@ARTICLE{8416774,

author={B. Brik and N. Lagraa and N. Tamani and A. Lakas and Y. Ghamri-Doudane},
journal={IEEE Transactions on Vehicular Technology},

Renting out Cloud Services in Mobile Vehicular Cloud

title={Renting Out Cloud Services in Mobile Vehicular Cloud},
year={2018},
volume={67},
number={10},
pages={9882-9895},
doi={10.1109/TVT.2018.2858002},

Bouziane Brik, LIM University of Laghouat, L3i University of La Rochelle, b.brik@mail.lagh-univ.dz, bouziane.brik@gmail.com
Nasreddine Lagraa, LIM University of Laghouat, nasrlag@gmail.com
Nouredine Tamani, L3i University of La Rochelle, nouredine.tamani@univ-lr.fr, ntamani@gmail.com
Abderrahmane Lakas, College of Information Technology, United Arab Emirates University, alakas@uaeu.ac.ae
Yacine Ghamri-Doudane, L3i University of la Rochelle, yacine.ghamri@univ-lr.fr

ISSN={0018-9545},
month={Oct},}

***Abstract*—Vehicular Cloud (VC) is a promising paradigm in which mobile vehicles can be both cloud users and service providers. It enables vehicles that have sufficient resources to act as mobile cloud servers by offering a wide range of services to user's vehicles. However, user vehicles need to discover provider vehicles along with their services, and request targeted services from them, especially with the high mobility of vehicles. Moreover, provider vehicles and their services are characterized by specific quality criteria and consumer vehicles need to select the most suitable services. In this paper, we design a new protocol which enables vehicles to discover and consume services of mobile cloud servers that are moving nearby. The protocol harnesses public buses, which provide a stability in terms of time (regular timetable) and space (fixed line in an urban area), making them act as cloud directories, where provider vehicle can register their services. Public buses spread service registration information to the buses within the same bus line, and to buses of connected lines, as well. A consumer vehicle can discover the available services, along with their constraints, by querying the buses in their vicinity. To ensure service consumption, the buses provide also a routing protocol allowing communications between the provider and the consumer. Furthermore, we extend our protocol to enable, on the one hand, vehicle providers to select the most appropriate public bus for an efficient service registration, and users to select, on the other hand, the most satisfactory service, which satisfies both provider constraints and user preferences. Simulation results showed the efficiency of our protocol in terms of service discovery and service consuming delays, as well as its ability to select both the most suitable cloud directory and service provider for both providers and consumers, respectively.**

***Index Terms*—Mobile Vehicular Cloud, Consumer Vehicles, Provider Vehicles, Cloud Directories, Public Buses, Service Selection.**

I. Introduction

CLOUD Computing (CC) is an Internet-based technology in which a pool of scalable computing functions and services are provided on-demand to cloud consumers, as a public utility [1]. Consumers obtain and pay for services such as shared hardware, software, databases, information, to name a few, without considering infrastructure and maintenance costs. As an extension, mobile cloud computing was introduced to extend service offering to mobile users [2].

Besides, advances in automotive area have allowed vehicles to be more intelligent by providing a comfortable, smoother and a safer driving experience [3], [4]. These advances rely on the in-vehicle capabilities, such as wireless communication, data storage and processing units, and on-board sensors. Statistics show that most vehicles spend many hours per day in a parking garage, parking lot, or driving to work [5]. Meantime, their on-board resources are still usually under-utilized. Recently, the idea of making use of vehicle resources in a cloud environment has given birth to vehicular cloud concept [6], [7].

Vehicular Cloud (VC) is an emerging paradigm which represents the merging of mobile cloud computing and Vehicular Ad hoc Network (VANET). It aims at harnessing the under-utilized on-board vehicular resources. Moreover, vehicles resources can be rented out to mobile user vehicles, and hence offer a wide range of services, such as Network as a Service (NaaS) to provide Internet access, STorage and Computing as a Service (STaaS, CaaS) for vehicles that may need additional storage or computing resources to run their applications, COoperation as a Service (COaaS) to perform specific tasks (e.g. participatory data collection), INformation as a Service (INaaS), and ENtertainment as a Service (ENaaS), which encompasses warnings about congestions and accidents, weather and/or road conditions, parking availability, video or music files [7], [8], etc.

Although both vehicular and conventional clouds are very similar, when they are compared from offered resources point of view, the main feature that distinguishes between them is the high mobility of vehicles, and consequently the volatility of service provider availability. In [7], [9], a taxonomy of VC has been presented, which defines two main VC scenarios: (i) static vehicular cloudlet, which is formed by unexploited resources of parked vehicles in airports, company parking spaces, parking lots, or traffic jam; (ii) Dynamic vehicular cloud in which providers can rent out their cloud services while they are on the move. In the latter case, Consumer Vehicles (CVs) need to discover Provider Vehicles (PVs) along with their services before requesting any services from them. To do so, a cloud directory is usually used to store information about PVs, and hence to form a dynamic index of such providers. Moreover, these cloud directories must also be able to localize PVs to enable CVs requesting services directly from PVs. In their turn, PVs should select the most suitable cloud directory where they will publish efficiently their services. In other words, storing PVs' services on the most adequate cloud directory means that PVs have a higher likelihood to offer their services to consumers. On the other side, CVs can end up with several possible providers offering a given requested service. In such a case, they need to have a capacity to select the most satisfactory service provider, according to both specific service features or constraints and user preferences. Therefore, in this paper, we address the following issues:

- How do PVs choose the most adequate cloud directory and how does the latter locate service providers accordingly?
- How do CVs select the most suitable provider vehicle?
- How are requested services consumed from provider vehicles?

In order to answer the above listed questions, we introduce a new protocol enabling vehicles with sufficient resources to act as mobile servers in urban environment. It exploits public buses as cloud directories in which providers store their services. We choose to use public buses for several reasons: first, their movements are predictable in time and space in urban environments. Second, they constitute a mobile backbone by covering almost all parts of urban areas, and hence they improve the network connectivity. Third, they do not need either to obtain an authorization from an authority to form a VC, or to use privacy mechanisms, they are trustworthy by nature. Therefore, PVs can register and share their offered services with public buses in their vicinity, and CVs can discover and consume requested services from PVs either directly or via wireless multi-hop communication, through public buses located also in their vicinity. Furthermore, we extend our protocol to enable, on the one hand, PVs to find the appropriate cloud directory to advertise their services, and on the other hand, CVs to select the most satisfactory service satisfying both their preferences and service constraints, by using a fuzzy quantified approach.

This paper extends our previously published contributions in [10]–[12], which can be listed as follows:

- a new public bus-based protocol to discover and consume vehicular cloud services; we called our protocol Discovering and Consuming Cloud Services in Vehicular Clouds (DCCS-VC) [10].
- a new provider localization technique, which is also based on public buses [10].
- a new cloud directory selection scheme [11], which is based on Simple Additive Weighting (SAW) technique [13].
- we extend in this paper our protocol to provide an efficient ranking scheme of service providers, which is based on linguistic quantifiers and linguistic quantified propositions to aggregate both service constraints and user preferences [12].
- we showed the efficiency of our schemes, in terms of minimizing service discovery and consuming delays, by performing extensive simulations comprising around $1,500$ vehicles in an urban city map of $9 \times 9\,km^2$.

The remainder of the paper is organized as follows. Section II is about related work in the field of vehicular cloud. Section III describes our DCCS-VC and its main steps, such that Subsection III-A introduces our cloud directory selection scheme, Subsection III-B is dedicated to our service provider localization, Subsection III-C details our provider selection scheme, and Subsection III-D describes the service consumption mechanism between provider and consumer vehicles. Section IV summarizes the simulations we have carried out, and discusses the results obtained about the performance of our protocol and both cloud directory and service provider selection schemes. Section V concludes the paper and draws some lines for future work.

II. Related Work

In the last decade, vehicular cloud has been studied in several research efforts, which can be classified into three main classes: (i) Architectures and Frameworks to design cloud systems and vehicular cloud [9], [14]–[18]; (ii) Protocols to discover and consume cloud services [19]–[24], which enable both provider vehicles to offer their services and consumer vehicles to discover and consume providers' services; (iii) Service provisioning to manage resource allocation and service selection [25]–[27].

A. Architectures and Frameworks

A new architecture is proposed in [9], which defines three vehicular cloud scenarios: Vehicular Clouds (VCs) in which a vehicle offers its services to another vehicle, Vehicles using the traditional Cloud computing (VuCs) through Road Side Units (RSUs), and Hybrid Clouds (HCs), which combine VCs and VuCs. Hence and as mentioned before, a vehicle in VCs can provide its services when it is either parked and/or stuck in a traffic jam (Static cloud), or on the move (dynamic cloud).

To improve the road safety, a generic model for VANET-Cloud has been developed in [14], which allows VANET end users (drivers and passengers) to discover and consume cloud services either from conventional cloud computing, such as Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS), or from mobile vehicles. This model is composed mainly of three layers: (i) Client layer, which is the lowest level in this model and consists in mobile end users (ii) Communication layer, to ensure the connections between the clients and both conventional cloud computing and VANET-cloud, and (iii) Cloud layer, which refers to the both traditional cloud and VANET resources.

A Vehicular Cloud for Road side scenarios (VCR) architecture has also been introduced in [15]. VCR aims at providing services related to safety and non-safety applications. It defines two communication modes: public vehicular cloud (or Vehicle to Infrastructure, V2I, communication, which allows vehicles to access to the conventional cloud computing through RSUs), and private vehicular cloud (or Vehicle to Vehicle, V2V, in which a vehicle can consume cloud services of another vehicle in an Ad hoc way).

