Fairness-Aware Game Theoretic Approach for Service Management in Vehicular Clouds

Moayad Aloqaily

Systems and Computer Engineering Department, Carleton University, Ottawa, ON, Canada, MoayadAloqaily@cunet.carleton.ca

Gnowit Inc., Ottawa, ON, Canada, Moayad@gnowit.com

Burak Kantarci, and Hussein T. Mouftah

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada {Burak.Kantarci; Mouftah}@uottawa.ca

Abstract — **Vehicular cloud computing can perform a broad set of on-demand applications and services, which makes it highly suitable for urban settings. Despite a wide range of benefits to various services and applications by vehicular clouds, there are several issues and challenges that need to be carefully addressed in the context of provisioning services. This paper proposes a cooperative distributed game model to handle service management in vehicular clouds. Under this model, service providers play a cooperative game to maximize their total utility taking into consideration their recourse availability, current load, and total payoff. The proposed game has been implemented and evaluated using simulations with scenarios of light and heavy weight services. The game demonstrates that a cooperative technique leads players to handle higher number of services when compared to a non-cooperative setting. Furthermore, we also show that the proposed game mimics the behaviour of an optimization-based baseline solution. Through various simulation scenarios, we show that the proposed scheme introduces more than 85% similarity to the optimal solution when a few number of players participate, and its similarity to the optimal solution is improved to 99% when the number of the players increases by only 50%.**

Keywords — **Vehicular cloud, mobile clouds, game theory, cooperation model, service management.**

I. Introduction

The emerging concept of vehicular cloud has been proposed in many recent works [1][2]. As vehicles get smarter with supplementary on-board gear, they are capable of handling more complex operations, and unlike other mobile devices, mobile *vehicle* devices can provide real-time functionality, location based services, provisioning services, and storage, with none of the drawbacks of traditional mobile devices. Combining vehicles' resources, and qualifying them to provide cloud services to the public has enabled to connect vehicles and turn them into "vehicular" service providers [3]. Today, vehicles are expected to support enhanced communication systems, and provide more storage, computing resources and sensing services. However, the high mobility of such environments, and the early stages of development lead to increased complexity and numerous challenges. Some challenges are related to the technical aspects of development, while others involve adapting the technology to the surrounding environment and authorized parties. Cloud vehicles typically share computing and storage resources via a wireless network backbone [4]. Vehicular cloud has been seen as a viable technology for provisioning services. However, service management in vehicular clouds still an open issue. This requires mathematical modelling of the problem to guarantee efficient and fair services distribution across cloud providers.

In our previous work, a QoE framework to provide several vehicular cloud services in a vehicular cloud at low price, with more privacy and minimal latency has been proposed [5][6]. The overall level of satisfaction of the provided services based rating measure to quantify the reputation of each service provider has also been implemented. QoE reputation value is dynamically calculated and presented to guide service requester with their selection. A great improvement to overall service charges, latency, and drivers' privacy has been proven. A vehicular trusted third-party approach has been also adopted [7]. However, the result has also showed unbalance in the service distribution among potential service providers (game players).

In this paper, we propose a multiagent observable cooperative decision theoretic approach to formulate the interaction among cloud service providers on the requested services by vehicle drivers. Cooperative games (the so called coalitional) are gaining considerable interest because of their contributions to such nature. For instance, the authors in [8] present a distributed game mode for cooperation among the roadside units in a vehicular network. The goal of our approach is to maximize the service provider's social utility while meeting drivers QoE. We employ the QoE model presented in [6] and introduce the concept of Service Providers Social Welfare (SPSW) by combining drivers' experience, efficiency of the service provision, and fairness among the game participants. Such solution guarantees best strategy for all participants leading to an optimum fair distribution of the service requests among service providers. The model has been evaluated using simulations and CPLEX optimizer and proves its feasibility.

The main contributions of the paper are as follows: 1) An optimization model formulation for multiagent interaction game system, 2) Maximizing social utility of service providers for provision services in vehicular clouds, and incorporating fairness and efficiency with Quality of Experience (QoE) framework through a cooperative game model. Therefore, the building blocks of this paper are a QoE framework, game approach, social welfare function, and utility function. It is worthwhile noting that QoE in the proposed model denotes a function of user comfort.

