MAS based Resource Provisioning in Vehicular Cloud Network

Mousumi Paul
Research Scholar, Department of CSE, National Institute of Technology, Durgapur, West Bengal, India.

Goutam Sanyal
Professor, Department of CSE, Department of CSE, National Institute of Technology, Durgapur, West Bengal, India.

Abstract
Modern day’s vehicles are equipped with various features, like sensing space, computing capabilities. They have the flexibility to communicate each other as well as with Road-side unit with wireless transceiver. Therefore by the combination of such resources meaningfully will create a tremendous influence on the domain of communication network. The underutilized vehicular resources can be rented to customers as a conventional cloud infrastructure. In this position paper, we propose a multi-agent system based information exchange in vehicular cloud network. The problem we address is that to select a useful solution from a number of information gathered from other vehicles as well as from RSU is intrinsically difficult. Our (proposed) solution is a three-layer architecture which integrates cooperative agents (responsive vehicles), autonomous agent (RSU) and learning agents, the idea being to select a smart solution by combining information from all three agents using multi-objective fitness function. NSGA II algorithm has been used as a multi-objective method.

Keywords: MAS, VANET, Multi-Objective Function, NSGA II, RSU.

Introduction
The After a long period of time of providing solutions in the domain of infrastructure based wired networks, several commercial applications cropped up which required providing services to clients on the go. This basic thirst of convenience of use (anywhere anytime) and dissemination cost led to the development of Wireless Ad-Hoc Networks. Different Ad-Hoc Networks have different initial contexts however they all share one common characteristic: no fixed infrastructure. The absence of fixed infrastructure made the network operations higher dependence on individual nodes than in wired networks. Modern day’s vehicles are equipped with sensing power, computing facilities, and storage space and communication capabilities. As a result there is increasing demand for availing network services through the wireless Vehicular Ad-Hoc Network (VANET). This rise in demand has opened up a new avenue of Intelligent Transportation Systems (ITS) and business for industries and at the same time, it has brought many challenges for the network designers and researchers. At early stage of VANET technology the resources of vehicles were used for road-safety oriented purpose. However the proper usage of vehicles resources is still unexplored. The very recent research on Vehicular Ad-Hoc Network presents novel approaches which combine vehicle resources to form large mesh cloud infrastructure. The different resources like net connections and storage facilities, computing power and other advance application software are pooled together which will provide various services to the drivers on the road or these resources can be rented to the customer as a conventional cloud computing paradigm. The technological feasibility and economical viability will make vehicular cloud a more popular way of information exchange in future communication network. VANET deployment in highly stressed networks, the design of distributed and cooperative agent-based systems is of interest. This is primarily due to the flexible problem solving approach offered by MAS, which provides a robust solution to address the challenges faced in VANET environments. Our emphasis is on the prospective applications and significant aspects of research challenges. This paper presents an approach which combines multi-agent technology and cloud computing in vehicular ad-hoc network to utilize the vehicular resources in its full extent. The self-organization characteristics and the interaction between a large numbers of self-organizing vehicles provide distributed teamwork-based solutions to complex problems. In addition, multi-agent system (MAS) technology is well accepted for dynamic, distributed problem solving in which distributed software agents are able to both sense and act within complex environments [3] and VANET possess all these characteristics. However, at present, MAS design and performance tradeoffs in disruptive and potentially vehicular ad-hoc networks are largely unexplored. VANETs can manifest self-organization as well as self-steering; having some control paradigms can share knowledge using any agreed languages, within the constraints of the system's communication protocol to achieve a common improvement. Further it is challenging to choose a best solution among various solutions gathered from the various agents fulfilling multiple objectives. In this work we have used NSGA II algorithm to find the best fit among two or more conflicting choices. The remainder of this paper is organized as follows. In Section II, we discussed some related work on multi-agent system and vehicular cloud. Section III explains the architecture of the system. Section IV explains the
optimization method. In section V, performance evaluation has been carried out. Section VI concludes the paper.