To deal with driver's safety and comfort, the authors in [16] developed another architecture called vehicular cloud or V-Cloud. It consists of three layers: Vehicular Cyber-Physical System (VCPS), V2V, and V2I. The VCPS layer comprises two types of sensors, namely vehicle internal sensors, which are responsible of providing vehicle context awareness (such as vehicle location), and smart-phone sensors, which are used inside the car to monitor driver's health and mood conditions.

A cloud assisted system for autonomous driving, called Carcel system, has been introduced in [17]. Carcel collects information from autonomous vehicles and static roadside sensors to assist cars in planning: (i) safer paths by avoiding obstacles such as pedestrians and other vehicles, and (ii) efficient paths by detecting unexpected events such as accidents or traffic jams.

To facilitate the deployment and delivery of mobile cloud applications over vehicular networks, a Service Centric Contextualized Vehicular cloud platform (SCCV) has been developed in [18]. SCCV is a multi-tier architecture, which consists of network, mobile devices that are embedded in cars, and cloud tiers. The network tier provides any wireless connectivity upon VANET and Internet, such as WiFi, DSRC, cellular technologies, while the cloud tier is in charge of collecting traffic data and Internet services (e.g., weather and geographic information) from multiple sources, in order to interpret them and deliver personalized mobile services to different vehicular users.

B. Protocols to discover and consume cloud services

An RSUs-based protocol has been defined in [19], called disCoveRing and cOnsuming services WithiN vehicular clouds or CROWN. The idea behind this protocol is to use RSUs as Cloud directories in which provider vehicles, called STAR, store their services. These RSUs enable consumer vehicles to discover STAR cloud services, and consume them through a routing protocol, also introduced by the same authors [28].

In [29], a new protocol called Renting out and Consuming Services in Vehicular Clouds (RCS-VC), has been designed. It enables provider vehicles to rent out their services and resources to other vehicles, by extending the LTE-A [30] infrastructure to exploit it as cloud directories with which provider vehicles register their offered services, and from which consumer vehicles discover their requested services.

In [20], the author made use of public buses as mobile gateways in vehicular clouds to connect user vehicles to the traditional cloud computing via the VANET infrastructure, but they are not used to form VC as in our work.

In [21], in another context, vehicles serve as witnesses using their mounted cameras, to provide a new VANET-cloud service, called Vehicle Witnesses as a Service (VWaaS). This protocol comprises two main entities: vehicles moving on the road and the infrastructure of the conventional cloud. The vehicles, after discovering an event such as terrorist attack, deadly accident, traffic jam, take pictures of the area of interest and send them to the cloud. The latter processes the collected pictures and analyses them to generate warning or precautionary measures. In addition, the cloud may also provide forensics details to public authorities, judiciary, law enforcement agencies, or insurance agencies.

To provide efficient Internet access, the authors in [23] introduced a cloud-supported gateway model, called Gateway as a Service (GaaS). The proposed protocol consists of three components: a gateway to connect vehicles to the Internet, which can be stationary (RSU) or mobile (vehicle), client vehicle, which requests access to the Internet, and a cloud server, which provides two sub-servers: GaaS register to maintain all information about gateways, and GaaS dispatcher to dispatch related gateways for the client vehicles.

A cluster-based protocol has been introduced in [22] to form mobile vehicular clouds, called Fuzzy clustering-based Vehicular cloud Architecture (FcVcA). In this work, two main contributions have been developed. The former is a fuzzy logic-based election algorithm to select a vehicle as a Cluster Head (CH). The latter, when a vehicle needs to consume a cloud service, it has to request the nearest CH, which selects the most adequate provider vehicle using the Q-learning technique [33]. The provider vehicle selection has been performed according to several criteria, such as average speed, bandwidth, storage and computing resources.

Similarly, to offer data as a service, another cluster-based scheme has been developed to form mobile vehicular cloud [24], called Distributed d-Hop Clustering algorithm for VANET (DHCV), where mobile vehicles are grouped into several clusters; each cluster is considered as a vehicular cloud. After cloud construction, an optimization scheduling algorithm is applied to improve both throughput and delay in transferring data from provider vehicles to their cluster heads.

C. Service provisioning

In [25], [26], [31], the authors proposed several scheduling models to manage vehicles resource allocation in an airport parking lot. In these works, vehicles, which arrive and leave at different periods of time, can offer their computing resources to be used by other vehicles. To prevent service interruption when a provider vehicle leaves the parking during a service execution, a checkpointing strategy has been developed, where the state of the computation is saved periodically, and the most recent stored image is used to recover the computation. A migration process that enables the resource allocation to be transferred from one provider to another has also been introduced.

Considering parked vehicles in a parking lot, authors in [27], [32] proposed a negotiation-based system between provider and consumer vehicles, to manage on-demand service provisioning. Consumer vehicles discover and select cloud services through a trusted third party, which is also in charge to control both providers misbehavior and payment operation.

D. Discussion

Table I compares the above mentioned studies according to three criteria: class type (architecture, protocol, and service provisioning), the targeted cloud type, which aims to connect vehicles either to conventional Cloud Computing (CC) or to Vehicular Cloud (VC), and the offered services by each proposal. We notice that the above mentioned work focus on vehicle cloud architectures, which are designed to facilitate the deployment of this new concept. Each architecture comprises a set of modules (layers); each of which plays a specific role. Furthermore, existing protocols enable provider vehicles to rent out their services, and consumer vehicles to discover

TABLE I
COMPARATIVE STUDY BETWEEN VEHICULAR CLOUD WORK.

	Work class			**Targeted Cloud**		**VC Scenario**		**Services**
	Architecture	**Protocol**	**Service provisioning**	**Cloud Computing**	**Vehicular Cloud**	**Static Cloudlet**	**Dynamic Cloudlet**	
(VC, VuC HC) [9]	X			X	X		X	VC and CC services
VANET CLOUD [14]	X			X	X		X	VC and CC services
VCR [15]	X			X	X		X	VC and CC services
VCPS [16]	X			X			X	Driver safety and comfort
Carcel [17]	X			X			X	Safer, efficient paths
SCCV [18]	X			X			X	Traffic data and Internet services
CROWN [19]		X			X		X	VC services
Buses as gateway [20]		X		X			X	CC services
FcVcA [22]		X	X		X		X	VC services
DHCV [24]		X			X		X	Data as a service
RCS-VC [29]		X			X		X	VC services
VWaaS [21]		X		X	X		X	Events on the road
GaaS [23]		X			X		X	Internet access
Scheduling models [25], [26], [31]			X		X	X		Computing resources
Negotiation system [27], [32]			X		X	X		VC services

and consume cloud services. These protocols can be divided into two major categories. The first group provides service consumption in mobile cloud environment, such as CROWN, DHCV, RCS-VC and FcVcA protocols, whereas the second group offers to vehicles the ability to consume services from the traditional cloud computing, as in [20]. However, no work offers to the providers the ability to select the appropriate cloud directory. Moreover, these protocols enabling to form a vehicular cloud architecture only when a vehicle needs to consume a service. Consequently, a user vehicle may not find a service provider offering the service requested with a given QoS. We also note that few work have been proposed for service provisioning in vehicular cloud and all work focus on static vehicular cloudlet (stationary vehicles).

In our work, we introduce a new proactive service discovering and consuming protocol[1]. It enables, on the one hand, providers to select the most adequate cloud directory and to share their offers, and on the other hand, user vehicles to discover and choose among many offers the one which is the most satisfactory regarding the QoS and user preferences. Therefore, user vehicles have a higher likelihood to select the best service provider, while they are on the move.

III. DCCS-VC PROTOCOL DESCRIPTION

The Discovering and Consuming Cloud Services in Vehicular Clouds protocol (DCCS-VC) aims at enabling capable vehicles to rent out their various resources directly to their neighboring vehicles, in a mobile vehicular cloud. In fact, DCCS-VC comprises three main actors: (i) Provider Vehicles (PV), which need first to discover cloud directories in their vicinity, and to select the most adequate one in order to register their services; (ii) Public Buses (PB) to exploit them as cloud directories, in which PVs store their services and from which consumers discover offered services; and (iii) Consumer Vehicles (CV), which need, in their turn, to discover PV services before performing any request. In addition, CVs can choose the most satisfactory service, in the case when they discover several possible providers for the requested service.

[1]Proactive protocol means that the providers start to store their offers at cloud directories, and form a vehicular cloud architecture even in the absence of service requesters.

Figure Fig. 1 illustrates the main steps of DCCS-VC protocol. In what follows, we describe how DCCS-VC deals with the aforementioned issues. Subsection III-A introduces our cloud directory selection scheme. Service provider localization is detailed in Subsection III-B. Service provider selection scheme is introduced in Subsection III-C. Subsection III-D describes the service consumption mechanism between provider and consumer vehicles.

A. *Provider Vehicles: How to select a public bus?*

When a PV decides to participate in the VC, it first defines its offered services and their Quality of Services (QoS). In [10], [29], we grouped the quality criteria of VC services into two main classes, namely: (i) common requirements of services such as service price and availability, and (ii) specific requirements, according to the service type, such as storage capacity for STaaS, bandwidth for NaaS, etc., as summarized in Table II. Besides, determining which bus should be selected for registration is an important issue, if a PV discovers many public buses in its vicinity (Step 1 in Fig. 1). Our bus selection process is based on the Simple Additive Weighting (SAW) technique [13], according to quality criteria. For instance, Table III identifies four quality criteria, which can be extended to any other criteria.

Let $C = \{c_1, \ldots, c_m\}$ be a set of $m \in \mathbb{N}^*$ bus criteria, and $B = \{b_1, \ldots, b_n\}$ a set of $n \in \mathbb{N}^*$ bus candidates discovered by a provider vehicle PV. We build a performance matrix $Q_B = B \times C = \left(Q_{b_i c_j}; 1 \leq i \leq n; 1 \leq j \leq m\right)$ by gathering for each bus candidate b_i, its value $Q_{b_i c_j}$ for each criterion c_j.

