Vehicular Communications 4 (2016) 47–56

Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom

Virtual machine migration and management for vehicular clouds

Tarek K. Refaat [a], Burak Kantarci [b], Hussein T. Mouftah [a,*]

[a] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada
[b] Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY, USA

ARTICLE INFO

Keywords:
Cloud computing
Data center management
Vehicular ad hoc networks
Vehicular clouds
Virtual machine migration

ABSTRACT

Vehicular Cloud Computing is a growing research field which consolidates the benefit of cloud computing into vehicular ad hoc networks. However, few studies address vehicles as potential Virtual Machine hosts. Due to the rapidly changing environment of a vehicular cloud, a host can easily change or leave coverage. As such, Virtual Machine Management and Migration schemes are necessary to ensure cloud subscribers have a satisfactory level of access to the resources. This paper proposes several Vehicular Virtual Machine Migration (VVMM) schemes: VVMM-U (Uniform), VVMM-LW (Least Workload), VVMM-MA (Mobility Aware) and MDWLAM (Mobility and Destination Workload Aware Migration). Their performance is evaluated with respect to a set of metrics through simulations with varying levels of vehicular traffic congestion, Virtual Machine sizes and levels of load restriction. The most advanced scheme (MDWLAM), takes into account, the workload and mobility of the original host as well as those of the potential destinations. By doing so a valid destination will both have time to receive the workload and migrate the new load when necessary. The behavior of various algorithms is compared and the MDWLAM has been shown to demonstrate best performance, exhibiting migration drop rates that are negligibly small.

1. Introduction

The incorporation of cloud computing into various aspects of technology is a growing trend, both in terms of depth and breadth of spectrum. The most common application users come across is typically the cloud storage. More complex applications include cloud-based software or processing [1]. With the progress in the research of the realization of the Internet of Things (IoT), cloud computing becomes more appealing as devices with network interfaces increase [2]. Vehicles are an example of such devices where interconnectivity is heavily studied. Typically, the field of Vehicular Ad Hoc Networks (VANETs) studies the communication from vehicle to vehicle (V2V), vehicle to infrastructure (V2I) and infrastructure to vehicle (I2V), with the purpose being to exchange safety or traffic information [3]. However, a new paradigm considers utilizing vehicles for cloud computing, effectively creating a Vehicular Cloud (VC) [4–7].

As vehicle designs incorporate an increasing amount of electronics, resources such as storage space and processing power, make the VC more realizable every day [8,9]. A vehicular cloud can make use of many of these often underutilized resources, operating like a typical cloud data center [10,11]. The unique setting of a vehicular environment facilitates services including Entertainment-as-a-Service (ENaaS), Traffic-Information-as-a-Service, Network-as-a-Service, and Storage-as-a-Service, among others [12–16].

In order to maximize flexibility, when allocating hardware resources to cloud users, Virtual Machines (VMs) are used. A VM allows a user to set up an environment and run their unique processes, without worrying about the physical resources involved. The resources are handled by hypervisors that, in turn, isolate VMs, virtually, although several may share the same physical resource, rendering each VM (and its corresponding user) oblivious of its neighbors. Moving the VM from one physical location to another is simple and potentially seamless. Such VM Migrations (VMMs) can be triggered to handle hot-spots or for resource load balancing [12].

There are several inherent obstacles that VMM schemes must address, and for the traditional cloud, there are several approaches available [13,17]. Aside from the typical issues of handling migration triggers, the VC has unique obstacles to address. The main aspect that is almost exclusive to the VC is the dynamicity and mobility of the resources, namely the vehicles.

It is worthwhile noting that while studies propose vehicles as sources of data for the VC, there is a limited number of studies addressing the use of vehicles as actual VM hosts [18]. This study focuses on vehicles as VM hosts and specifically, the VMM algo-

* Corresponding author at: School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, Ottawa, ON, K1N 6N5, Canada. Tel.: +1 613 562 5800x2173.

E-mail addresses: tarek.k.r@ieee.org (T.K. Refaat), bkantarc@clarkson.edu (B. Kantarci), mouftah@uottawa.ca (H.T. Mouftah).

http://dx.doi.org/10.1016/j.vehcom.2016.05.001
2214-2096/

List of symbols

T_s	time needed to migrate source workload	s
T_d	time needed to migrate destination workload	s
T	simulated time	s
d_t	total dropped VM workload, at time t in the grid	MB
s_t	total successfully migrated VM load, at time t, throughout the entire grid	MB
v_t	total VM load, successfully migrated to the infrastructure, at time t, throughout the entire grid	MB
l_t	total VM load hosted on all the vehicles currently the grid, at time t	MB
l_m	maximum VM load capacity the grid can offer, based on the expected average number of vehicles at the congestion level simulated	MB/s
ul_t	total upload bandwidth utilized, at time t, by all vehicles throughout the entire grid	MB/s
ul_m	maximum upload bandwidth available in the grid	MB/s
dl_t	total download bandwidth utilized, at time t, by all vehicles throughout the entire grid	MB/s
dl_m	maximum download bandwidth available in the grid	MB/s
P_t	probability, at time t, that a packet is dropped due to interference or connection instability	
P_{max}	maximum value for P_t (set at 0.5)	
Int_t	normalized interference due to concurrent migrations at time t	
CQ_t	normalized connection quality between source and destination, based on distance between them, at time t	
SD_t	distance between source and destination	
SD_m	maximum value for SD_t, the diagonal distance from one corner of the grid to the other	

rithms for such a VC. As the vehicular VM hosts move, the need for a VMM scheme arises. Several Vehicular VMM (VVMM) schemes are proposed, tested and compared against a generic benchmark algorithm. The VC paradigm proposed involves the vehicles (VM hosts) and Roadside Units (RSUs), linked to the cellular infrastructure.

This paper presents the following VVMM algorithms: Uniform (U), Least Workload (LW), Mobility Aware (MA), Mobility and Destination Workload Aware Migration (MDWLAM). The VVMM-U algorithm is the benchmark, whereby migration destinations are selected using a uniform likelihood. The VVMM-LW selects destinations based on those with the least workload. The mobility of the destination is taken into account in VVMM-MA and finally, the destination workload is also considered for MDWLAM. The various algorithms are analyzed using various metrics, varying VM sizes and vehicular traffic congestion levels. The MDWLAM scheme achieves negligibly small migration drop rates and infrastructure migrations that are negligibly small, outperforming the other proposed algorithms.

The rest of the paper is organized as follows. Section 2 summarizes related studies for VC paradigms. Section 3 introduces the VVMM algorithms. The performance of the various algorithms is evaluated in Section 4 and the paper is concluded in Section 5.

2. Related work

The underutilized resources in modern vehicles make a VC quite appealing. As opposed to investing significant capital in a traditional data center, vehicle resources can be leased to and managed by a cloud administrator [19]. The cooperation of vehicles will involve communication issues due to a dynamic environment, network congestion, energy efficiency, and latencies, as well as security and privacy limitations [20]. However, should a VC be realized, there will be a significant benefit due to the existing underutilized resources.

Initially, studies focused on avoiding mobility issues by targeting parked vehicles [21,22]. One study considers avoiding communication issues also by implementing a VC using an airport parking lot. Vehicles parked for relatively long periods of time can be connected to a wired network via ports in their parking spots, effectively creating a traditional data center throughout the parking lot [12]. Such an approach is quite simple to realize as it avoids any wireless communication issues as well as the more challenging issue of mobility.

