000001 (1 Gwei) Ether per gas as in [16]. The price of each Ether is around 141.53$ on April 1st in 2020. The two algorithms only cost 1.419S(0.2304+1.189) to get distribution and payment result. The resource consumption is within a reasonable range.

VI. CONCLUSION

In this study, we propose a nondeterministic vehicular team task model to efficiently utilize the joint computing power of numerous vehicles in cities. Our model employs team cooperation to ensure the completion of crowdsensing tasks. The model runs on the blockchain platform and smart contracts can guarantee the security of the system. Based on the reverse auction method, we propose WTS algorithm and CTP algorithm for the BNCT to maximize the social welfare and minimize the time consumption. Theoretical analyses and extensive simulations demonstrate that the proposed model performs better than the baseline methods and it achieves the maximum social welfare. At last, implementation with Ethereum suggests our model can operate within a reasonable cost.

REFERENCES


TABLE IV: Resource Consumption Evaluation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Function</th>
<th>Description</th>
<th>Transaction Cost (gas)</th>
<th>Execution Cost (gas)</th>
<th>Total Cost (gas)</th>
<th>Dollar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization</td>
<td>init()</td>
<td>Initial smart contract</td>
<td>315590</td>
<td>2351810</td>
<td>5507800</td>
<td>0.3748</td>
</tr>
<tr>
<td>Add Team</td>
<td>addTeam()</td>
<td>Collect candidate teams</td>
<td>195379</td>
<td>172315</td>
<td>367694</td>
<td>0.0515</td>
</tr>
<tr>
<td>Set Task Issuer</td>
<td>setCall()</td>
<td>Set task issuer address</td>
<td>65372</td>
<td>42180</td>
<td>107552</td>
<td>0.0151</td>
</tr>
<tr>
<td>Check Info</td>
<td>checkTeam()</td>
<td>Get team information</td>
<td>88537</td>
<td>64565</td>
<td>150402</td>
<td>0.0211</td>
</tr>
<tr>
<td>Set Task Aim</td>
<td>setTaskAim()</td>
<td>Set aim task scale</td>
<td>42601</td>
<td>21137</td>
<td>63738</td>
<td>0.0089</td>
</tr>
<tr>
<td>WTS Algorithm</td>
<td>getWinnerTeam()</td>
<td>Get winner team set</td>
<td>1496032</td>
<td>1493960</td>
<td>2989992</td>
<td>0.2304</td>
</tr>
<tr>
<td>CTP Algorithm</td>
<td>getPayment()</td>
<td>Get payment for winner team set</td>
<td>4180347</td>
<td>4312675</td>
<td>8493022</td>
<td>1.1890</td>
</tr>
<tr>
<td>Bulk Transfer</td>
<td>transfer()</td>
<td>Transfer for each winner team</td>
<td>39691</td>
<td>14643</td>
<td>54334</td>
<td>0.0076</td>
</tr>
</tbody>
</table>


Jianrong Wang was born in China in 1976. He received his Ph.D. master’s degree and master’s degree in computer application technology from Tianjin University in 2009, 2004 and 1998. From 1998 to 2004, he was an assistant professor at the School of Computing, Tianjin University. Since 2009, he has been an associate professor at the School of Computer Science, Tianjin University. Prof. Wang has been the instructor of the RoboCup team of Tianjin University, the instructor of the Freescale smart car team of Tianjin University, and the instructor of the TE Cup Embedded Special Invitation Team of Tianjin University. And Prof. Wang has won many national and regional awards.

Xinlei Feng was born in 1993. She received the B.S. degree in computer science and technology from Tianjin University, Tianjin, China, in 2017. where she is currently pursuing the Master’s degree. Her current research interest includes the areas of mobile crowdsensing with blockchain and auction mechanisms.

Tianyi Xu received Master degree in computer science and technology from Tianjin University, Tianjin, China, in 2015. He is currently an engineer at School of Computer Science and Technology, Tianjin University, China. His research interest includes Internet of Things and Blockchain and data mining. He has 13 papers in International journals and conference proceedings, such as IEEE ACCESS, International Journal of Distributed Sensor Networks etc.

Huansheng Ning received his B.S. degree from Anhui University in 1996 and his Ph.D. degree from Beihang University in 2001. Now, he is a professor and vice dean of the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research focuses on the Internet of Things and IoT-Enabled cyberspace. He is the founder of the Cyberspace and Cybermatics International Science and Technology Cooperation Base.

Tie Qiu Tie Qiu (M’12-SM’16) received Ph.D. degree in computer science from Dalian University of Technology in 2012. He is currently a Full Professor at School of Computer Science and Technology, Tianjin University, China. Prior to this position, he held assistant professor in 2008 and associate professor in 2013 at School of Software, Dalian University of Technology. He was a visiting professor at electrical and computer engineering at Iowa State University in U.S. (2014-2015). He serves as an associate editor of IEEE Transactions on Systems, Man, and Cybernetics: Systems, area editor of Ad Hoc Networks (Elsevier), associate editor of IEEE Access Journal, Computers and Electrical Engineering (Elsevier), a guest editor of Future Generation Computer Systems. He serves as General Chair, Program Chair, Workshop Chair, Publicity Chair, Publication Chair or TPC Member of a number of international conferences. He has authored/co-authored 9 books, over 150 scientific papers in international journals and conference proceedings, such as IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Vehicular Technology, IEEE Communications Surveys and Tutorials, IEEE Communications Magazine, INFOCOM etc. There are 12 papers listed as ESI highly cited papers. He has contributed to the development of 3 copyrighted software systems and invented 16 patents. He is a senior member of China Computer Federation (CCF) and a Senior Member of IEEE and ACM.