A weighted mean-based aggregation method is then applied on matrix Q_B to compute the score of each bus. This score is

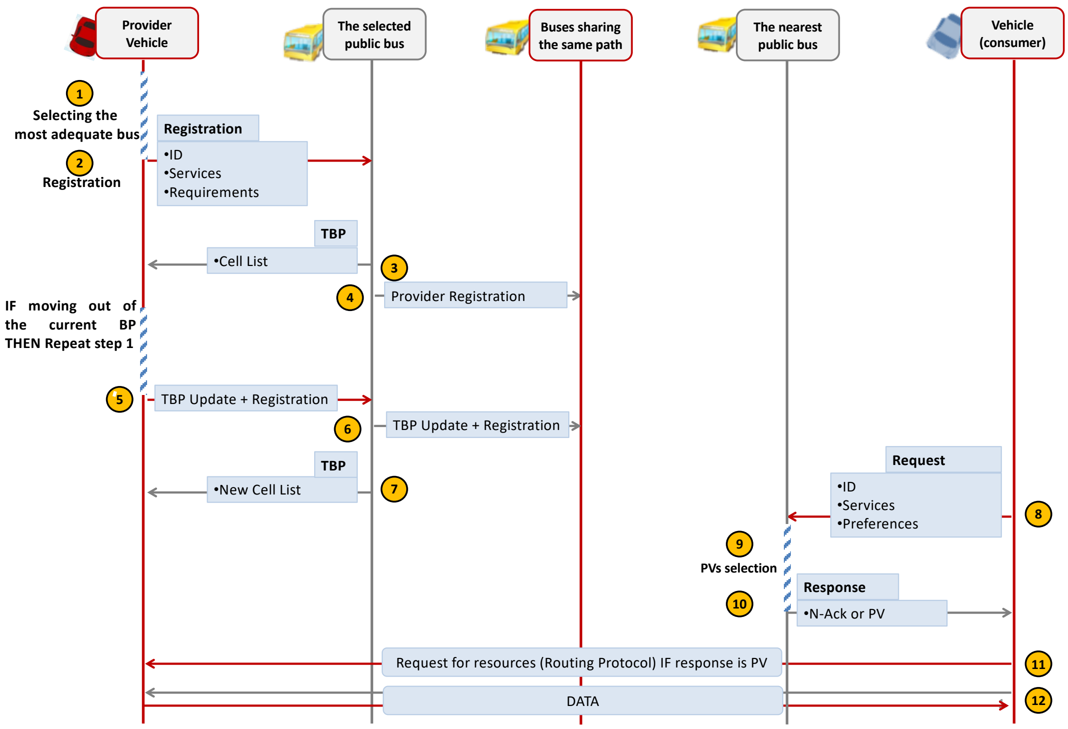


Fig. 1. Main operations performed in DCCS-VC as a sequence diagram.

TABLE II
CRITERIA DEFINED FOR VEHICLE CLOUD SERVICES.

Criteria	NaaS	STaaS	COaaS	INaaS & ENaaS
Specific	Bandwidth	Max capacity	Max Storage	Max Information Capacity
	Delay	Max Storage Time	Processing Capability	
			Max delay	
Common	Cost (price) per Hour	Cost per storage unit	Cost per hour	Cost per information unit
	Availability			

TABLE III
QUALITY CRITERIA OF PUBLIC BUSES.

Criterion	Definition	Measure Unit
N_{PV}	Number of registered PVs at this bus	-
$A_{storage}$	Available storage space in this bus	Gb
S_{rate}	Rating of PVs which had offered correctly their services through this bus	-
B_{Cost}	The cost per storage unit	(\$/Mb)

used to select the most satisfactory bus, according to The bus selection process consists in the two following main steps:

1) **Normalization step**: to normalize the m quality criteria. Quality criteria are grouped into two main classes: (i) negative criteria where the higher the value is, the lower the quality is, such as the storage cost; their values are normalized by using Formula (1), and (ii) positive criteria where the higher the value is, the higher the quality is, they are normalized according to Formula (2).

$$V_{b_i c_j} = \begin{cases} \frac{Q_{c_j}^{max} - Q_{b_i c_j}}{Q_{c_j}^{max} - Q_{c_j}^{min}} & \left(Q_{c_j}^{max} - Q_{c_j}^{min} \neq 0\right) \\ 1 & \left(Q_{c_j}^{max} - Q_{c_j}^{min} = 0\right) \end{cases} \quad (1)$$

$$V_{b_i c_j} = \begin{cases} \frac{Q_{c_i b_j} - Q_{c_j}^{min}}{Q_{c_j}^{max} - Q_{c_j}^{min}} & \left(Q_{c_j}^{max} - Q_{c_j}^{min} \neq 0\right) \\ 1 & \left(Q_{c_j}^{max} - Q_{c_j}^{min} = 0\right) \end{cases} \quad (2)$$

where $b_i, i \in \{1, \dots, n\}$ is a bus, $c_j, j \in \{1, \dots, m\}$ is a criterion, $Q_{c_j}^{max}$ is the maximal value of quality criterion c_j in matrix Q_B, i.e., $Q_{c_j}^{max} = max_{i=1}^{n}\left(Q_{b_i c_j}\right)$, $Q_{c_j}^{min}$ is the minimal value of quality criterion c_j in matrix Q_B, i.e., $Q_{c_j}^{min} = min_{i=1}^{n}\left(Q_{b_i c_j}\right)$, and $Q_{b_i c_j}$ is the quality criterion value of bus b_i for criterion c_j.
By applying equations (1) and (2), we obtain another matrix $V_B = \left(V_{c_i b_j} : 1 \leq i \leq n, 1 \leq j \leq m\right)$, where a row corresponds to a bus candidate, and a column corresponds to its quality criterion normalized values.

2) **Weighting step**: as bus criteria do not have the same

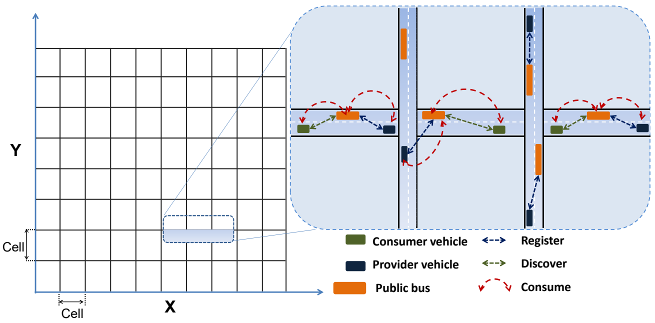


Fig. 2. Map city partitioning.

importance from the service provider point of view, then we attach to each criterion c_j a weight w_j, such that $w_j \in [0,1]$ and $\sum_{j=1}^{m} w_j = 1$, to take into account the vehicle provider preferences. Therefore, to compute the overall quality score for each bus candidate, we make use of the following formula:

$$Score(b_i) = \sum_{j=1}^{m} w_j . V_{b_i c_j} \quad (3)$$

For a given bus candidate set, the PV will choose a bus which has the highest score, expressing a compromise between PV's preferences. If there are several buses with maximal score, one of them is selected randomly.

B. Public Buses: how to localize providers?

Once a PV registers its services with a public bus (Step 2 in Fig. 1), this latter informs all buses in its group about this registration (Step 4 in Fig. 1). As mentioned before, the public bus must also be able to localize PVs in order to enable CVs to consume requested services directly from PVs. To do so, we proposed new grid-based PVs localization technique, named Grid-based Tracking Cell technique (GTC). GTC is also based on public buses and partitioning the city map into several cells (cf. Fig. 2). Then, the cells are grouped into Tracking Bus Path (TBP) where every TBP corresponds to a predetermined path of a group of buses. Therefore, when a PV registers a service at a public bus, this latter answers the PV by its corresponding TBP (Step 3 in Fig. 1). In other words, the PV will be informed by all cells covered by this bus. If this PV moves out this TBP, it must register again with another selected bus and receives a new TBP (Steps 5-7 in Fig. 1).

C. Consumer Vehicles: how to select a provider vehicle?

When a vehicle requests a service such as Internet access, resources for storage or computation, or information, etc. it sends a request packet to the nearest public bus (Step 8 in Fig. 1). In addition to the targeted service, the request contains the consumer preferences expressed over the service, such as service price and execution duration.

Moreover, the asked bus may discover several providers, which can offer the targeted services. In such a case, it has to choose the most satisfactory provider, according to both consumer preferences and service constraints (QoS) (Step 9 in Fig. 1). To do so, we are based on a fuzzy approach which combines linguistic quantifiers and linguistic quantified propositions to aggregate both offered quality of service and consumer preferences, enabling an efficient ranking of provider vehicles, in terms of delay. Before we proceed further, we first introduce the basic concepts related to linguistic quantifiers and linguistic quantified propositions.