As research progressed, mobile vehicles for VCs were considered. While the complexity of realizing such a VC increases, it is more appealing. The number of vehicle groups in motion is far greater (and more common) than large parking lots like airports [23]. Such VCs will necessitate wireless V2V, V2I and/or I2V communication. The infrastructure involvement can decrease the complexity of administration as it is a static element of the cloud [24].

A generic model for VANET cloud computing is presented in [6]. The model includes two different types of VC models: permanent and temporary. In the permanent cloud, the majority of vehicles involved are stationary. The mobile vehicles, forming a sub-cloud are the temporary cloud. However, it is important to note that there is always a heavy reliance on the RSU. It was the only gateway for vehicle to vehicle interaction. The authors coin the term VANET-Cloud to describe their system, separated into three layers: Cloud, Communication and Client [6]. Finally, the authors report the future challenges a VANET-Cloud must address including security, data aggregation, energy efficiency and coordination between sub-clouds.

For many studies, vehicles are considered cloud subscribers or data sources for the cloud. Traffic congestion information, video surveillance on public transport, cooperative file downloading and data mining are all examples of potential vehicle utilizations [25]. However, it is important to consider the possibility that vehicles can act as VM hosts.

As VM hosts, VMM schemes are necessary to maintain a quality of service for cloud subscribers. While there are differences between the traditional cloud and the VC, it is important to consider existing VMM schemes for the traditional cloud, such as hot/cold migrations, black-box and gray-box migrations [26–28]. They serve as a starting point for VVMM schemes, albeit with several modifications. The optimization of VMM schemes is presented in [28]. Mainly, the study addresses bandwidth optimization, by determining the exact VMM scheme suitability based on various possible cases. The authors analyze several schemes in detail and summarize the pros and cons of each based on different metrics and overheads [28]. The study also covers optimizing energy efficiency and also storage efficiency for VMM schemes.

One of the few studies addressing the VC with vehicular resources being used for the cloud is presented in [5]. The study describes the job assignment to the various (mainly parked) vehicles, and more importantly incorporating fault-tolerance. This is specifically to address the dynamicity of the resources, as cars may unpredictably leave the cloud [5]. In order to minimize the failure of assigned jobs, due to departing cars, a single job is assigned to two separate cars. The method then involves using periodic checkpoints that repeatedly check if the cars are still available. As such, should a car leave before a job is complete, another car will be assigned to continue from the last successful checkpoint [5]. Another study addresses the contention between vehicles, competing for migration [29].

3. Proposed algorithms

In cloud systems, VM migration is a critical event, and it can be triggered by various goals such as prevention of hotspots, load balancing, and maintenance among others. In vehicular clouds, mobility and ubiquity of resources is the main trigger of VM migration. Due to the dynamicity of a vehicular data center, there is a need to address several inherent and inevitable situations that jeopardize the cloud users' access to the resource. If a vehicular cloud is to truly emulate a conventional data center, with vehicles acting as VM hosts, the data will be segmented into several geographical regions. A certain grid of streets, falling in range of a group of cooperating wireless base stations, will be hosting a set of data that will remain in the same geographic area. However, as a vehicle hosting a VM in such a region leaves the grid, it must keep the data within the grid – which introduces the main need for VM migration.

Ideally, vehicles go off the grid migrating their VMs to other vehicles that are still in the grid. Thus, grids experience a fluctuation of resources, as some vehicles leave and others enter. In order to retain the data in the grid, vehicles must accomplish VM migration before exiting. Should they be unable to migrate to another vehicle, they must have it migrate to the infrastructure to ensure a client will still have access. Several algorithms are proposed next, to handle the migration process.

Algorithm 1, presents the pseudocode that forms a basis for all VM migration algorithms for vehicular clouds. The goal is to select a migration destination for an *exiting* vehicle, in order to avoid any data loss. It is worthwhile mentioning that data loss refers to the situation where a vehicle cannot shift its workload to other hosts (i.e., vehicles) in the grid and has to store it within the roadside infrastructure. Although the data must be retained in the grid, should no potential destination vehicle for migration can be found, the source vehicle will have to upload the VM to the infrastructure. This could entail storing the VM at a local RSU, or it could mean uploading the VM to the Internet cloud. While the process outlined in Algorithm 1 is valid for all the algorithms proposed here, each has unique *search criteria* to be discussed next.

There are 3 proposed algorithms, benchmarked by the generic Algorithm – Vehicular Virtual Machine Migration with Uniform host selection (VVMM-U). Each is explained below.

3.1. VVMM-U

The benchmark algorithm used to gauge the performance of the proposed algorithms is VVMM-U. This follows the process described in Algorithm 1. The search criteria involve merely selecting a vehicle from the pool of candidates, initially searching at the street level, and at the grid level if there are no vehicles sharing the street. This selection process does not take into account any of the factors that could increase the chances of a successful migration. The migrating vehicle treats all potential candidates equally. The possibility that the selected migration destination may not have enough space, or enough time to receive the migration, is unpredictable and totally randomized.

Algorithm 1 General VM migration pseudocode.

```
1: {src: migration source} // exiting vehicle
2: {dst: migration destination} // result of algorithm
3: Begin
4: pool: pool of potential candidates
5: let pool = vehicles in same street as src
6: if (|pool| > 0) then
7:    let pool = vehicles in pool satisfying search criteria;
8:       if (|pool| > 0) then
9:          let dst = V, vehicle in pool, nearest to src
10:      else
11:         let dst = I// vehicle will attempt to migrate to infrastructure
12:      end if
13: else
14:    let pool = vehicles in the same grid
15:       Go to line-6
16: end if
17: // destination selected, attempt migration
18: if (dst = V) then
19:    if (V will migrate before src completes) then
20:       return Unsuccessful migration
21:    else
22:       if (V does not have sufficient space) then
23:          return Unsuccessful migration
24:       else
25:          if (network has insufficient bandwidth) then
26:             return Unsuccessful migration
27:          else
28:             return Successful migration
29:          end if
30:       end if
31:    end if
32: else
33:    if (dst = I) then
34:       if (network has insufficient bandwidth) then
35:          return Unsuccessful migration
36:       else
37:          return Successful migration (to infrastructure)
38:       end if
39:    else
40:       return Unsuccessful migration
41:    end if
42: end if
43: End
```

3.2. VVMM-LW

The first modification to Algorithm 1 is Vehicular Virtual Machine Migration with Least Workload (VVMM-LW). This algorithm involves search criteria that consider both the workload of the potential destinations and their distance from the migrating vehicle, to make a more informed decision. The vehicle will search for vehicles that have enough resource space to host its VM load. It is worthwhile noting that the search is limited by the pool of the source vehicle. If there are multiple viable candidates, the source selects those with the least workload utilization. Should there be several potential candidates with equal resource space, it will select the one nearest in terms of distance. This algorithm does not consider time needed for migration, and is likely to suffer from an unsuccessful migration due to the selected destination not having enough time remaining in the grid to receive the migration. However, VVMM-LW is expected to outperform the VVMM-U algorithm in terms of VM drop rate as it uses a priori knowledge on the workload profiles of potential destinations.