The rest of the paper is organized as follows. Section II covers the proposed system model including system architecture, game approach and optimization model. Performance evaluation presented and discussed in Section III. Finally, we conclude the paper and give future directions in Section IV.

II. System Model

A) Overall System Architecture

With this model, we seek a fair and efficient distribution of service requests among potential vehicular service providers who acquire such services. In this model, a vehicle driver can be a service provider and at the same time a service seeker. Presumably, the service seekers receive services analogous to best effort services in communications in terms of latency, service cost, and information privacy. QoE framework presented in [6] which denotes user comfort is associated with the proposed architecture, and it guarantees vehicle drivers with bounded latency, reduced service cost, and improved privacy. We used QoE reputation value of each system component to enable trusted third party to group the best driver/provider matching and bind them to each other. While this has brought huge benefits to the service seekers, the QoE framework is still missing the management of the service requests at the service provider's side. In our previous work, we experienced some cases where some service providers are overloaded with service requests beyond their recourse capabilities while the resources of other providers are left underutilized. Accordingly, and in order to maximize overall social welfare of service providers, a Game Engine Service Management (GESM) module at the service provider's side is needed. Figure 1 illustrates the developed system architecture. Three modules make up the architecture: Game Engine Service Management, service buyers, and service providers.

Game Engine Service Management (GESM) acquires all service providers' information including resources, available services, current involved user's, service charges, computation capabilities, game participants, game events, and QoE reputation values for each service. The architecture overviews concurrent services over the network requested from the receivers (drivers) to the sender (service provider). There are two main challenges carried by service provision in the model: 1) The interaction of multiple concurrent service providers to provide a set of services to a group of vehicle drivers. Such interaction can take different forms with many open challenges. One of most pressing issues in such model is the selfishness and interoperability. Each participant seeks his best interest without considering other participant's welfare. 2) The cooperation among different service providers is still immature in current practice. Both of these issues are directly related to the optimal of sequential decision making under uncertainty which requires full state of information about the process or an action of this process. For instance: Do service providers prefer non-cooperative settings? What are the motivations to encourage different providers to participate in maximizing their social welfare? To model such problem, a multiagent modelling technique such as cooperative games is necessary.

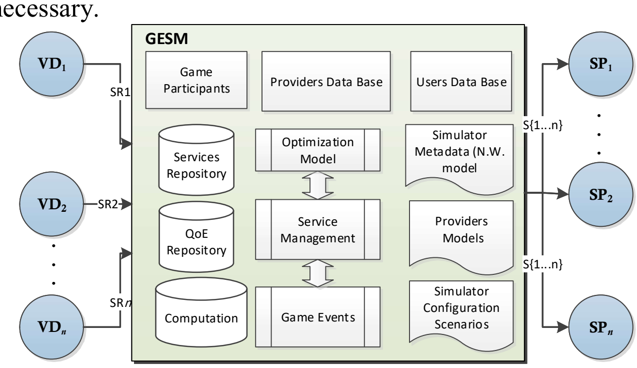


Figure 1: Overall system architecture

B) Service Providers Utility

User comfort-based QoE in vehicular cloud service provision has not been considered in today's literate besides our recent work that form a basis to the proposal in this paper. The majority of the studies tackle subjective criteria while user bias is mostly neglected.

In the proposed model, game theoretic approaches have been chosen to manage service provider's resources in order to provide efficiency and fairness at the same time. In such multiagent multiservice environment, selfishness is the common action which results in reducing all service providers' welfare. On the contrary, game theory assesses how the efficiency of overall system degrades despite the selfish behaviour of the players, and provides series of events and actions to improve overall system welfare. The game is also capable of avoiding that by improving cooperation among its players, and capable of incentivizing noncooperative players to cooperate.

An essential assumption is that all players are willing to cooperate and behave in altruistically manners as well as provide their response without common knowledge of the game outcomes or its players. This is to emphasize the concept of service provider's social welfare.