1) Fuzzy Set Theory and Linguistic Quantifiers:

- **Fuzzy set theory** [34] expresses the gradual membership of an element to a set. A fuzzy set F is defined on a univers U by a membership function $\mu_F : U \mapsto [0,1]$ such that $\mu_F(x)$ denotes the membership grade of x in F. In particular, $\mu_F(x) = 1$ denotes the full membership, $\mu_F(x) = 0$ expresses the absolute non-membership and when $0 < \mu_F(x) < 1$, it reflects a partial membership (the closer to 1 $\mu_F(x)$, the more x belongs to F). If a fuzzy set is a discrete set then it is denoted $F = \{(x_1, \mu_F(x_1)), ..., (x_n, \mu_F(x_n))\}$, otherwise, it is defined by its membership function, often a trapezoidal function. The union $\cup$ and the intersection $\cap$ operators are defined by a couple of a t-norm and a t-conorm, such as (min, max). Let F, G be two fuzzy sets, $\mu_{F\cup G}(x) = max(\mu_F(x), \mu_G(x))$, $\mu_{F\cap G}(x) = min(\mu_F(x), \mu_G(x))$, and the complement of F, denoted F^c, is $\mu_{F^c}(x) = 1 - \mu_F(x)$. Logical counterparts of $\cap$, $\cup$ and the complement are respectively $\wedge, \vee$ and $\neg$.
- **Linguistic quantifiers** express fuzzy quantities such as most of, around 4, few, etc. corresponding to a flexible attitude between the existential ($\exists$) and the universal ($\forall$) quantifiers. A linguistic quantifier Q can be absolute or relative. Absolute quantifiers such as "at least 3", "at most 5" and "around 4" express a number and their interpretations do not depend on the cardinality of the considered set. They are defined from $\mathbb{R}$ or $\mathbb{N}$ to $[0,1]$. Relative quantifiers such as "few", "almost all" and "around half" express a proportion and their interpretations depend on the cardinality of the considered set. They are defined from $[0,1]$ to $[0,1]$.
  The truth value of a linguistic quantifier Q is denoted by μ_Q.

2) Fuzzy quantified propositions: Linguistic quantifiers allow the definition of fuzzy quantified propositions by combining fuzzy predicates and quantifications. Let Q, X, A and B be respectively a linguistic quantifier, a set of elements and two fuzzy predicates. A fuzzy quantified proposition can have one of the following forms "Q X are A" or "Q B X are A". The former means that among elements of X, there are Q elements that satisfy the fuzzy predicate A, and the latter means that among elements of X that satisfying B, there are Q elements that satisfy A. Within a fuzzy quantified proposition, Q can be an increasing quantifier if the truth-value of the quantified proposition does not decrease if the satisfaction to fuzzy predicate A by elements of X increases. "at least 3" and "most of" are examples of increasing quantifiers. A quantifier Q can also be a decreasing quantifier if the truth-value of the quantified proposition does not increase if the satisfaction to fuzzy predicate A by elements of X increases. "at most 3" and “few” are examples of decreasing quantifiers. A quantifier is monotonic if it is either increasing or decreasing. Besides, non-monotonic or unimodal quantifiers refer to quantities such as “around 5”, “around a quarter”, etc.

Many approaches have been proposed to evaluate statements of the form “Q X are A”, but any evaluation should verify at least monotony properties (see [35] for more details).

The following interpretations of fuzzy quantified propositions are suitable for user preferences evaluation [35]:

- Decomposition based approach [36],
- OWA-based approach [37],
- Fuzzy integral (Choquet/Sugeno) based approach [35]. However, Choquet (resp. Sugeno) fuzzy integral-based approach is equivalent to the decomposition-based approach (resp. OWA-based approach) [35].

Hereinafter, we focus on the decomposition-based approach because of its polynomial complexity which fits the time constraint of delay in mobile vehicular clouds.

3) Quantification-based Ranking Approach: We recall in this subsection our ranking approach developed in [12], where we have combined a fuzzy expression of user preferences and a fuzzy aggregation method to aggregate both user preferences expressed over a service, and service constraints expressed by a vehicle provider over the advertised service.

The decomposition-based approach [36] evaluates fuzzy quantified propositions of the forms “Q X are A”. Let “$P : Q$ X are A” be a fuzzy quantified proposition such that Q is an increasing linguistic quantifier. The decomposition-based approach computes its truth-value, denoted by τ_P, based on the best crisp subset $E \subseteq X$ which contains Q elements that satisfy A, and defines Formula (4) if Q is an absolute quantifier and Formula (5) if Q is a relative one:

$$\tau_P = max_{i \in \{1,...,n\}} min(\mu_A(x_i), \mu_Q(i)) \quad (4)$$

$$\tau_P = max_{i \in \{1,...,n\}} min(\mu_A(x_i), \mu_Q(\frac{i}{n})) \quad (5)$$

where $\mu_A(x_1) \geq \mu_A(x_2) \geq ... \geq \mu_A(x_n)$.

The ranking rule is *“the closer to* 1 τ_P *is, the more satisfactory the vehicle provider is”*. The underlying preference relation is defined as follows:

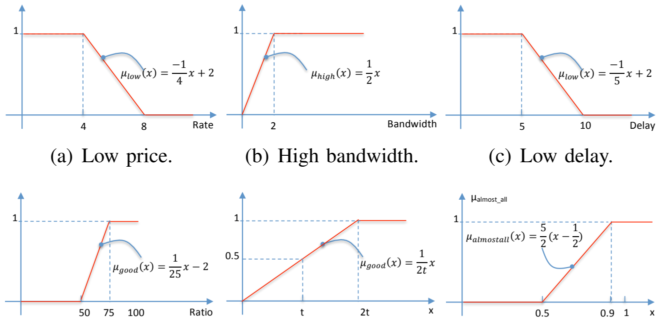
Definition 1 (preference relation $\tilde{\succeq}$): Let V_j and $V_{j'}$ be two alternatives. Then, $V_j \tilde{\succeq} V_{j'} \Leftrightarrow \tau_P(V_j) \geq \tau_P(V_{j'})$.

Remark 1: It is easy to prove that preference relation $\tilde{\succeq}$ is a total order.

Remark 2: The decomposition-based approach considers only increasing quantifiers. When Q is decreasing, the evaluation of “Q X are A” is equivalent to proposition “Q' X are not A”, where Q' is the antonym of Q.

4) Ranking of vehicle service providers: To rank service provider vehicles, we consider criteria related to both providers (PV) and their offered services (see Table II for instance). For a PV, we are based on the two following quality criteria:

1) **Successful Execution Ratio** (q_{ser}): it is the ratio $\frac{N(s)}{K}$ of number of times $N(s)$ that a PV’s service s has been successfully consumed to the total number K of invocations for service s.
2) **Execution duration** (q_{ed}): defined as the expected duration to deliver the requested service. It includes processing time $T_{Process}$ and transmission time T_{Trans}; thus $q_{ed} = T_{Process}(s) + T_{Trans}(s)$. We note that T_{Trans} is determined according to the required quantity of service s by a CV and the PV’s data throughput.



(a) Low price. (b) High bandwidth. (c) Low delay.

(d) Good ratio. (e) High availability. (f) Almost_all quantifier.

Fig. 3. Examples of membership functions.

5) Application of the Ranking Approach: Let us model a vehicle provider PV_i as a set of k couples (c_{ij}, o_{ij}) where c_{ij} with $j \in \{1, ..., k\}$ is a criterion related to the service provider constraint (of PV_i) as detailed in Table II, and o_{ij} is its value. For consumer vehicle $CV_{i'}$, we have a set of l couples $(c_{i'j'}, p_{i'j'})$ such that $c_{i'j'}$ with $j' \in \{1, ..., l\}$ is a quality criterion, and $p_{i'j'}$ is a user preference expressed over criterion $c_{i'j'}$. Then, we define PV_i service satisfaction degree regarding $CV_{i'}$, denoted by $\tau(V_{i|i'})$, as the truth value of the following fuzzy quantified proposition: *“Almost all preferences expressed by $CV_{i'}$ are satisfied by the vehicle provider PV_i”*. By using the decomposition-based approach, we obtain the following formula (6):

$$\tau(V_{i|i'}) = max_{r=1,...,k}(min(\mu_p(c'_{ir}), \mu_{almost\_all}(\frac{r}{l}))) \quad (6)$$

where $\mu_p(c'_{ir})$ is the satisfaction degree of constraint c'_{ir} to preference p, such that $\mu_p(c'_{i1}) \geq \mu_p(c'_{i2}) \geq ... \geq \mu_p(c'_{ir})$ are the descending raking of the criteria according to their evaluation $\mu_p(c_{ir})$.

Example 1: Let us suppose a consumer vehicle (CV) asking for a 10 mn Internet access. It discovers four vehicle providers PV_1-PV_4 offering NaaS service with criteria $Price$, $Bandwidth$, $Delay$ and $Ratio$. The values advertised by each provider for each criterion are listed in Table IV. Besides, figures Fig. 3(a)-Fig. 3(d) illustrate membership functions translating into graphics the user preferences expressed by CV over the criteria. For instance, Fig. 3(a) illustrates the preference “Low price” expressed on the criterion $Price(cost)$ by the following membership function defined from $\mathbb{R}_+$ to $[0, 1]$, such that $\mu_{Low}(x) = 1$ if $x \in [0, 4]$, $\mu_{Low}(x) = \frac{-x}{4} + 2$ if $x \in [4, 8]$, and $\mu_{Low}(x) = 0$ if $x \geq 8$. Let us define the membership function of the linguistic quantifier $almost\_all$ as illustrated in Fig. 3(f). It represents the following function defined from $[0, 1]$ to $[0, 1]$, such that $\mu_{almost\_all}(x) = 0$ if $x \in [0, 0.5]$, $\mu_{almost\_all}(x) = \frac{5}{2}(x - \frac{1}{2})$ if $x \in [0.5, 0.9]$, and $\mu_{almost\_all}(x) = 1$ if $x \in [0.9, 1]$.