3.3. VVMM-MA

In order to avoid unsuccessful migrations occur due to the destination not having enough time remaining, Vehicular Virtual Machine Migration with Mobility-Awareness (VVMM-MA) refines the search criteria further. The migrating vehicle takes the amount of time potential destinations have remaining in the grid. If a vehicle will not be in the grid long enough to receive the workload migration, it will not be considered as a viable candidate. Once a vehicle is forecasted to have enough time to receive the load, it is then checked for VVMM-LW criteria (resource space followed by proximity to the source). It is worthwhile noting that a viable candidate should have enough residual in-network time and memory/compute capacity to receive the workload migration. However, there are some cases when the migration will be marked as unsuccessful.

In order to decide the time of migration, a vehicle considers its current resource workload and the total duration of migration based on an expected upload speed. If a vehicle is selected as a destination for migration, the workload will gradually increase as data is moved from source to destination. As the onboard workload increases, the time needed to migrate will increase accordingly. This will force the vehicle to set its expected migration time to an earlier point. A situation may occur where the updated migration time needed is greater than the residual in-network time of the destination. In that case, the destination vehicle will recalculate, and in its own interest, need and begin to have its new load migrate prior to completion of the ongoing migration. This will result in an unsuccessful migration for the source. This can be described as a '*first hop*' failure. Another case could occur when the initial migration is successful and the destination now has more workload but its remaining in-network time is not enough to have its new workload migrate. This can be described as a '*second hop*' failure.

3.4. MDWLAM

To avoid the 'first hop' problem, the source vehicle must select a destination with remaining time in the grid, greater than or equal to the summation of the time necessary for the source to migrate its load, and for the destination to migrate its load. In order to avoid the 'second hop' problem, the migrating vehicle must further consider the time necessary for the destination vehicle to migrate both its current load, and the migrated load. As such, Mobility and Destination Workload Aware Migration (MDWLAM) utilizes Eq. (1), to calculate the cutoff time for the search criteria.

$$Cutoff = 2(T_s) + T_d \tag{1}$$

Only vehicles remaining in the grid longer than *Cutoff* are considered as viable candidates. The source vehicle consequently selects the vehicle with the longest time remaining among the viable candidates. The idea behind this selection is to maximize data retention in the grid. In case there is a tie between multiple candidates for migration, the selection among them will involve the remaining criteria proposed by the VVMM-LW, i.e., resource space followed by proximity to the source.

This algorithm involves the most well-informed decision in selecting a migration destination, and as such it outperforms the previous algorithms as we present in the next section.

4. Performance evaluation

In order to evaluate the feasibility and performance of each of the proposed algorithms, a set of simulations was conducted, on MATLAB, with several metrics to be observed.

Our previous study involved some preliminary simulations with several assumptions and simplifications [30]: A simplified mobility model was used restricting vehicles to select between a set of predetermined paths. VMs were of no tangible units and arbitrarily set to Light, Medium and Heavy. Vehicles were initialized to host only one type of VM. The system was observed for only a 5 minute window. Migration delays were all assumed to be fixed to 25 seconds, regardless of the VM size. The system was assumed to have infinite bandwidth, disregarding network congestion and capacity limitations of existing protocols. The MDWLAM algorithm was not tested. In this paper, these restrictions and simplifications are addressed and the simulation settings have been revised as presented in the next subsections.

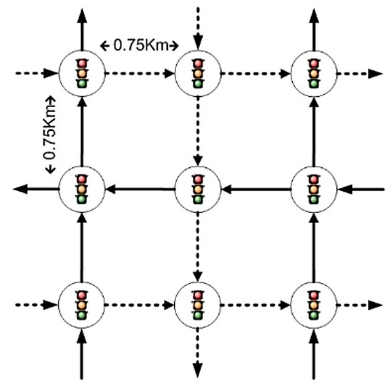

Fig. 1. The layout of the simulated grid.

4.1. Simulation settings

The simulation model (built in MATLAB) involves several main modules [31]. First, the vehicle mobility model based on a certain topology is set. The vehicles are then assigned VM workloads. Finally, the VVMM algorithm is executed.

4.1.1. Grid topology and mobility model

The topology is considered as a grid composed of 9 intersections, as shown in Fig. 1. The streets are assumed to be one-way, indicated by the direction of their arrows. Each segment (from one intersection to another) is 0.75 km long. The overall area is 1.5 km × 1.5 km, covered by 3 LTE-capable antennas. Thus, the total available upload bandwidth is 3 × 50 Mbps. It is worthwhile noting that the available bandwidth affects the network capacity and consequently the probability of unsuccessful migrations.

Each intersection is assumed to have a traffic light which can be red or green for $t \in \{30, 45, 60\}$ seconds. A vehicle enters the grid and begins to travel in a certain direction. As it moves along, its speed varies smoothly (but randomly) between 10 km/h and 50 km/h. As it reaches an intersection, it naturally decelerates, stops at red light, and resumes its course once it turns green. If there are two possible directions to take, it randomly selects one of the two paths. However, unlike in the previous study in [30], the vehicle is allowed to revisit a segment (between two intersections) up to 3 times. If a vehicle reaches a state when it has two options but both have been traversed 3 times, it selects the segment that is the nearest to an exit route for the grid.

The mobility model also assumes that a vehicle in a parking spot can resume moving in an hour or in 10 minutes by 5% or 20% probability, respectively.

To model a realistic and dynamic flow of traffic, new vehicles are generated based on a probabilistic process considering the number of vehicles in the grid at each point in time. The number of vehicles in the grid is modeled so as to oscillate between two values ($0.8x$ and x) where $x \in \{25, 50, 100\}$ for $congestion \in \{low, medium, high\}$, respectively. The maximum speed is decreased as congestion increases, such that for $congestion = \{low, medium, high\}$, $max\ speed = \{50, 40, 30\}$, respectively. Leaving time of a vehicle is not known in advance as it is an output of the continuous trajectory generation. Thus, number of vehicles in the network varies every second. As the number gets closer to the lower limit $0.8x$, the probability a new vehicle arrival increases. On the other hand, as the number of vehicles gets closer to the upper limit x, vehicle arrival rate decreases. Employing this methodology stabilizes the number of vehicles in the grid to reflect a certain congestion level.

4.1.2. VM workload distribution

VMs are split into three main sizes, light, medium and heavy, with size y, $2y$ and $4y$, respectively. In order to observe the impact of changing the size of a VM, simulations test cases where $y \in \{10, 25, 50\}$ MB. There are no restrictions on the initial workload a vehicle can host other than its maximum capacity. For the sake of simplicity, all vehicles are assumed to have the same capacity ($8y$). First the number of VM types is randomly selected (from 1 to 3). Then, based on the number of VM types, any of the possible combinations is randomly selected. Finally, the exact number of each possible VM is randomly selected. These selections also include initializing a vehicle to an idle load.

4.2. Simulation metrics and stages

The mobility model, topology, VM initialization and subsequent application of the various proposed algorithms have been simulated with MATLAB. The simulations are used to evaluate the performance of each of the proposed algorithms, based on several metrics that are collected and analyzed. Based on the performance of each algorithm, restrictions can be heuristically proposed and tested, to further enhance system performance.