Given a system with a number of cloud providers and vehicle drivers, the potential participants can be clustered into sellers and buyers as shown in Figure 1. Sellers are capable of providing different types of services to different number of buyers (drivers). Each one of them has their own QoE reputation value for each type of different services, as follows, QoE $= \{ QoE_{VD}, QoE_{SP}\}$. Let S *and* $B \rightarrow \mathbb{R}$ denote the sets of sellers and buyers, respectively, given that S *and* B are the total number of sellers and buyers in the game. At some point in the game, derivatives of the sub-sets of S *and* B are produced, namely, S' *and* B', which represent the group of sellers who possess the requested services by the buyers. Each seller, S', provides a number of services defined by the following set, $R = \{1, 2, 3, \ldots, r\}$, where r is the total number services. Each seller $s_i \epsilon S$ has a service r_s^i available to provide to a buyer $b_j \epsilon B$ who is actively seeking this service, $r_b^j . X^i(j)$ is the total number of services provided from seller i to buyer j.

The QoE model adopted assumes that trusted third parties buy the services on behalf of the vehicle drivers and communication aspects are handled through them. Thus, trusted third parties time and resources considered within this model. A trusted third party's communication cost, C_{TTP}, represents the cost associated as a compensation per use time. $T_r^i(j)$ is the total number of third parties used between seller i and buyer j to buy a service r.

In order to balance the load of the requested services from the buyers to the sellers and provide fairness in distrusting services among service providers, GESM module ensures that all service providers are equally loaded depending on their system capacity and resource availability. Overloading one service provider over the others will introduce negative impact on its QoE reputation value. Moreover, managing total load of services achieves fairness and satisfies the game objectives by maximizing service providers' utilities while efficiently delivering requested services to their buyers.

Here, the objective is to maximize the total utility of all service providers while being efficient and fair. Fairness is the guarantee that all participants receive fair treatment while they are engaged in the game. Thus, social welfare function is a weighted sum of system total utilities [15] considering the QoE for each provided service.

Let $u_{QoE}(n) = [u_{QoE_1}^{r_1} + u_{QoE_2}^{r_2} + \cdots + u_{QoE_n}^{r_n}]$ denote the vector of summation of utilities. Thus, our constructed social welfare function is:

$$\Phi(u) = \left[X^i(j) * \log\left(\sum_{n=1}^{N} u_{QoE_n}^{r_n}\right)\right] - (T_r^i(j)) \tag{1}$$

C) Optimization Model

To formulate an optimization model that maximizes the total utility(Φ) and provide scalability, two functions are required, total cost function (C) and load function ($L(i)$). Load function is expressed by the index of seller, i, while cost function depends on the load function, as follow:

$$C(L(i)) = W_r^i \times L(i) \tag{2}$$

where W_r^i is the total number of services per seller i. GESM has all the information about each seller (i), their available services, their total service sale, and their service cost (prices). This knowledge is leveraged during the game to achieve fairness and maintain efficiency. Thus, the total cost of service r sold from seller i to buyer j is:

$$C_r^i(j) = X^i(j) \times T^i(j) \times C_{TTP} \tag{3}$$

The utility of seller (i) is a function defined from the set of sellers, S to the set of buyers, B. The function parameters are the amount of data provided by the sellers in S' to the buyers in B' and the cost of trusted third parties. At this point, we can define our seller's utility function ($u(x)$), as follows,

$$u(i) = \sum_{i=1}^{S\prime} (X^i(j) \times C_r^i(j) \times C(L(i))) \tag{4}$$

The ultimate goal of GESM is to distribute requested services among sub-group of sellers (S') in order to maximize the overall system utility. Thus, the optimization problem is defined by (5).

$$maximize \sum_{i=1}^{S\prime} (u(i)) \tag{5}$$

This optimization includes three metrics: The utilities of the buyers (B') and sellers (S'), and trusted third parties' reliability cost (C_{TTP}). A robust model has to ensure that service sellers are not overloaded during the optimization. To this end, the following constrains are applied:

$$\sum_{j=1}^{S} X^i(j) \leq r^n, \forall_i \in S' \tag{6}$$

The first constraint in (6) ensures that a seller cannot be loaded more than its limit of service capacity. In the equation, r^n is the maximum number of services that seller i can handle. S' is the sub-set of S, and the members of S' possess the requested services by B'.

$$X^i(j) = [0, r^n],\ X^i(j) \in \mathbb{N}, \forall_i \in S', \forall_j \in B' \tag{7}$$

The second constraint in (7) guarantees that services will be distributed fairly among sub-set of sellers S' given that they have the services available, and the service buyer j cannot buy from seller i the maximum number of services offered until all sellers in S' have equal utilities.