Related Work
Various research works have been carried out on the applicability of VANET and MAS as a platform for Ad Hoc Network. Jun Luo and J.-P. Hubaux [1] described a system called as Ad Hoc City, in which wireless devices are mounted on mobile vehicles like cars, bus etc. This is a multilayer wireless ad hoc network routing architecture for general-purpose wide-area communication. Justin W. Dean, Joseph P. Macker and William Chao [2] developed a model that enables joint MAS and MANET research to be carried out together by applying multicast forwarding. [3] Proposed a vehicular information ad-hoc network that consists of three tier network architecture using Multi-Agent System (MAS) technology. In [4] addresses the issues pertaining to MAC in highly dynamic automotive networks and proposed a distributed positioning algorithm, called the kernel algorithm that reduce latency and provide reliable communication. Fang, Z., Liu, X. developed [5] an intelligent transport system framework based on multi-agent paradigm called CSCW (Intelligent Computer Supported Cooperative work). Md. El Amine Ameur and Habiba Drias developed Multi-agent system based Traffic Congestion Management in VANET environment [6]. Hall, S. and Draa, B. C. [9] proposed a model of teamwork used in multi-agent systems as a decentralized alternative to previous coordination centralized on the platoon’s leader and outline its benefits using collaborative driving simulation scenarios. Meanwhile the above mentioned works explores the trend to use vehicles underutilized resources as large distributive dynamic resource infrastructure. Recently in a few research works it has been seen that cloud computing and VANET are combined together. S. Olariu, M. Eltoweissy and M. Younis [13], proposed the concept of Autonomous Vehicular Clouds (AVC) to exploit the under-utilized resources in VANET. In [14] a platform as a Service (PaaS) model is designed to support cloud services for mobile vehicles. The work in [15] proposed an architecture of Vehicular Clouds (VC), Vehicles using Clouds (VuC) and Hybrid Clouds (HC). Hence the critical factor in this system is to select an optimal result received from a number or information received by other communicating nodes.

In this paper, we have proposed a MAS based three tier architecture which helps drivers to gain proper information for a service and then to select optimal response by using multi objective optimization function.

Proposed Architecture
In VANET vehicles are considered as multiple interacting intelligent agents within an environment. The composition of such agents can be used to solve problems that are difficult for a monolithic system to solve. A general multi-agent scenario is shown in Figure 1. In multi-agent system three types of agents exists. Autonomous agent, Cooperative Agent and Learning agent. Autonomous agents are the individual vehicles that are at least partially independent, self-aware, and autonomous. Cooperative Agents are the nearby vehicles to whom the requests are sent for the required information. It may be vehicles or nearby road-side unit (RSU). Learning agents are intelligent once. They learn or infer from the knowledge to achieve their goals. Figure 2 shows how Cooperative, Autonomous and Learning agent, all together collaborating each other to form a Smart Agent, i.e. the ultimate benefit of MAS.

Figure 1: General scenario of Multi-Agent System in VANET

Figure 2: Benefits of Multi-Agent System

Figure 3 shows the proposed hierarchical vehicular cloud architecture that consists of two interacting layers: vehicular cloud (VC), roadside cloud (RC).

- Vehicular cloud (VC): A group of vehicles establish a cloud. The vehicles of a group cooperatively create a vehicular cloud.
- Roadside Cloud (RC): A group of RSU forms a mesh roadside cloud. RSU are the gateway to the internet. By RSU vehicles can be connected to outer world.

In cloud computing paradigm IAAS (Infrastructure as a service), PAAS (Platform as a service) and SAAS (Software as a service) are the three service models. Vehicle cloud architecture also contains these three service models.
IAAS (Infrastructure as a service):
The wireless sensors and electronic equipments present in the vehicles collect the road information, in-car information and traffic condition etc.

Platform as a service):
The information collected by the vehicles is passed to SAAS via PAAS. In a parking lot, vehicle’s resources are basically wasted. They are not in use. At this time the group of vehicles forms a cloud whose resources can be rented for computing purpose.

SAAS (Software as a service):
Network service has a close relation with SAAS. A vehicle can run their application on another vehicle through providing API (Application Programming Interface). Again the information obtained, give useful services such as traffic congestion, fuel, highway management, navigation etc.

Multi-agent designs improve the potentiality of operations in VANET platform. With the proposed architecture, drivers of individual vehicles are capable to make quick responses to requested services sent by other vehicles or on situation like traffic congestion or accident zone. Therefore drivers of requested vehicle would have enormous information to take appropriate decision.