4.2.1. Metrics

The various metrics used to measure system performance include:

The percentage of dropped migrations ($\boldsymbol{D}$) is calculated as shown in Equation (2).

$$D = \frac{\sum_{t=1}^{T} d_t}{\sum_{t=1}^{T} d_t + \sum_{t=1}^{T} s_t} \tag{2}$$

The percentage of successful migrations that were in fact V2I migrations ($\boldsymbol{V}$) is calculated as shown in Equation (3).

$$V = \frac{\sum_{t=1}^{T} v_t}{\sum_{t=1}^{T} s_t} \tag{3}$$

The average percentage utilization of the collective available resources ($\boldsymbol{R}$) is calculated as shown in Equation (4).

$$R = \frac{\sum_{t=1}^{T} l_t}{T \cdot r_m} \tag{4}$$

The average percentage utilization of the available upload bandwidth ($\boldsymbol{UL}$) is calculated as shown in Equation (5).

$$UL = \frac{\sum_{t=1}^{T} ul_t}{T \cdot ul_m} \tag{5}$$

The average percentage utilization of the available download bandwidth ($\boldsymbol{DL}$) is calculated as shown in Equation (6).

$$DL = \frac{\sum_{t=1}^{T} dl_t}{T \cdot dl_m} \tag{6}$$

All simulations are run for a constant $T = 3000$ s. It is also assumed that the large area simulated, is covered by 3 LTE antennas, offering an upload bandwidth of 6.25 MB/s (50 Mb/s) and download bandwidth of 12.5 MB/s (100 Mb/s) each, resulting in a maximum available upload bandwidth $ul_m = 18.75$ MB/s and download bandwidth $dl_m = 37.5$ MB/s. The uplink being the higher constraint coupled with the guarantee that it will always be higher than any downlink communication created by the cloud, the focus will be on the uplink. Finally, as the number of vehicles is set to oscillate between two values, $0.8x$ and x, if the computing resources of each vehicle is fully utilized, it would carry a load of size $8y$, resulting in the maximum VM load capacity that the grid can offer is bounded below by $l_m = (8y)(0.8x)$, and is bounded above by $l_m = (8y)(x)$ where x depends on the congestion level and y depends on the selected VM size for simulation. $\boldsymbol{R}$ is the calculated to be the average of the case using the lower and the upper limit of l_m. It is also assumed that a typical upload (and consequently download) speed is 1MB/s, for an individual vehicle attempting migration. These details are summarized in Table 1.

Table 1
Simulation settings.

Variable	Value
Grid size	1.5 km × 1.5 km
# of LTE antennas	3
T	3000 s
ul_m	18.75 MB/s
dl_m	37.5 MB/s
Avg UL/DL speed	1 MB/s
Low congestion	20 ∼ 25 vehicles
Medium congestion	40 ∼ 50 vehicles
High congestion	80 ∼ 100 vehicles

4.2.2. Simulation stages

While measuring the above metrics, the model was simulated in several stages, for various degrees of congestion as well as different VM sizes. Initially, the performance of the various proposed algorithms is analyzed without any imposed restrictions on the available resources allocated to VM loads. As calculated using Eq. (4), the average percentage utilization of the collective available resources ($\boldsymbol{R}$) can be regarded as an indicator of the point of saturation, where the grid is unable to accommodate further loads. In a subsequent set of simulations, this value is set as an imposed, administrative limit, to the maximum load that can be allotted to the available resources in the grid. Such a restriction is analogous to a conventional cloud administrator managing the utilization of the available resources. Naturally the administrator is aware of the available resource ‘space’ and will only lease out a load that the resources can handle, to maximize performance. In this case, it is projected that such an imposed restriction, coupled with the utilization of one of the various proposed algorithms, should maximize performance.

4.3. Simulation results

The various metrics and model settings introduced in the previous section, were used to simulate several scenarios. The simulations involved:

- Vehicular Congestion: Low, Medium, High
- VM Size: 10 MB, 25 MB, 50 MB
- Resource Restrictions: Unrestricted, Restricted

The breakdown of all the scenarios can be found in Table 2. Each individual scenario was run for 30 seeds and the metrics collected were averaged over all the seeds. The following sections present each metric to compare and analyze the performance of the proposed algorithms. For clarity in the figures, MDWLAM is shortened to MDW.

4.3.1. Percentage of dropped migrations ($\boldsymbol{D}$)

A VVMM algorithm is desired to have a low percentage of Dropped Migrations ($\boldsymbol{D}$). Aside from the other metrics, an algorithm should most importantly have a low $\boldsymbol{D}$. While, for example, a low $\boldsymbol{UL}$ is also preferred, this should not come at the expense of a higher $\boldsymbol{D}$. This is due to $\boldsymbol{D}$ directly implying a loss of cloud data,

Table 2
Simulated scenarios.

Congestion (# vehicles)	VM size (MB)	Restriction (%)
Low (20 ∼ 25)	10	–
Low (20 ∼ 25)	10	64
Low (20 ∼ 25)	25	–
Low (20 ∼ 25)	25	50
Low (20 ∼ 25)	50	–
Low (20 ∼ 25)	50	43
Medium (40 ∼ 50)	10	–
Medium (40 ∼ 50)	10	66
Medium (40 ∼ 50)	25	–
Medium (40 ∼ 50)	25	56
Medium (40 ∼ 50)	50	–
Medium (40 ∼ 50)	50	46
High (80 ∼ 100)	10	–
High (80 ∼ 100)	10	66
High (80 ∼ 100)	25	–
High (80 ∼ 100)	25	56
High (80 ∼ 100)	50	–
High (80 ∼ 100)	50	42

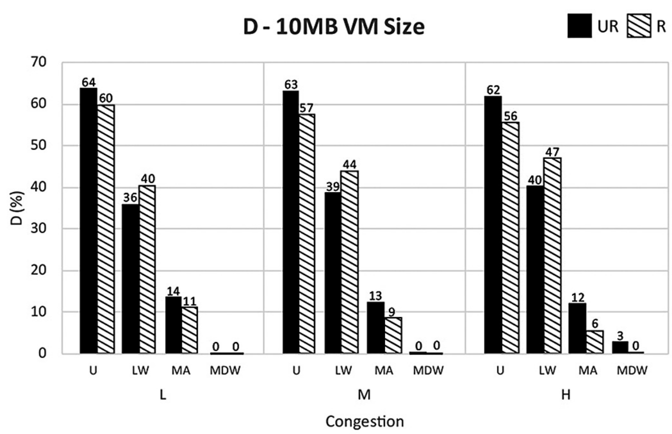

Fig. 2. Percentage of dropped migrations ($\boldsymbol{D}$) vs. Congestion – 10 MB VM Size.

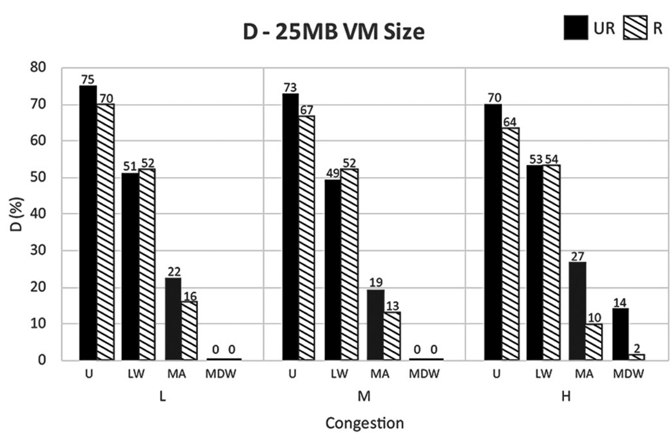

Fig. 3. Percentage of dropped migrations ($\boldsymbol{D}$) vs. Congestion – 25 MB VM size.