D) Service Management Game Model

So far, the system social welfare and an optimization model to maximize system utility have been presented; however, the game model still needs to be formulated between the system players. To this end, we have developed a cooperative game scheme between the potential players, more precisely, sub-set of the service sellers, S'. First, the types and roles of the players must be defined, as well as the possible interactions between them.

We consider two main players Vehicular Driver and Service Provider $\{VD_i$ *and* $SP_j\}$ participating in the game, each of which is independent and has preferences that represent their best interests. The expected outcomes can be defined by the following set formulated in (8).

$$\Omega_{outcomes} = \{ QoE_{VD}, QoE_{SP} \} \tag{8}$$

The proposed cooperative game receives the buyers' service request, finds the suitable service sellers, and shifts the service load among service sellers when needed. The selection procedure of the proper sellers and the afterward load shift perform according to the seller's utility status. A payoff function is associated to each service seller and depending on their cooperation, the game decides on the fair distribution of the rewards (i.e. payoffs). Thus, the payoff of player $i \in S'$ is denoted by: $O_r^i = u^i(r)$ for service r. The benefit of this game model that each seller is required to monitor their payoff function independently from other players' payoff. GESM stores all players' payoff private which also ensures player privacy.

With the concept of player payoff, we can show the interactions between the players for specific or multiple outcomes. A player in the game has to make a decision (action) as the outcome is resulted in by this action. The final result of all interactions between the players is the latest outcome of the game result, and it is based on the finalized payoff function of the game engine. Each player has only two possible actions to consider {A, R}, where *A* stands for *accept* and *R* stands for *reject* to cooperate. To consider accept or reject the game offer, each player has to calculate their payoff value. In order to do so, each player has to know its own share load of the service request and the total cost for this share using (2).

The game engine performs a number of interactions between players (i.e. S') to solve the service distribution problem. The payoff function for cooperative players during the game is calculated based on their actions. A player's action can be to reject but with potential other proposals for other games (i.e. service requests), those still receive some payoff for their future contributions. Each player is required to continuously report their resource availability, service availability, and current load to the game engine. Thus, game engine, has the sub-set of sellers and their available load, which will be used to coordinate the total payoff. The action of each player $i \epsilon S'$ is presented as a service vector $S(i) = [S^1(i),\ S^2(i), \ldots,\ S^n(i)]$, where $S^n(i)$, is the services provided from S'. So, the total payoff, P, of each player can be formulated as in (9)

$$P^i\big(S(i), S(-i)\big) = -u(i) \qquad (9)$$

Where $S(i)$ and $S(-i)$ represent the action selected by player $i \epsilon S'$, and all other player's actions other than i, respectively. The game engine leverages their centralized knowledge of players' resources to maximize their payoff. As more players cooperate and involve in the game, the load will equally be distributed. Consequently, this will lead to a positive impact on their total payoff and system utility ($u(i)$). This procedure is repeated in different stages while the game engine receives players' update about their resource availability. By applying the new information about $S(i), S(-i)$, each player's response can be calculated by maximizing P^i. The service management selection procedure can be optimized by (10) and using Algorithm 1.

III. Performance Evaluation

A) Simulation Settings

To evaluate the performance of the proposed service management game model, we used Network Simulator 3 (NS-3) and CPLEX 9.0 optimizer to solve the problem. The purpose of evaluating the proposed model is to investigate the worthiness of using cooperative game model to manage service provision in vehicular cloud. We would like also to determine how the proposed algorithm affects service provider's social welfare, and measure the level of complexity to calculate system payoff with the proposed QoE framework.