Optimization Method
In this paper the multi-objective optimization approach has been used to schedule the vehicular resources. Multi-objective optimization is an area of multiple criteria decision making that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. Multi-objective optimization has been applied in various fields of science, including economics, logistics and engineering where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Suppose a vehicle sending a request to nearby vehicles for the information of nearest Fuel Station. Nearby vehicles then send information based on their location to the requesting vehicles. The requesting vehicle then receives \( n \) number of information. The challenge here is to select an optimized solution from this information. Received Information contains different criteria. For example, some vehicles may have sent nearest Fuel station data which is nearest based on its own current location which may not be nearest for the requesting car. Or the location may not be on the route of destination where the requesting car is travelling. Or the cost to reach the fuel station is high due to the constraints of lane change (i.e. the road may be one way traffic). Therefore the objective of the solution should be shortest distance from the current location, low cost and on the location of the fuel station should be on the route on which the car is moving. In this paper we have considered three kinds of resources like storage space, computing capacity and bandwidth. In some situation the number of virtual resources (requested resource) may be greater than the physical resources (resource present in vehicles). So the objective is to find the better match to map the virtual resources into physical resources. The solution to map the virtual resources into physical resources is a NP-hard problem. Suppose \( n \) number of virtual resources say \( V_i \) \((i = 1, 2, \ldots, n)\) have to be mapped in \( m \) number of physical resources \( P_i \) \((i = 1, 2, \ldots, m)\). Therefore for each resource there are \( n \) number of choice available. Hence the complexity \( O(3\text{n}) \) is a hard problem. Since modern vehicle are equipped with various resources, like memory, CPU for computation and bandwidth etc. So the objective is to balance the load of these features. Now we define the virtual resources with three different features as \( V_i \) \((m_i, \text{cpu}_{vi}, b_{vi})\) and physical resources as \( P_i \) \((m_p, \text{cpu}_{pi}, b_{pi})\), where \( m \) represent memory, \( \text{cpu} \) for CPU and \( b \) is bandwidth. Therefore the load function of these features are written as

\[
L_{total,m} = \frac{\sum_{i=1}^{n} m_{vi}}{\sum_{i=1}^{m} m_{pi}} (1)
\]

\[
L_{mi} = \frac{\sum_{i=1}^{n} m_{vi}}{m_{pi}} (2)
\]

where \( m_{vi} = \begin{cases} m_{vi} & a_i = i \\ 0 & \text{others} \end{cases} \)

Therefore balance of load function of memory in physical resources is given as

\[
P_m = \sqrt{\sum_{i=1}^{n} \left(L_{mi} - L_{total,m}\right)^2} (3)
\]

Similarly the balance function of load of CPU and bandwidth in physical resources is shown as:
Here each physical resource (vehicles) has an ID. At a time instant the virtual resource which is scheduled to physical resources are represented as $V_{ai}$ where $V_{ai} \in \{P\}$. In this work we have used NSGA II to search the best physical resource where a virtual resource will be mapped. However, it is time-consuming to solve the Pareto non-dominated solutions. At first the population is sorted according with the first object function. Then a binary tree is constructed and binary tree sorting algorithm is employed. Binary tree is used because it is impossible that an individual be dominated by the individual after it [17]. The algorithm of NSGA II is written as follows:

**Algorithm 1:**

popul _ation _\Rightarrow_ Initialize _population{population,
problem}_
Evalu _e _against _objective _function{population}
Non _do _min _ated _sort{population}
select _ed _individuals _\Rightarrow _select _from _parentsByRank{population,population}
children _\Rightarrow _crossover _and _mutation{selected _individuals,P,population}
while(\text{StoppingCriteria})
Evalu _e _against _objective _function{children}
end _while

The non-dominated set is obtained from the binary tree as using algorithm 2:

**Algorithm 2:**

for(Q := Tree; Q != null; Q := Q.left){
    Assign Q in Front;
}
Fronts := Front;
while(size(Fronts) < Half of tree nodes){
    for(each individual Q in Fronts, -1){
        for(Q := Q.right; Q != null; Q := Q.left){
            if (Q is not do min ated by Fronts){
                Fronts := Q;
            }
            else{
                left := Q;
            }
            for(Q is in left){
                if (Q is not do min ated by Fronts){
                    Fronts := Q;
                }
            }
        }
        if(Q do min ated by P in Fronts){
            left := P
        }
    }
}
for(P to population size Fronts){
    parents := P;
    end _for
}
End _while

The non-dominated set is obtained from the binary tree as using algorithm 2:
The flow chart of the whole process is shown in figure 4.