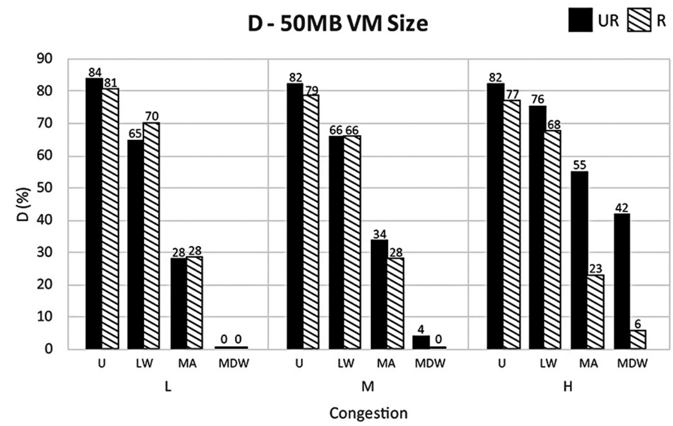

Fig. 4. Percentage of dropped migrations ($\boldsymbol{D}$) vs. Congestion – 50 MB VM size.

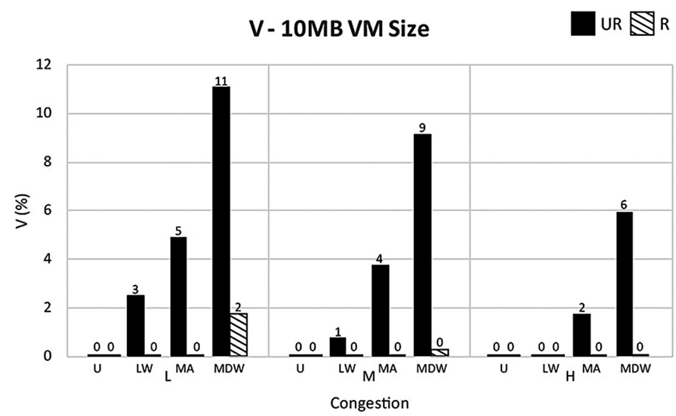

Fig. 5. Percentage of V2I migrations ($\boldsymbol{V}$) vs. Congestion – 10 MB VM size.

potentially violating the Service Level Agreement (SLA), while unfavorable values of the other metrics merely imply undesired grid conditions. In general, lower values of $\boldsymbol{D}$, $\boldsymbol{V}$, $\boldsymbol{UL}$, $\boldsymbol{DL}$, are preferred whereas higher values for $\boldsymbol{R}$ are preferred.

There is a general trend of a decrease in unsuccessful migrations, as the algorithms become more advanced. This can be observed in Figs. 2, 3 and 4, logically, showing that as the algorithm complexity increases, the $\boldsymbol{D}$ decreases. There is also a clear reduction as the restrictions to maximum data allowed are imposed. Fig. 2 shows the $\boldsymbol{D}$ for 4 algorithms, for 3 congestions, both Unrestricted (labeled 'UR' in the figures) and Restricted (labeled 'R' in the figures), at a VM Size of 10 MB. Figs. 3 and 4 show the same, for 25 MB and 50 MB respectively.

4.3.2. *Percentage of V2I migrations (*$\boldsymbol{V}$*)*

As shown in Figs. 5 to 7, the V2I migrations increase (for the unrestricted scenarios), as the algorithm becomes more advanced. This is because the source vehicles make a more informed decision, now realizing that rather than migrate to a vehicle that will not meet the search criteria, it will instead migrate to the infrastructure. The restrictions imposed drastically decrease the V2I migrations, also giving a more favorable performance, as the algorithm becomes more advanced. Fig. 5 shows the $\boldsymbol{V}$ for 4 algorithms, for 3 congestions, both Unrestricted (UR) and Restricted (R), at a VM Size of 10 MB. Figs. 6 and 7 show the same, for 25 MB and 50 MB respectively.

4.3.3. *Average percentage utilization of collective available resources (*$\boldsymbol{R}$*)*

The resource utilization $\boldsymbol{R}$ follows a different trend as opposed to the other metrics; it increases as the algorithm becomes more advanced. However, and yet as expected, the imposed restrictions cause the utilization to decrease. This is a tradeoff that is acceptable in order to minimize the two previous metrics, drop percentage and V2I percentage. Fig. 8 shows the $\boldsymbol{R}$ for 4 algorithms, for 3 congestions, both Unrestricted (UR) and Restricted (R), at a VM Size of 10 MB. Figs. 9 and 10 show the same, for 25 MB and 50 MB respectively.

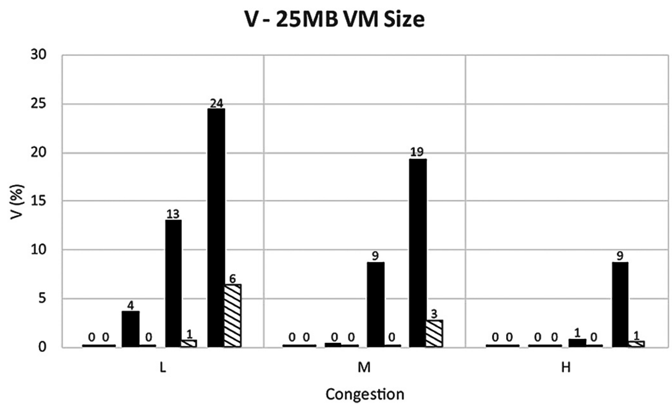

Fig. 6. Percentage of V2I migrations ($\boldsymbol{V}$) vs. Congestion – 25 MB VM size.

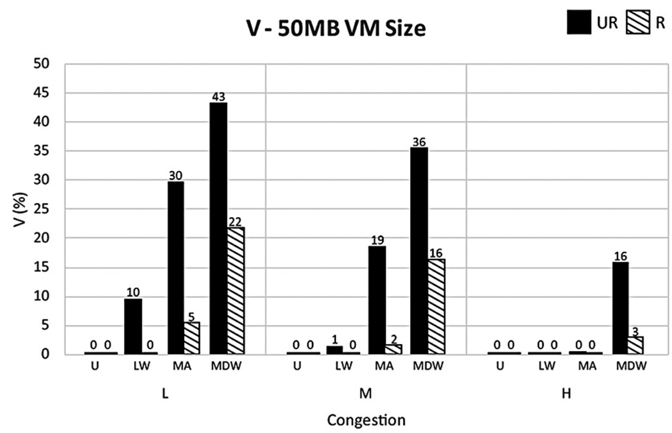

Fig. 7. Percentage of V2I migrations ($\boldsymbol{V}$) vs. Congestion – 50 MB VM size.