The simulation employs the IEEE 802.11.p communication protocol. In the following set of simulations, each was repeated 10 times with a rang utility of 5 to 50 vehicles requesting services from a range of 5 to 40 service providers. We assign the number of services to each provider based on uniform distribution within the set {2, 4, 5} so that the providers are expected to provide them with limited amount resources. The sellers' resources and level of capacity change depending on the number of services they can provide. On the other hand, the buyers request different types of services during the game. We introduce two types of provisioning services in two groups: Lightweight Services (LS): *Gas price*, *Traffic conditions* and *Weather conditions*, and Heavy Services (HS): *Audio streaming* and *Video streaming*. The service cost ($C_r^i(j)$) varies from one service to another. Lightweight services are set to 0.5 to 1.0 cost/time unit whereas heavy services are worth 2.0-3.0 cost/unit time. Since we are using the user comfort-based QoE framework, the trusted third-party cost factor (C_{TTP}) is set to 0.25 cost/unit time.

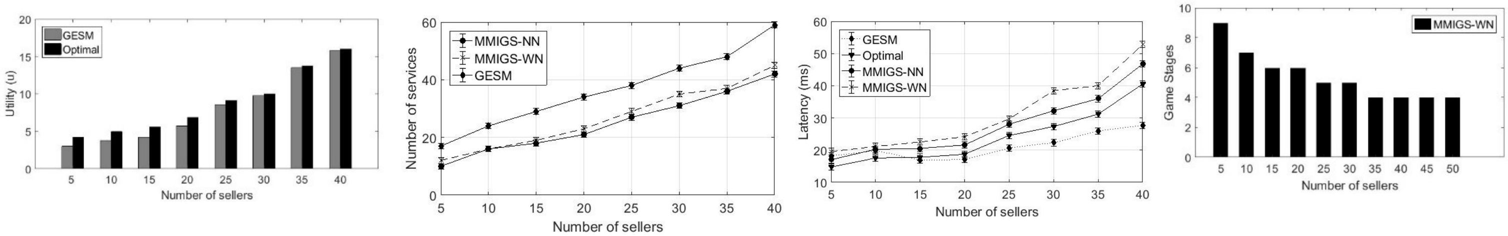
$$maximize\big[\, P^i\big(S(i), S(-i)\big)\big], \forall S^i(j) \in S(i) \qquad (10)$$

Algorithm 1: Service management for player $i \in S'$

Input: Service providers set (S'), vector of resources held by S', and vector of service request $S(i)$.
Output: $S(i), S(-i)$ for every player in S'.
Begin
1: Receive service requests set, R, from service buyers.
2: Extract S' sub-set from S.
3: Calculate current payoff for each player $i \in S'$.
4: **loop**:
5: **if** ($\sum_{i=1}^{N} O_r^i > 0$) **then**
6: let $pl = pl + \{O_r^i\}$;
7: **else**
8: $pl = \{\emptyset\}$;
9: **end if**
10: Compare set of payoffs, pl;
11: Update $\big(S(i), S(-i)\big)\ \forall i \in pl$, accordingly;
12: **if** ($S(i)$ == accept) **then**
13: Compute $u(i)$ using (4);
14: Compute optimization of (12);
15: Update $\big(S(i), S(-i)\big)\ \forall i \in pl$;
16: **else**
17: **if** ($S(i) == reject$) **then**
18: Update pl set, accordingly;
19: Go to line 12;
20: **end if**
21: **end if**
22: **end loop**

B) Simulation results

In the first set of simulations, we compared the utility of each service provider under the proposed algorithm and optimization model (i.e. Optimal). Figure 2.a illustrates that the utility has a gap of ~7.5% of the proposed algorithm to the optimal solution within the first five groups only. However, the gap starts to be reduced and becomes negligible in the rest of the groups. In other words, the optimal model outperforms the proposed algorithm under a sparse network, and tends to introduce more of equal utilities when the node density in the network increases. This is due the fact the algorithm has less players cooperating at the beginning of the game while heavy load of service requests arrive for limited amount of resources. However, once $pl = \{O_r^i\}$ starts to increase, it gives the game more resources which leads to increase their utilities as shown in groups 20 to 35.