**Figure 4: Flow of work**

Performance evaluation
The experiment has been done using Matlab. Parameters set for the experiment is shown in table 1.

### Table 1: Parameters of Experiment.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>100</td>
</tr>
<tr>
<td>max-iteration</td>
<td>1000</td>
</tr>
<tr>
<td>Crossover factor</td>
<td>0.8</td>
</tr>
<tr>
<td>mutation factor</td>
<td>$\frac{1}{L}$ where $L$ is length of gene</td>
</tr>
</tbody>
</table>

Since VANET is highly dynamic, we have chosen the resources according to their stability factor. Therefore when computing resources are requested we will choose that individual whose CPU balance factor is more. But again different static can be choosing based on the demand. Table 2 shows the result of non-dominated set obtained using NSGA II.

### Table 2: The non-dominated set obtained by NSGA II

<table>
<thead>
<tr>
<th>Solutions</th>
<th>CPU Balance Degree</th>
<th>Memory Balance Degree</th>
<th>Bandwidth Balance Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>2122121056</td>
<td>3.1768</td>
<td>0.3456</td>
<td>1.1282</td>
</tr>
<tr>
<td>2512342501</td>
<td>0.1316</td>
<td>1.5412</td>
<td>1.6685</td>
</tr>
<tr>
<td>5210532201</td>
<td>1.1605</td>
<td>1.8787</td>
<td>0.7617</td>
</tr>
<tr>
<td>0853187242</td>
<td>3.6868</td>
<td>0.1528</td>
<td>2.1298</td>
</tr>
<tr>
<td>1654298112</td>
<td>1.2351</td>
<td>0.7793</td>
<td>4.8721</td>
</tr>
<tr>
<td>1253111245</td>
<td>0.0816</td>
<td>0.2455</td>
<td>2.9080</td>
</tr>
<tr>
<td>0954245007</td>
<td>2.7826</td>
<td>0.1066</td>
<td>0.2167</td>
</tr>
<tr>
<td>0412012552</td>
<td>1.3886</td>
<td>1.0227</td>
<td>0.4454</td>
</tr>
</tbody>
</table>

We have compared the result with Rank algorithm and Random algorithm. Table 3 shows comparison result.

### Table 3. Compared NSGA II with other algorithms by solving the example in this dissertation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Static</th>
<th>Solution</th>
<th>CPU Balance Degree</th>
<th>Memory Balance Degree</th>
<th>Bandwidth Balance Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA II</td>
<td>CPU</td>
<td>214852784</td>
<td>0.26</td>
<td>1.42</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>Strategy</td>
<td>1748230355</td>
<td>3.78</td>
<td>0.34</td>
<td>0.973</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>2169002446</td>
<td>1.28</td>
<td>0.64</td>
<td>0.235</td>
</tr>
<tr>
<td></td>
<td>Bandwidth</td>
<td>5048022114</td>
<td>0.361</td>
<td>0.29</td>
<td>0.39</td>
</tr>
<tr>
<td>Radom</td>
<td>CPU</td>
<td>3640132001</td>
<td>19.37</td>
<td>19.54</td>
<td>141.81</td>
</tr>
<tr>
<td>Radom</td>
<td>Strategy</td>
<td>0335450415</td>
<td>7.220</td>
<td>1.39</td>
<td>5.19</td>
</tr>
<tr>
<td>Static</td>
<td>CPU</td>
<td>1114352995</td>
<td>2.48</td>
<td>2.28</td>
<td>2.61</td>
</tr>
<tr>
<td>Rank</td>
<td>CPU</td>
<td>3640132001</td>
<td>19.37</td>
<td>19.54</td>
<td>141.81</td>
</tr>
</tbody>
</table>

From Table 3 it can be seen that NSGA II gives better solution in terms of number of solution and resource balance factor.
Conclusion
In this paper an attempt has been made to use the underutilized vehicular resources as a conventional cloud infrastructure. The combination of multi-agent system and cloud computing in VANET meets the future requirement of wireless network. Using NSGA II to solve this scheduling and the performance of NSGA II is improved by introducing binary tree sorting. At last, the result is compared with other two algorithms: Random Algorithm and Rank Algorithm. The result shows the efficacy of the algorithm.

References