Table V summarizes the evaluation of each service provider criterion over the corresponding user (consumer) preference. By using Formula (6), we have aggregated for each

TABLE IV
EXAMPLES OF VEHICLE PROVIDERS WITH THEIR CORRESPONDING CRITERION VALUES

Vehicle Provider	Price ($)	Bandwidth (Mo/s)	Delay q_{ed} (s)	Ratio q_{ser} (%)	Availability (mn)
PV_1	6.4	0.8	8.5	80	13
PV_2	10	1.5	6	55	5
PV_3	4.5	0.75	8	25	4
PV_4	6.8	1.6	7	60	7

TABLE V
VEHICLE PROVIDERS WITH CRITERION VALUES MAPPED TO USER PREFERENCE DEFINITIONS

Vehicle Provider	Low Price	High Bandwidth	Low Delay q_{ed}	Good Ratio q_{ser}	High Availability
PV_1	0.4	0.4	0.3	1	0.65
PV_2	0	0.75	0.8	0.2	0.25
PV_3	0.875	0.375	0.4	0	0.2
PV_4	0.3	0.8	0.6	0.4	0.35

provider alternative the performance obtained for each criterion. The evaluation of each alternatives gives the following results: $\tau_P(PV_1) = 0.4, \tau_P(PV_2) = 0.25, \tau_P(PV_3) = 0.375, \tau_P(PV_4) = 0.4$.

The ranking delivers PV_1 and PV_4 as the best alternatives. In this case, the user can select randomly among these best alternatives the most satisfactory vehicle provider. Finally, the requested bus sends the selected PV (Step 10 in Fig. 1) to the consumer vehicle. In the case where the requested bus does not find any available PV, then it sends a negative acknowledgment to the consumer vehicle.

D. Consumer Vehicle Vs Provider Vehicle: how to consume requested services?

When a CV receives a positive response from a bus, it sends its request directly to the selected PV (Step 11 in Fig. 1). In addition to the requested service, the packet also contains all required information before starting service consumption (Step 12 in Fig. 1), such as method of payment, order of data, technique used to insure both user privacy and data confidentiality, etc.

To perform the two last steps of DCCS-VC (Steps 11 and 12 in Fig. 1), we adopt a bus-based routing protocol defined in [38], which routes packets through only public buses. In our case, we use the public buses as forwarder nodes to route both request packets from CVs to PVs and Data packets from PVs to CVs. Moreover, we note that we allow the re-broadcasting of packets only among public buses sharing the same TBP, in order to avoid the broadcast storm problem.

IV. PERFORMANCE EVALUATION

In this section, we present the performance evaluation of our DCCS-VC protocol, cloud directory (bus) selection (subsection IV-A), and service provider selection approach (subsection IV-B). We used SUMO mobility simulator [39] to generate the vehicle movement files in urban mobility model, which are then used as input into the OMNet++ network simulator [40], to simulate network communication.

A. Simulation Setup and Parameters

We implemented DCCS-VC in a Manhattan-based map of $9\times9\ km^2$ where there are 180 roads of $1\ km$ and 16 junctions.

TABLE VI
SIMULATION PARAMETERS.

Parameter	**Value**
Simulation Framework	Veins (Omnet++, SUMO)
Mobility Model	Manhattan
Path-loss Model	Obstacle pathloss [41]
Simulation Time	$1000\,s$
Simulation Runs	3
Simulation Area	$9 \times 9\ km^2$
Transmission Range	500 m
Transfer rate	18 Mb/s
Vehicles and buses speed	Up to 70 km/h
Vehicle Density (VD)	$[300 - 1500]$ vehicles
Public Buses Density per TBP (PBD)	20, 40, and 60 buses
PVs density (P)	$1/4$, $1/3$, and $1/2$ of VD
The size of Registration, TBP, Request, and response Packets	128 Bytes
Data Packet Size	$[1 - 5]$ KBytes
PV's Queue Size	5 CVs
Maximum number of offered services per PV	3
Maximum number of requested services per CV	1

We covered this map by five Tracking Bus Paths (TBP) of 20 km, each of which is covered by n buses. We considered three values for bus density: 20, 40, and 60 buses in each TBP. In addition, we varied the Vehicle Density (VD) between 300 and $1,500$, where three values of PVs density are considered: one-fourth, one-third, and one-half of each level of VD, given that the default value of PVs density is one-third of VD. The main parameters of our simulation are listed in Table **??**.

B. Simulation Scenarios and Metrics

During the first $300s$ of our simulation, each PV has to register its offered services along with their requirements, with the selected public bus. After that, CVs start to send their requests for services (with their preferences). At receiving a request packet from a CV, if the chosen PV is idle, it starts serving the CV's request. Otherwise, it replies with a negative acknowledgement. We choose the following metrics to evaluate the DCCS-VC protocol:

1) Service **D**iscovery **D**elay (**DD**): measures the time elapsed from sending a request for service to receiving a response packet from a bus.

2) Service **C**onsuming **D**elay (**CD**): measures the time elapsed from sending a request for service to the PV by a CV to receiving all data packets; we limit the number of data packets in our case to 3.
3) Data packet **E**nd-**to**-**E**nd **D**elay (**E2ED**): measures the average time a data packet takes to reach the CV from the PV through public buses.
4) Service Selection Delay: measures time period taken by a CV to select a PV.
5) Successful PV's registration rate: the rate of PVs which have been registered successfully with the most adequate bus according to their QoS criteria (N_{PV}, $A_{storage}$, S_{rate}, and B_{Cost}, for instance).
6) Successful Execution Ratio: it measures the rate of CVs' requests are being responded correctly by selected PVs within the maximum expected time.

C. Protocol Comparison

To evaluate the performance of DCCS-VC, we compared it with 4 other schemes:

1) Broadcast DCCS-VC: we replaced the adopted bus-based routing scheme [38] by broadcasting operations. During step 11 of DCCS-VC (cf. Fig. 1), each CV broadcasts its request packet to its neighbors with a TTL (Time To Live) equal to 40. Then, each neighbor decreases the TTL by 1, and rebroadcasts the packet to its neighbors, until it reaches the PV (or $TTL = 0$). With a maximum distance between two neighbors equal to 500 meters, the maximum distance between a PV and a CV is 20 km, which represents the TBP length. We note that we use the same scheme to route data packets from PV to the CV.
2) CROWN protocol [19] in terms of DD, CD, and E2ED.
3) B-CROWN protocol [19] in which the various operations of CROWN were replaced by broadcasting operations. To be fair in our comparison, we set the same simulation parameters as used in [19].
4) Bus Line-based Effective Routing (BLER) [42]: it is a bus-based routing protocol in urban environment which routes data packets from a source node to a destination through existing public buses. We compare with BLER protocol to evaluate the performance of our routing protocol [38] in terms of both CD and E2E delays.

Besides, to evaluate our service selection scheme, we adopt the following schemes, proposed for static vehicular cloudlet:

1) Neutral scheme [32]: it selects the first encountered provider vehicle whatever its QoS.
2) Negotiation-based service selection [27]: it manages the service selection between provider and consumer vehicles through trusted vehicles. At receiving a service request, the trusted vehicles are responsible for the service negotiation by searching the best price.

Each simulation scenario is repeated three times and the final results are the average of the three runs.

D. Performance Evaluation of DCCS-VC Protocol

In Figures Fig. 4, Fig. 5, Fig. 6, and Fig. 7, we display the performance of DCCS-VC in terms of Discovering Delay (DD), Consuming Delay (CD), End To End Delay (E2ED), and number of consumed services compared to the number of offered services, respectively. As shown in Fig. 4(a), the DD of DCCS-VC has a stable performance as both VD and P increase. To discover a cloud service in DCCS-VC, a CV must ask directly the nearest public bus for that. Hence, the service discovering operation does not depend on either the VD or the number of PVs (P), but on the presence of the public buses in the vicinity. In fact, the performance of DCCS-VC relies mainly on the public bus density, where buses serve as forwarders between the PV and the CV. Fig. 4(b) depicts the performance of DCCS-VC while varying the Public Bus Density (PBD) in each TBP. It plots that DCCS-VC has almost the same DD for a PBD per TBP between 20 and 60, and its DD value is equal to 0.0062. As explained before, the DD is based on one-hop wireless communication, which is established between the CVs and the public buses. Consequently, DD does not depend on the public bus density, but on their presence in the vicinity. Fig. 4(c) illustrates the performance comparison between DCCS-VC, broadcast DCCS-VC, CROWN and B-CROWN, given that we set P and Public Buses Density per TBP (PBD) to one-third of VD and 40 buses, respectively. We infer that both DCCS-VC and Broadcast DCCS-VC have the same DD, which is lower than that of CROWN and B-CROWN. This can be justified by the fact that CROWN and B-CROWN use a routing scheme while DCCS-VC and broadcast DCCS-VC are based only on one-hop communication to discover a cloud service (as explained for Fig. 4(a) and Fig. 4(b)). We also remark that the DD of CROWN decreases as VD increases, unlike B-CROWN where the DD increases as the VD increases. The high density of vehicles allows packets to be more forwarded and less stored-and-carried, which is the routing scheme used in [19]. However, the behavior of B-CROWN is essentially due to the huge number of broadcasting packets that appears when we increase the VD.

Figures Fig. 5(a) and Fig. 5(b) and Fig. 6(a) and Fig. 6(b) show, respectively, that CD and E2ED have a stable performance as VD increases. However, they decrease as P increases. The presence of more PVs implies that a CV has a likelihood of being answered by these PVs, causing both CD and E2ED to decrease. Moreover, it is worth noticing that both CD and E2ED depend mainly on the PBD, which serve as forwarders between the CV and PV. Consequently, they decrease when we increase the PBD per TBP (cf. Fig. 5(b) and Fig. 6(b)). Nevertheless, we also remark that CD and E2ED values for 40 PBD per TBP are lower than those for 60 PBD per TBP. Using a high number of public buses as forwarders will increase the number of hops whereby data packets pass to reach their destinations, causing CD and E2ED to increase.