(a) (b) (c) (d)

Figure 2.a: Utility of different sellers under GESM and optimal. Figure 2.b: Number of services under different approaches. Figure 2.c: Average latency of different approaches. Figure 2.d: GESM number of game stages

In the second set of simulations, we measure the number of services that each service provider handles under the proposed game model and the models presented in [6], namely an Interaction Game System with and without negotiation, MMIGS-NN and MMIGS-WN, respectively. Figure 2.b clearly shows that GESM approach outperforms the other two approaches at all network densities. The 95% confidence interval shown. In addition, GESM improved the number of the services that each service provider can handle. The algorithm perceives all the components of the system and utilizes provider's recourses efficiently while fairly distributing the load among them leading to an improved overall system performance. In MMIGS approaches, players participate in the game non-cooperatively which explains the gap between the number of services handled at under all network densities.

In the third set of simulations, we measured the average latency of game model and compared it to the previous approaches (i.e. MMIGS-NN, MMIGS-WN, and optimal model), as shown in Figure 2.c. The results demonstrate that the game with negotiation (MMIGS-WN) takes longer time comparing to the other models. These approaches include two layers of negotiations over the requested services, and this explains the cause behind such latency. GESM under the sparse node density in the network, appears to have similar latency compared to the other approaches. However, once the number of service sellers (i.e. game players) increases in the game the average time to find a suitable service provider increases. Indeed, the number of services is increased as well with the increase in the number of players.

In the last set of the simulations, we tested the duration of the game to stabilize (converge) under various numbers of the game players, as shown in Figure 2.d. The results show that large number of stages with few number of the players, and the number of the stages starts to decrease once the number of the players increases. It is worthwhile noting that the game starts to converge in 4 stages when the number of the players is around 35.

IV. CONCLUSION

In this paper, we have proposed a game theoretical distributed model to manage fair provisioning services among service providers in a vehicular cloud network setting. The proposed game model maximizes participants total payoff. The game distributes the services among the best sellers considering their resource availability, current load, and total payoff. As a benchmark, an optimization model to maximize the total system utility has also been proposed. Through simulations, we have shown that the proposed game achieved very similar results to the optimized model in terms of utility (more than 85% of total utility achieved when few players participated, and increased to ≈99% when the number of the players increases by 50%.). In addition, the game clearly demonstrates that a cooperation technique enables the players to handle higher number of services when compared to a non-cooperative setting (i.e. 22% increase in terms of number of services).

REFERENCES

[1] M. Soyturk, K. N. Muhammad, M. N. Avcil, B. Kantarci, and J. Matthews, "From vehicular networks to vehicular clouds in smart cities," in *Smart Cities and Homes*, M. S. Obaidat and P. Nicopolitidis, Eds. Boston: Morgan Kaufmann, 2016, pp. 149–171.

[2] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 241–246.

[3] S. Olariu, M. Eltoweissy, and M. Younis, "Towards autonomous vehicular clouds," *Transactions on Mobile Communications and Applications, Springer*, vol. 11, no. 1, 2011.

[4] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A Survey on Vehicular Cloud Computing," *Journal of Network and Computer Applications, Academic Press Ltd.*, vol. 40, pp. 325–344, 2014.

[5] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "An Auction-driven Multi-objective Provisioning Framework in a Vehicular Cloud," in *Proceedings of the 2015 IEEE GlobeCom Workshop on Cloud Computing Systems, Networks and Applications (CCSNA)*, 2015.

[6] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "Multiagent/Multiobjective Interaction Game System for Service Provisioning in Vehicular Cloud," *IEEE Access: Advances in Vehicular Clouds*, vol. 4, no. 1, pp. 3153–3168, 2016.

[7] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "Trusted Third Party for Service Management in Vehicular Clouds," in *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 1–6.

[8] W. Saad, Z. Han, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition Formation Games for Distributed Cooperation Among Roadside Units in Vehicular Networks," *IEEE J.Sel. A. Commun.*, vol. 29, no. 1, pp. 48–60, Jan. 2011.