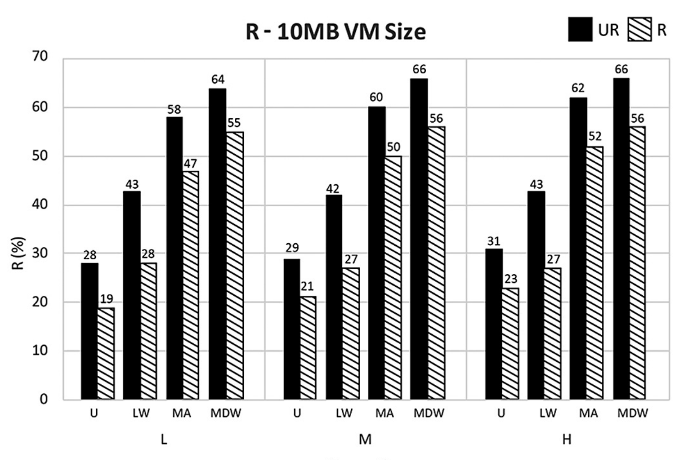

Fig. 8. Average percentage collective resource utilization ($\boldsymbol{R}$) vs. Congestion – 10 MB VM size.

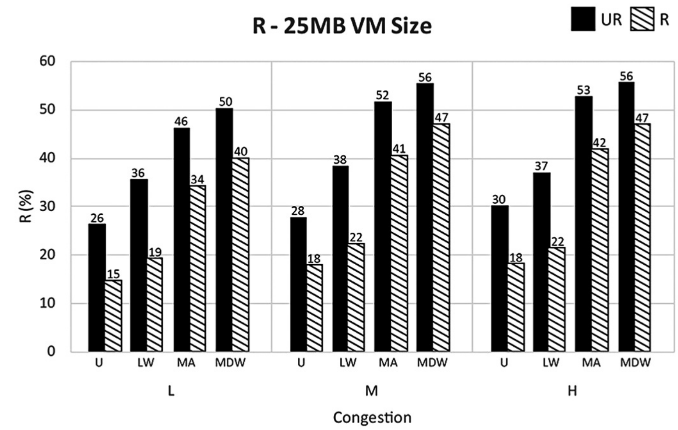

Fig. 9. Average percentage collective resource utilization ($\boldsymbol{R}$) vs. Congestion – 25 MB VM size.

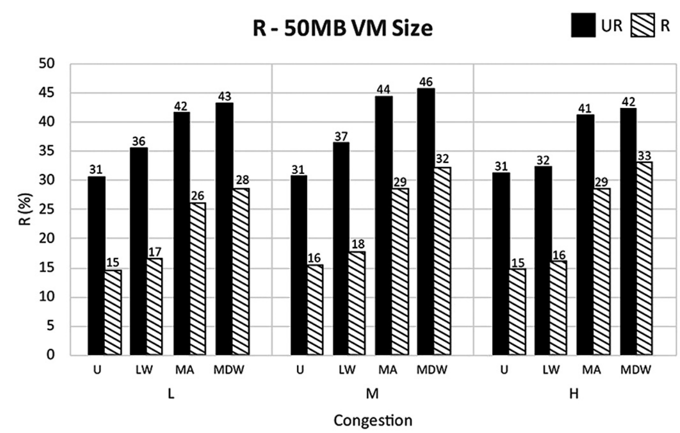

Fig. 10. Average percentage collective resource utilization ($\boldsymbol{R}$) vs. Congestion – 50 MB VM size.

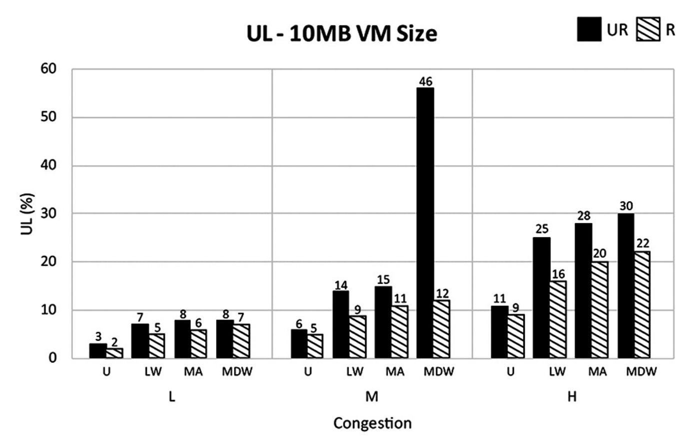

Fig. 11. Average percentage utilization available upload bandwidth ($\boldsymbol{UL}$) vs. Congestion – 10 MB VM size.

4.3.4. Average percentage utilization of available bandwidth

Fig. 11 shows the $\boldsymbol{UL}$ for 4 algorithms, for 3 congestions, both Unrestricted (UR) and Restricted (R), at a VM Size of 10 MB. Figs. 12 and 13 show the same, for 25 MB and 50 MB respectively, and Figs. 14, 15 and 16 show the same, but for $\boldsymbol{DL}$, for 10 MB, 25 MB and 50 MB respectively. The results show very favorable performance with regards to bandwidth utilization (both uplink and downlink). Typical usage of the LTE network (by normal, non-vehicle subscribers) involves mostly downlink communication. As can be observed from Fig. 16, the downlink utilization by the vehicular cloud, peaks at 31%. The uplink utilization peaks at around 65%, as shown in Fig. 13. This has a direct impact on the Quality of Service (QoS) that the LTE network offers. The higher the utilization by the VC, the less bandwidth available for the LTE users. This is both difficult to quantify and highly dependent on the excess bandwidth the network provider allocates to the grid.

In summary, overall, there are several aspects to consider. Generally, a higher congestion is more favorable. The more vehicles available in the grid, the more VMs can be hosted, the more potential candidates for migration are available, with the lowest drop and V2I percentages. It is also notable that lower VM sizes are

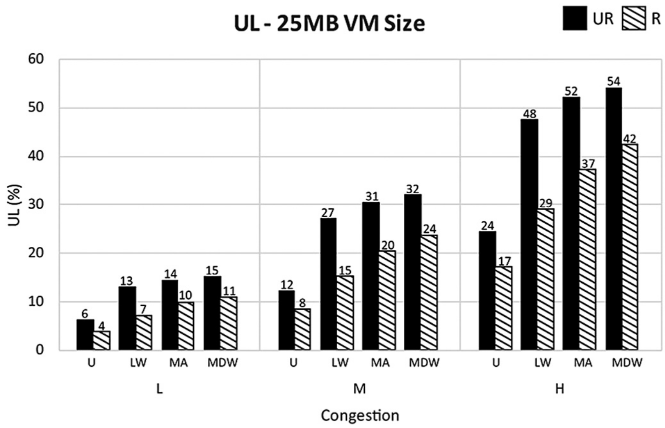

Fig. 12. Average percentage utilization available upload bandwidth ($\boldsymbol{UL}$) vs. Congestion – 25 MB VM size.

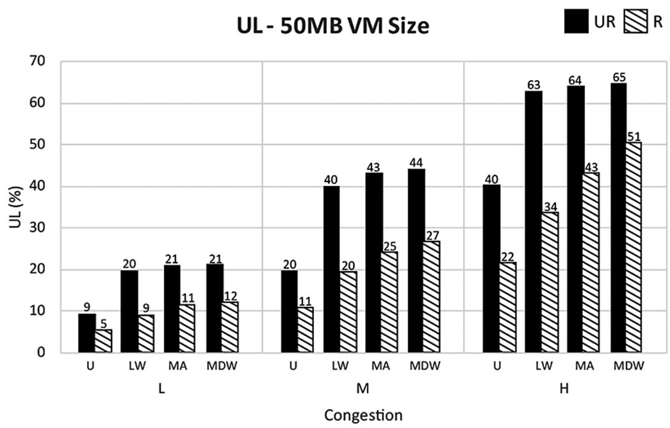

Fig. 13. Average percentage utilization available upload bandwidth ($\boldsymbol{UL}$) vs. Congestion – 50 MB VM size.