Moreover, DCCS-VC provides the lowest CD and E2ED when compared to the other schemes (cf. Fig. 5(c) and Fig. 6(c)). This is due mainly to two following reasons:

1) The grid-based TBP architecture which limits the ser-

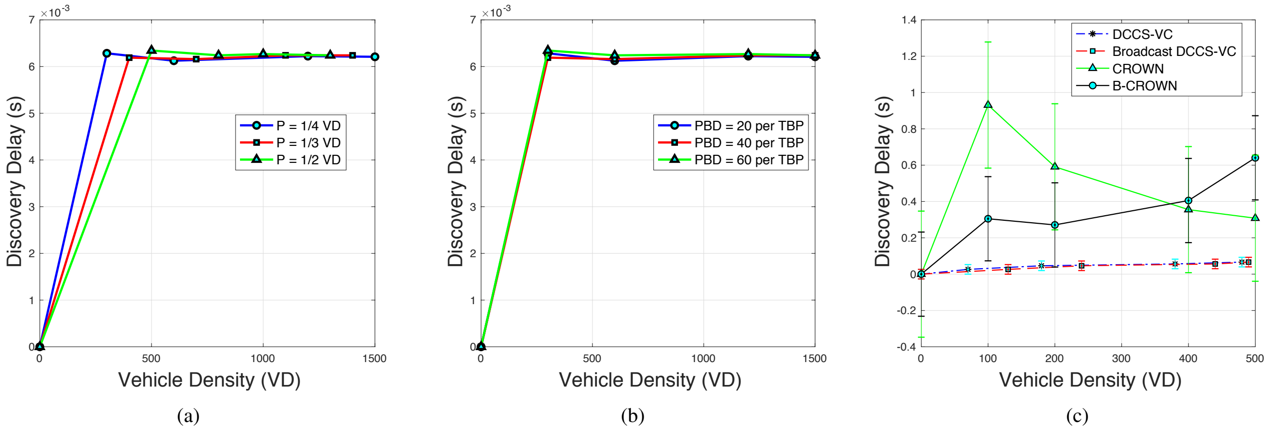


Fig. 4. Performance evaluations of DCCS-VC in terms of Service Discovery Delay. (a) while varying the PV density (P), (b) while varying the Public bus density (PBD), (c) comparison between DCCS-VC, Broadcast DCCS-VC, CROWN and B-CROWN.

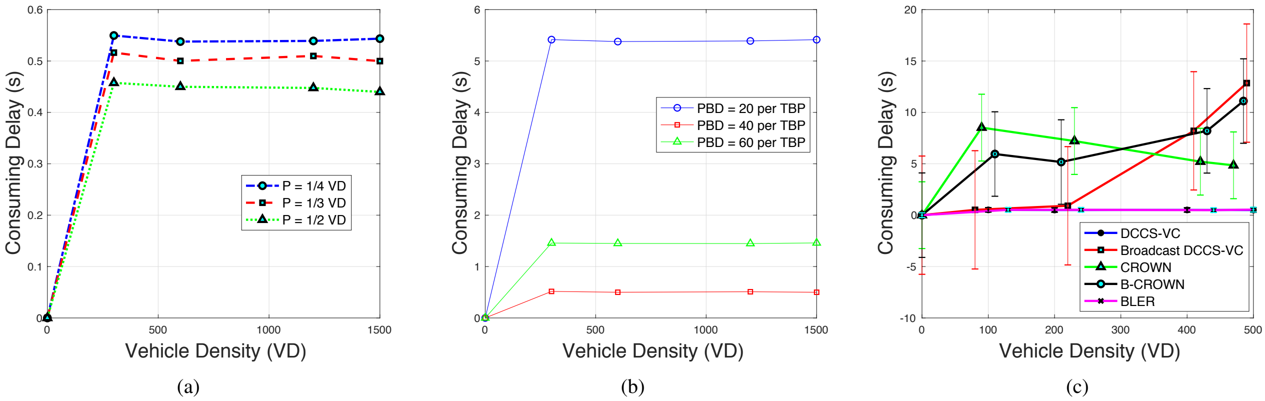


Fig. 5. Performance evaluations of DCCS-VC in terms of Service Consuming Delay: (a) while varying the PV density (P). (b) while varying the Public bus density (PBD) (c) comparison between DCCS-VC, Broadcast DCCS-VC, CROWN, B-CROWN and BLER.

vice discovery and consuming to only between vehicles belonging to the same TBP.

2) The adopted Bus-based routing protocol [38] in which we limit packet routing to only between public buses of the same TBP.

We also note that both DCCS-VC and BLER generate the same performance in terms of both CD and E2E delays (cf. Fig. 5(c) and Fig. 6(c)). In fact, to route a packet, BLER protocol considers buses lines (TBPs) as nodes and forwards the packet from a bus in a given bus line to another bus belonging to another bus line. When the packet reaches the destination bus line, it will be forwarded through public buses belonging to this bus line, until it reaches the destination node. Hence, both DCCS-VC and BLER protocols are practically, based on the same routing mechanism which justifies the obtained results.

Besides, the CD of CROWN is higher than that of DCCS-VC which depends on the routing scheme used in CROWN. In such a scheme, we use stored-and-carried operation to route the packets to their destinations, which increases the CD and the E2ED, especially in a sparse network. We also notice that broadcast DCCS-VC and DCCS-VC have the same CD and E2ED for a VD up to 200 vehicles, beyond which the broadcast DCCS-VC's CD and E2ED increase to very high values. This behavior is due to the broadcasting scheme which generates a high congestion when the VD increases. We note that both B-CROWN and Broadcast DCCS-VC have the same behavior, as they are based on the same communication scheme (Broadcasting).

Figures Fig. 7(a) and (b) depict that the number of consumed services increases as we increase, respectively, the vehicle density (VD) and the public bus density (PBD) in each TBP. It is clear that the higher the number of consumer vehicles is, the higher the number of consumed services is. On the other hand, the presence of more public buses enables registered services to be shared and published efficiently and quickly which also causes the number of consumed services to increase, as shown in Fig. 7(b). Fig. 7(c) shows that both B-CROWN and Broadcast DCCS-VC have almost the same performance which outperform the other protocols. In fact, both B-CROWN and Broadcast DCCS-VC are based on broadcasting operations, which allows all packets to be more forwarded. Hence, the number of consumed services of both B-CROWN and Broadcast DCCS-VC is higher than that of both DCCS-VC and CROWN, but, with a high consuming and E2E delays, as illustrated in Fig 5(c) and 6(c), respectively.

To assess the performance of the optimization technique, Table VII illustrates the ratio of PVs which selected the most

TABLE VII
SUCCESSFUL PVS REGISTRATION RATE (%).

	20 buses per TBP	**40 buses per TBP**	**60 buses per TBP**
$P = 1/4$ VD	60%	70%	75%
$P = 1/3$ VD	58%	65%	75%
$P = 1/2$ VD	50%	59%	67%

suitable bus according to their preferences (cf. Table III). We observe that the ratio increases when we increase the public bus density in each TBP, and it decreases as we increase the PVs density. The presence of more public buses increases the likelihood of a PV to find a public bus satisfying all its preferences. Therefore, we can deduce that DCCS-VC provides stable performance even when we increase the number of consumer vehicles. We can also deduce that DCCS-VC outperforms CROWN protocol in terms of discovering and consuming delays. This is due to the adopted routing scheme which routes packets only through public buses. Consequently, it avoids broadcast storms, and the packets take a short time to reach their destinations compared to CROWN routing protocol. Moreover, the grid-based TBP architecture of DCCS-VC limits the service discovery and consuming to only vehicles belonging to the same TBP, leading also to the E2ED to decrease.

E. Performance Evaluation of Service Provider Selection

Figure Fig. 8 shows the average Service Delay while varying both consumer and provider densities. We note that Neutral mode outperforms the other schemes. In fact, for a given consumer request, the neutral mode selects the first encountered provider vehicle without any QoS aspects, which reduces the generated delay overhead compared to our selection technique or to the negotiation-based scheme. However, the latency introduced in both our scheme and negotiation-based scheme improves the successful execution ratio compared to the neutral mode (cf. Fig. 9). We note that the service delay of the negotiation-based scheme is higher than that of our selection scheme, since it is based on a trusted third vehicle through which providers and consumers communicate. Consequently, it requires more packets to be exchanged in order to reach an agreement about the cloud service. We see also that the service delay increases when we increase the consumer density (CD). But, it decreases as we increase the provider density (P). Increasing number of providers gives to consumers a better chance to find PVs in real time, causing the service delay to decrease. Similarly, Fig. 9 depicts the successful execution ratio of services while varying both consumer and provider densities. As we can see, both our scheme and Negotiation-based scheme provide almost the same ratio which is higher than that of the Neutral mode. As we mentioned before, a consumer vehicle in the Neutral mode selects the first available service provider whatever its QoS. Therefore, a consumer may select a provider offering a poor QoSs which reduces the successful execution ratio. We also remark that the successful execution ratio decreases when we increase the CD and it increases as we increase the PVs density (P). As explained for Fig. 8, these results are essentially due to the increasing number of both requester vehicles and service providers.

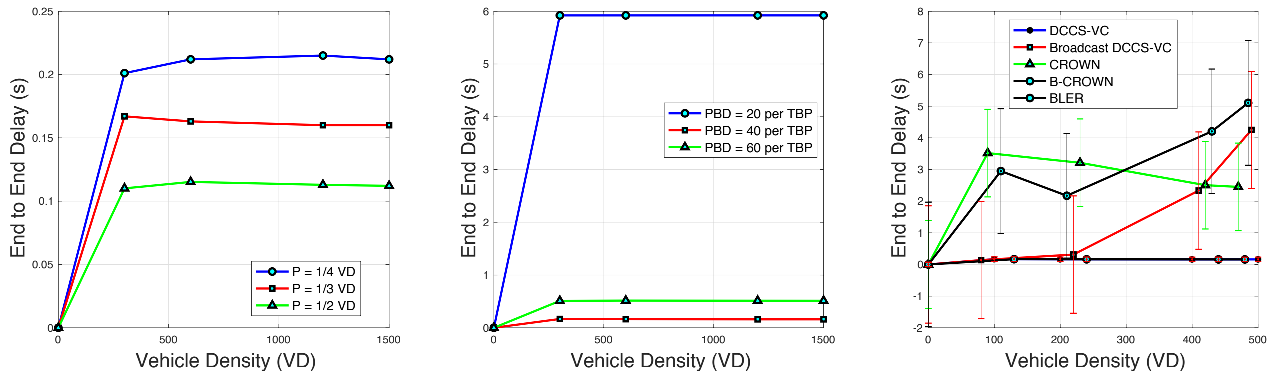
V. CONCLUSION

In this paper, we developed a new cloud service discovery protocol for mobile vehicular cloud, called DCCS-VC, in which vehicles can act as service consumers as well as cloud providers. We described how the public buses can be exploited as cloud directories to improve both services discovering and consuming operations. Thus, we extend our protocol with two more schemes. The former enables service providers to select the most suitable public bus as a cloud directory, while the latter enables an efficient ranking of provider vehicles in order to determine the most satisfactory provider, according to consumer's preferences and service constraints. We have validated the performance of our protocol along with its two schemes throughout simulation experiments and have compared them with other techniques, such as CROWN and B-CROWN protocols, and Negotiation-based and Neutral mode as provider selection schemes. The simulation results showed that DCCS-VC improves greatly service discovery and consuming delays in addition to both cloud directory and service provider selection.

As future work, we plan to extend DCCS-VC protocol to deal with road areas, which are not covered by public buses and to consider security, confidentiality and privacy within its main phases.

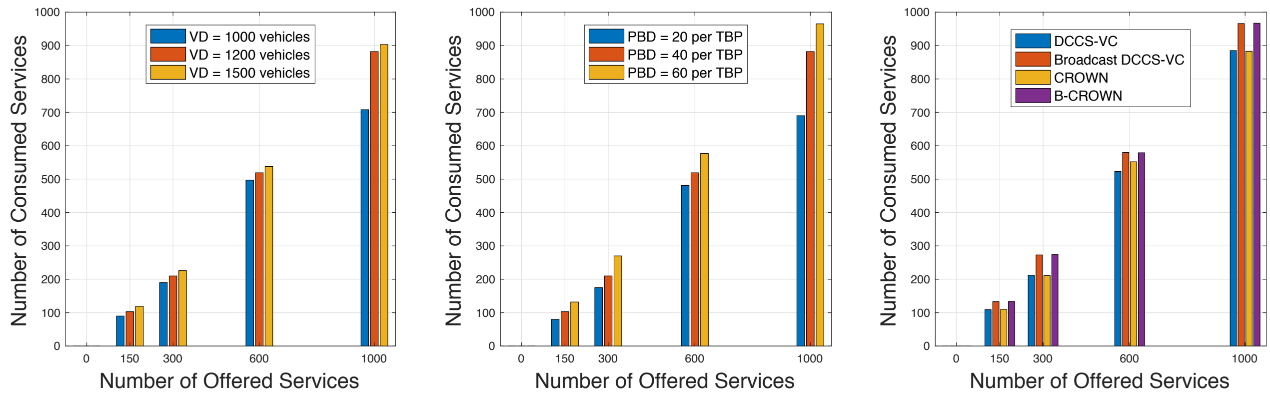
REFERENCES

[1] B. Hayes, "Cloud computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, Jul. 2008. [Online]. Available: http://doi.acm.org/10.1145/1364782.1364786
[2] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 369–392, First 2014.
[3] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *Communications Magazine, IEEE*, vol. 47, no. 11, pp. 84–95, November 2009.
[4] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet," *Computer Networks*, vol. 55, no. 2, pp. 457 – 469, 2011, wireless for the Future Internet. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610003324
[5] "Fred dews. ninety percent of americans drive to work." [Online]. Available: https://www.brookings.edu
[6] M. Gerla, "Vehicular cloud computing," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, June 2012, pp. 152–155.
[7] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, no. 0, pp. 325 – 344, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804513001793
[8] M. Abuelela and S. Olariu, "Taking vanet to the clouds," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, ser. MoMM '10. New York, NY, USA: ACM, 2010, pp. 6–13. [Online]. Available: http://doi.acm.org/10.1145/1971519.1971522
[9] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking vehicular communications: Merging vanet with cloud computing," in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, Dec 2012, pp. 606–609.
[10] B. Brik, N. Lagraa, A. Lakas, and Y. Ghamri-Doudane, "Finding a public bus to rent out services in vehicular clouds," in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*. IEEE, 2015, pp. 1–7.



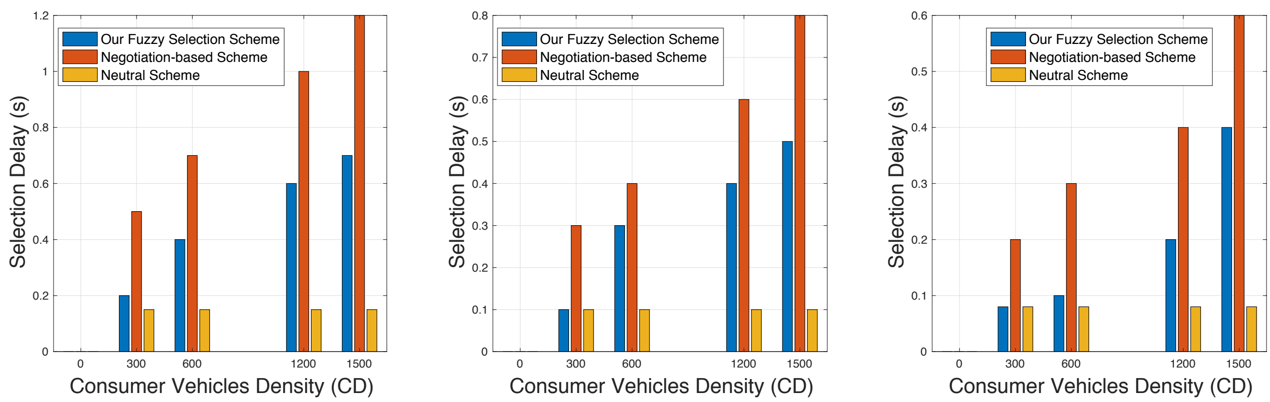
(a) (b) (c)

Fig. 6. Performance evaluations of DCCS-VC in terms of E2E Delay of data packets: (a) while varying the PV density (P). (b) while varying the Public bus density (PBD) (c) comparison between DCCS-VC, Broadcast DCCS-VC, CROWN, B-CROWN and BLER.



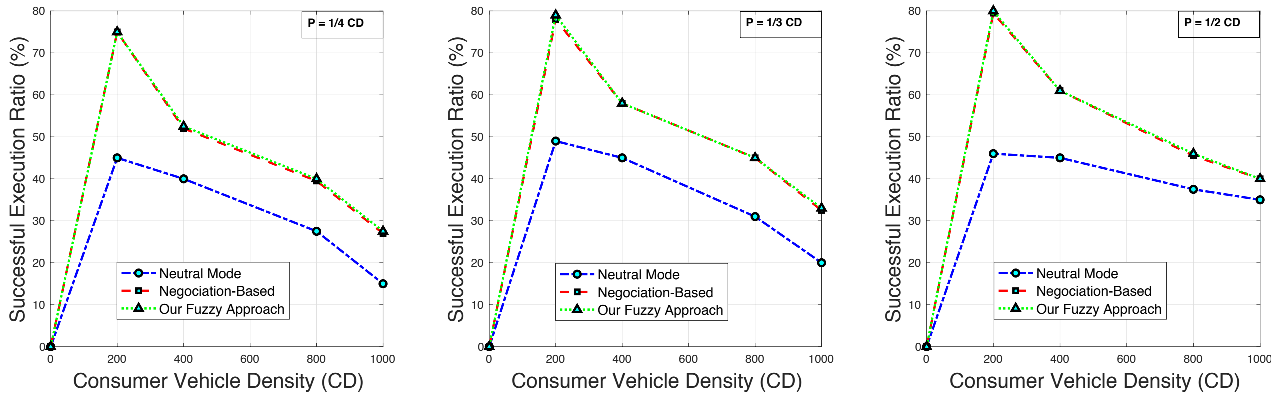
(a) (b) (c)

Fig. 7. Performance evaluations of DCCS-VC in terms of offered services Vs consumed services: (a) while varying the vehicle density (VD), with P = $1/3$ VD and PBD = 40 per TBP. (b) while varying the Public bus density (PBD), with P = $1/3$ VD and VD = 1200 vehicles (c) comparison between DCCS-VC, Broadcast DCCS-VC, CROWN and B-CROWN, with P = $1/3$ VD and PBD = 40 per TBP, and VD = 1200 vehicles.



(a) (b) (c)

Fig. 8. Service delay comparison between our fuzzy selection technique, Negotiation-based and Neutral mode, with: (a) P = $1/4$ CD, (b) P = $1/3$ CD, (c) P = $1/2$ CD.



(a) (b) (c)

Fig. 9. Successful Execution Ratio comparison between our selection technique, Negotiation-based and Neutral mode.

[11] B. Brik, N. Lagraa, Y. Ghamri-Doudane, and A. Lakas, "Finding the most adequate public bus in vehicular clouds," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Oct 2016, pp. 67–74.