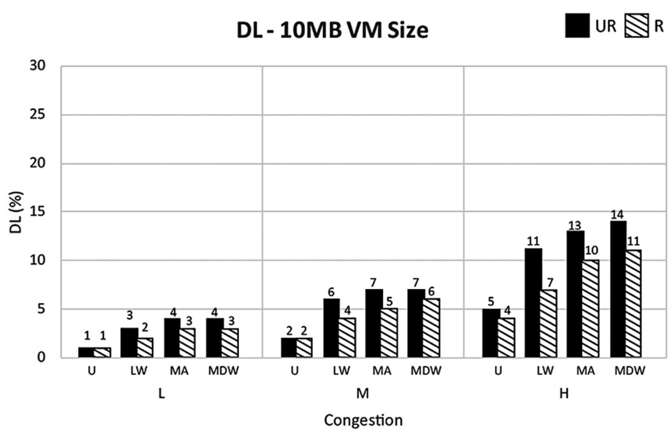

Fig. 14. Average percentage utilization available download bandwidth ($\boldsymbol{DL}$) vs. Congestion – 10 MB VM size.

preferable. While a higher VM size may offer a lower overall resource utilization percentage, this is translated to a greater amount of physical resources available to the cloud. Selecting the VM size is important to consider as it will most affect the network/bandwidth congestion.

4.4. Non-ideal communication

While the simulations presented earlier assume ideal conditions, it is important to consider the impact interference and con-

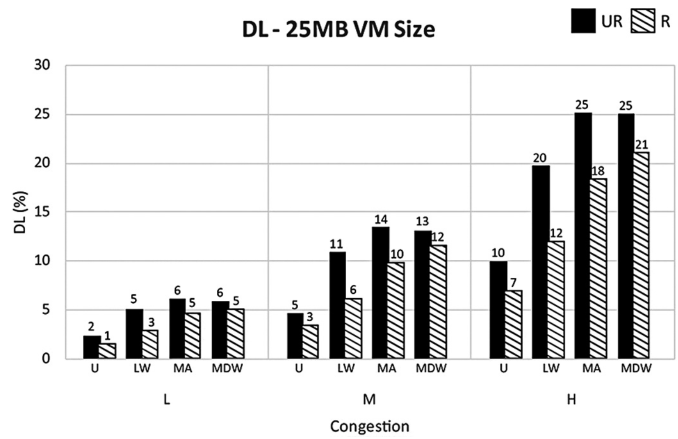

Fig. 15. Average percentage utilization available download bandwidth ($\boldsymbol{DL}$) vs. Congestion – 25 MB VM size.

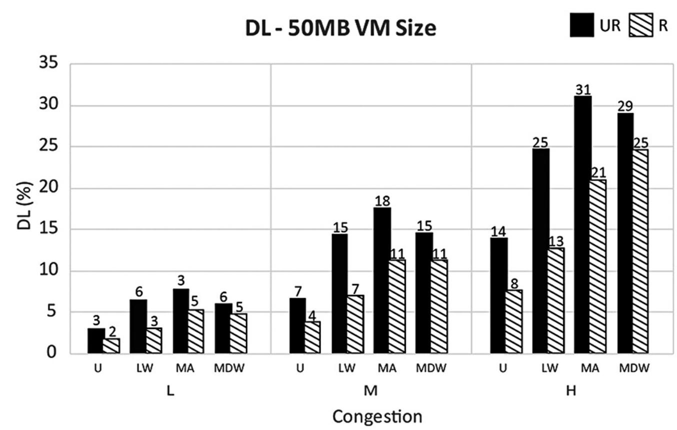

Fig. 16. Average percentage utilization available download bandwidth ($\boldsymbol{DL}$) vs. Congestion – 50 MB VM size.

nection instabilities may have on the performance of the VVMM algorithms. A simplified method of incorporating interference and connection imperfections is presented in what follows.

The probability of unsuccessful packet transmission at time t ($\boldsymbol{P_t}$) is calculated as shown in Equation (7).

$$P_t = P_{max} \cdot \frac{Int_t + CQ_t}{2} \tag{7}$$

where the interference due to concurrent migrations at time t ($\boldsymbol{Int_t}$) is calculated (and normalized) as shown in Equation (8).

$$Int_t = \frac{ul_t}{ul_m} \tag{8}$$

The connection quality between source and destination, based on the distance between them, at time t ($\boldsymbol{CQ_t}$) is calculated (and normalized) as shown in Equation (9).

$$CQ_t = \frac{SD_t}{SD_m} \tag{9}$$

A single scenario, for both unrestricted and restricted was selected to test the effect of non-ideal conditions: Light congestion and a VM Size of 25 MB. When the resource restriction is imposed, 50% is used, just as shown earlier in Table 2 for the corresponding scenario.

As can be deduced from Table 3, all algorithms are expected to experience increased drop rates, once non-ideal conditions are considered. While this may be disconcerting, it is actually quite logical and easily rectified. The algorithms were all designed to

Table 3
Simulation results for non-ideal conditions with a safety factor – light congestion and 25 MB VM size.

Algorithm	Restriction	Safety factor (s)	***D*** (%)	***V*** (%)	***UL*** (%)	***DL*** (%)	***R*** (%)
VVMM-U	–	100	***80.56***	0.00	7.90	2.05	27.22
VVMM-LW	–	100	***60.21***	0.00	18.96	5.29	35.70
VVMM-MA	–	100	***44.35***	0.04	18.48	7.08	43.61
VVMM-MDWLAM	–	100	***36.69***	0.08	16.97	7.15	45.44
VVMM-U	50%	100	***73.11***	0.00	5.16	1.43	15.23
VVMM-LW	50%	100	***55.75***	0.00	11.94	3.00	20.02
VVMM-MA	50%	100	***28.99***	0.00	10.76	4.49	30.09
VVMM-MDWLAM	50%	100	***16.85***	0.00	11.24	5.30	34.51

take into account ideal conditions, and as such, the time needed for a vehicle to migrate its load, has no room for packet loss. If even one packet is lost, it guarantees failure of the migration. To avoid this, vehicles must utilize a ***time safety factor***. The simulations were repeated to prove the validity of this solution, and the results are shown in Table 3. The results exhibit similar performance trends as previously shown in the ideal conditions. Future work will study the effect of varying the safety factor in an adaptive manner, as well as refining the modeling for the non-ideal conditions.

5. Conclusion

The vehicular cloud has many potential applications as well as different approaches with regards to implementation. One such paradigm utilizes the vehicles as hosts for Virtual Machines (VM), effectively constructing a data center, where the resources are mobile. While the conventional cloud problems, such as hotspots or load balancing, are the main triggers for VM migration, vehicular clouds have different triggers for VM migration. Due to the inherent dynamicity of the vehicular environment, vehicles hosting VM loads, may exit the coverage of a cloud area, and as such must migrate their workload to another vehicle that is still accessible to the cloud. We have proposed several migration algorithms for such scenarios, and evaluated their performance via simulations. Simulations show that an unrestricted allotment of VM loads can result in a significant percentage of unsuccessful migrations, regardless of the performance improvement offered by the various algorithms proposed.

To counteract the issue of unsuccessful migrations, based on the performance of the unrestricted system, certain metrics are used to empirically predict a restriction to be imposed on the system to improve performance. These restrictions are simulated and shown to drastically decrease the percentage of unsuccessful migrations.

In our future work, we are planning to consider the Quality of Service (QoS) requirements as vehicular VM migration calls for novel solutions to address differentiated service quality requirements. Furthermore, although ensuring minimum VM migration latency and minimum service disruption was not the primary focus of this paper, the problem is emergent in vehicular clouds. As our main focus is on the communication aspects of VM migration in vehicular clouds, we have not tackled security and privacy issues however our future studies also include addressing virtualization-based vulnerabilities in vehicular VM management and migration.

References

[1] K. Hwang, G. Fox, J.J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann, Waltham, MA, 2011.
[2] J. Wan, C. Zou, K. Zhou, R. Lu, D. Li, IoT sensing framework with inter-cloud computing capability in vehicular networking, Electron. Commer. Res. 14 (13) (November 2014) 389–416.
[3] N. Cordeschi, D. Amendola, E. Baccarelli, Hard and soft optimal resource allocation for primary and secondary users in infrastructure vehicular networks, in: Proceedings of the 12th IEEE Consumer Communications and Networking Conference (CCNC), January 2015, pp. 708–713.
[4] R. Hussain, Z. Rezaeifar, H. Oh, A paradigm shift from vehicular ad hoc networks to VANET-based clouds, in: Wireless Personal Communications, Springer, 2015, pp. 1–28.
[5] P. Ghazizadeh, S. Olariu, A.G. Zadeh, S. El-Tawab, Towards fault-tolerant job assignment in vehicular cloud, in: Proceedings of the IEEE International Conference on Services Computing (SCC), June 2015, pp. 17–24.
[6] S. Bitam, A. Mellouk, S. Zeadally, VANET-cloud: a generic cloud computing model for vehicular ad hoc networks, IEEE Wirel. Commun. 22 (1) (February 2015) 96–102.
[7] M. Gerla, E.-K. Lee, G. Pau, U. Lee, Internet of vehicles: from intelligent grid to autonomous cars and vehicular clouds, in: Proceedings of the IEEE World Forum on Internet of Things, Seoul, Korea, March 2014.
[8] M. Whaiduzzamana, M. Sookhaka, A. Gania, R. Buyyab, A survey on vehicular cloud computing, J. Netw. Comput. Appl. (2013).
[9] M. Gerla, Vehicular cloud computing, in: Proceedings of 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Ayia, Napa, Cyprus, June 2012.
[10] R. Hussain, J. Son, H. Eun, S. Kim, H. Oh, Rethinking vehicular communications: merging VANET with cloud computing, in: Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science, Taipei, Taiwan, 2012.
[11] S. Abdelhamid, H. Hassanein, G. Takahara, Vehicle as a resource (VaaR), IEEE Netw. 29 (1) (January 2015) 12–17.
[12] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, I. Khalil, Datacenter at the airport: reasoning about time-dependent parking lot occupancy, IEEE Trans. Parallel Distrib. Syst. 23 (11) (2012) 2067–2080.
[13] H. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: architecture, applications and approaches, Wirel. Commun. Mob. Comput. 13 (18) (2013) 1587–1611.
[14] K. Ezirim, S. Jain, Taxi-cab cloud architecture to offload data traffic from cellular networks, in: Proceedings of IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), June 2015, pp. 1–6.
[15] H. Mousannif, I. Khalil, H. al Moatassime, Cooperation as a service in VANETs, J. Univers. Comput. Sci. 17 (8) (2011) 1202–1218.
[16] R. Hussain, F. Abbas, J. Son, H. Oh, TIaaS: secure cloud-assisted traffic information dissemination in vehicular ad hoc networks, in: Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Delft, Netherlands, 2013.
[17] M. Mishra, A. Das, P. Kulkarni, A. Sahoo, Dynamic resource management using virtual machine migrations, IEEE Commun. Mag. 50 (9) (September 2012) 34, 40.
[18] M. Eltoweissy, S. Olariu, M. Younis, Towards autonomous vehicular clouds, in: Ad Hoc Networks, Springer, 2010, pp. 1–16.
[19] E. Lee, E.-K. Lee, M. Gerla, Vehicular cloud networking: architecture and design principles, IEEE Commun. Mag. 52 (2) (February 2014) 148–155.
[20] N. Bila, E.J. Wright, E. De Lara, K. Joshi, H.A. Lagar-Cavilla, E. Park, A. Goel, M. Hiltunen, M. Satyanarayanan, Energy-oriented partial desktop virtual machine migration, ACM Trans. Comput. Syst. 33 (1) (March 2015).
[21] N. Vignesh, R. Shankar, S. Sathyamoorthy, V.M.A. Rajam, Value added services on stationary vehicular cloud, in: Distributed Computing and Internet Technology, vol. 8337, 2014, pp. 92–97.
[22] K. Gu, D. Zeng, S. Guo, B. Ye, Leverage parking cars in a two-tier data center, in: Proceedings of the IEEE Wireless Communications and Networking Conference, Shanghai, China, April 2013, pp. 4665–4670.
[23] S. Olariu, I. Khalil, M. Abuelela, Taking VANET to the clouds, Int. J. Pervasive Comput. Commun. 7 (1) (2011) 7–21.
[24] S. Olariu, T. Hristov, G. Yan, The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds, Mobile Ad Hoc Networking, John Wiley & Sons, Inc., Hoboken, NJ, 2013.
[25] R. Yu, Y. Zhang, S. Gjessing, W. Xia, K. Yang, Toward cloud-based vehicular networks with efficient resource management, IEEE Netw. 27 (5) (2013) 48–55.
[26] R. Boutaba, Q. Zhang, M.F. Zhani, Virtual machine migration in cloud computing environments: benefits, challenges, and approaches, in: H. Mouftah,

B. Kantarci (Eds.), Communication Infrastructures for Cloud Computing, IGI-Global, USA, September 2013, pp. 383–408.
[27] T. Wood, P. Shenoy, A. Venkataramani, M. Yousif, Black-box and gray-box strategies for virtual machine migration, in: Proceedings of the USENIX Conference on Networked Systems Design and Implementation, Cambridge, MA, April 2007, pp. 17–32.
[28] R.W. Ahmad, A. Gani, S.H. Hamid, M. Shiraz, A. Yousafzai, F. Xia, A survey on virtual machine migration and server consolidation frameworks for cloud data centers, J. Netw. Comput. Appl. 52 (June 2015) 11–25.
[29] N. Kumar, R. Iqbal, S. Misra, J.J. Rodrigues, Bayesian coalition game for contention-aware reliable data forwarding in vehicular mobile cloud, Future Gener. Comput. Syst. 48 (July 2015) 60–72.
[30] T.K. Refaat, B. Kantarci, H.T. Mouftah, Dynamic virtual machine migration in a vehicular cloud, in: Proceedings of the IEEE Symposium on Computers and Communication (ISCC), Madeira, Portugal, June 2014.
[31] 3GPP LTE Standard, 3GPP TS 36.321 V10.6.0 (2012-09), Medium Access Control (MAC) Protocol Specification (Release 10).