[12] N. Tamani, B. Brik, N. Lagraa, and Y. Ghamri-Doudane, "Vehicular cloud service provider selection: A flexible approach," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[13] K. Y. C.L. Hwang, "Multiple criteria decision making. lecture notes in economics and mathematical systems." 1981.

[14] S. Bitam, A. Mellouk, and S. Zeadally, "Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 96–102, February 2015.

[15] D. Baby, R. D. Sabareesh, R. A. K. Saravanaguru, and A. Thangavelu, *VCR: Vehicular Cloud for Road Side Scenarios*, 2013, pp. 541–552.

[16] H. Abid, L. T. T. Phuong, J. Wang, S. Lee, and S. Qaisar, "V-cloud: Vehicular cyber-physical systems and cloud computing," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, ser. ISABEL '11. ACM, 2011, pp. 165:1–165:5.

[17] S. Kumar, S. Gollakota, and D. Katabi, "A cloud-assisted design for autonomous driving," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 41–46.

[18] X. Hu, L. Wang, Z. Sheng, P. TalebiFard, L. Zhou, J. Liu, and V. C. Leung, "Towards a service centric contextualized vehicular cloud," in *Proceedings of the Fourth ACM International Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, ser. DIVANet '14. New York, NY, USA: ACM, 2014, pp. 73–80.

[19] K. Mershad and H. Artail, "Finding a star in a vehicular cloud," *Intelligent Transportation Systems Magazine, IEEE*, vol. 5, no. 2, pp. 55–68, Summer 2013.

[20] R. Hussain, F. Abbas, J. Son, S. Kim, and H. Oh, "Using public buses as mobile gateways in vehicular clouds," in *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, Jan 2014, pp. 175–176.

[21] R. Hussain, F. Abbas, J. Son, D. Kim, S. Kim, and H. Oh, "Vehicle witnesses as a service: Leveraging vehicles as witnesses on the road in vanet clouds," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 1, Dec 2013, pp. 439–444.

[22] H. R. Arkian, R. E. Atani, A. Diyanat, and A. Pourkhalili, "A cluster-based vehicular cloud architecture with learning-based resource management," *The Journal of Supercomputing*, vol. 71, no. 4, pp. 1401–1426, 2015.

[23] Y.-W. Lin, J.-M. Shen, and H.-C. Weng, "Cloud-supported seamless internet access in intelligent transportation systems," *Wireless Personal Communications*, vol. 72, no. 4, pp. 2081–2106, 2013.

[24] M. Azizian, S. Cherkaoui, and A. Hafid, "An optimized flow allocation in vehicular cloud," *IEEE Access*, vol. 4, pp. 6766–6779, 2016.

[25] P. Ghazizadeh, R. Mukkamala, and S. El-Tawab, "Scheduling in vehicular cloud using mixed integer linear programming," in *Proceedings of the First International Workshop on Mobile Sensing, Computing and Communication*, ser. MSCC '14. New York, NY, USA: ACM, 2014, pp. 7–12.

[26] P. Ghazizadeh, S. Olariu, A. G. Zadeh, and S. El-Tawab, "Towards fault-tolerant job assignment in vehicular cloud," in *2015 IEEE International Conference on Services Computing*, June 2015, pp. 17–24.

[27] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "Multiagent/multiobjective interaction game system for service provisioning in vehicular cloud," *IEEE Access*, vol. 4, pp. 3153–3168, 2016.

[28] K. Mershad, H. Artail, , and M. Gerla, "We can deliver messages to far vehicles," *IEEE Trans Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1099–1115, Sept 2012.

[29] B. Brik, N. Lagraa, A. Lakas, and Y. Ghamri-Doudane, "Rcs-vc: Renting out and consuming services in vehicular clouds based on lte-a," in *Global Information Infrastructure and Networking Symposium (GIIS), 2015*. IEEE, 2015, pp. 1–6.

[30] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, and S. Parkvall, "Lte: the evolution of mobile broadband," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 44–51, April 2009.

[31] P. Ghazizadeh, R. Florin, A. G. Zadeh, and S. Olariu, "Reasoning about mean time to failure in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 751–761, March 2016.

[32] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "An auction-driven multi-objective provisioning framework in a vehicular cloud," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec 2015, pp. 1–6.

[33] C. J. Watkins and P. Dayan, "Technical note: Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[34] L. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[35] L. Liétard, "Contribution à l'interrogation flexible de bases de données: Etude des propositions quantifiées floues," Ph.D. dissertation, 1995.

[36] R. R. Yager, "General multiple-objective decision functions and linguistically quantified statements," *International Journal of Man-Machine Studies*, vol. 21, no. 5, pp. 389–400, 1984.

[37] ——, "Fuzzy quotient operators for fuzzy relational data bases," *Fuzzy Engineering toward Human Friendly Systems*, vol. 1, pp. 13–15, 1991.

[38] W. Tonghong Li ; Hazra, S.K. ; Seah, "A position-based routing protocol for metropolitan bus networks," in *Proceedings of the 61st International Conference on Vehicular Technology Conference*. IEEE, 2005, pp. 2315 – 2319.

[39] "Sumo, simulation for urban mobility," http: //sourceforge.net/apps/mediawiki/sumo, [Online].

[40] "OMNET++ simulateur," http://docplayer.fr/13487930-Simulateur-omnet.html., [Online].

[41] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments," in *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*, Jan 2011, pp. 84–90.

[42] M. Sede, X. Li, D. Li, M. Y. Wu, M. Li, and W. Shu, "Routing in large-scale buses ad hoc networks," in *2008 IEEE Wireless Communications and Networking Conference*, March 2008, pp. 2711–2716.

Bouziane Brik received an engineering degree in computer science, ranked first in his class, from university of Laghouat, Algiers, Algeria, in 2010, a Magister degree from university of Laghouat, Algeria in 2013, and a Ph.D. degree from the same university in 2017. His research interests include Wireless Sensor Networks (WSN), Vehicular Networks, Vehicular Cloud, Internet of Things (IoT), IoT in industrial systems, Smart grid. He also acted or still acts as reviewer of the following IFIP, ACM or IEEE conferences (ICC, Globecom, PIMRC, WCNC, VTC, IM/NOMS, IWCMC, GIIS, WiMob, Wireless Days,...) and workshops (WICTD, NiVi, IVCS, On-MOVE, Vehi-Mobi, DANMS, UBIROADS, SSMO, ...). He is a Member of IEEE.

Nasreddine Lagraa received his PhD in automation from the Ecole Nationale Polytechnique, Algeria (2008). He is currently the director of the Laboratoire d'Informatique et de Mathematiques, University of Laghouat, Algeria. He is teaching various courses on computer architecture, multimedia, mobile networks and network security. His research interests are: Vehicular networks, network security, decentralized control, and fuzzy and artificial neural networks. He is serving in the technical program committees of many international conferences, Wireless and Mobile Computing, Networking and Communications (WiMob), and Innovations in Information Technology (IIT).

Nouredine Tamani (Dr. Eng.) received his Ph.D. degree in computer science from the university of Rennes (France) in 2012. His research activities encompass the following main topics: knowledge representation (DL, DataLog+-, ontologies), inconsistency handling, user preferences and flexible querying, fuzzy sets, fuzzy logics, and decision support systems. Currently, he works at the Laboratoire Informatique, Image et Interaction (L3i), university of La Rochelle (France). He does research in user behaviour detection within IoT based applications for user data protection and user privacy preservation, reasoning under uncertainty for context explanation, and aggregation operators for decision support systems. Nouredine has been involved as a reviewer for many journals such as Knowledge and Information Systems, Journal of Transport Policy, Computers and Electronics in Agriculture, etc. and as an external reviewer for many conferences such as IEEE Fuzzy set and Systems, IEEE Global Communications Conference, Information Processing and Management of Uncertainty in Knowledge-Based Systems, etc.

Yacine Ghamri-Doudane is currently Full Professor at the University of La Rochelle (ULR) in France, and the director of its Laboratory of Informatics, Image and Interaction (L3i). Before that, Yacine held an Assistant/Associate Professor position at ENSIIE, a major French post-graduate school located in Evry, France, and was a member of the Gaspard-Monge Computer Science Laboratory (LIGM - UMR 8049) at Marne-la-Valle, France. From February 2011 till July 2012, he was regularly visiting the Performance Engineering Laboratory of University College Dublin, Dublin, Ireland. Yacine received an engineering degree in computer science from the National Institute of Computer Science (INI), Algiers, Algeria, in 1998, an M.S. degree in signal, image and speech processing from the National Institute of Applied Sciences (INSA), Lyon, France, in 1999, a Ph.D. degree in computer networks from University Pierre & Marie Curie, Paris 6, France, in 2003, and a Habilitation to Supervise Research (HDR) in Computer Science from Universit Paris-Est, in 2010. His current research interests lays in the area of wireless networking and mobile computing with a current emphasis on topics related to the Internet of Things (IoT), Wireless Sensor Networks (WSN), Vehicular Networks as well as Digital Trust. Yacine holds three (3) international patents and he authored or co-authored seven (7) book chapters, more than 30 peer-reviewed international journal articles and more than 130 peer-reviewed conference papers. Yacine also acted as the Chair the IEEE Communications Society (ComSoc) Technical Committee on Information Infrastructure & Networking (TCIIN - previously TCII) from January 2010 till December 2013 and also chaired the IEEE ComSoc Humanitarian Communications Technologies Ad hoc Committee (HCTC) from January 2012 till December 2015. He is or had been an editorial board member of the Elsevier JNCA, Elsevier ComNet, Springer AoT Journals, Wiley WCMC, Guest Editor of IEEE ComMag, IEEE IoT Journal, Springer/EURASIP WCN Journal, co-Editor in Chief of the Elsevier/KICS ICT Express Journal as well as the founding editor-in-chief of the IEEE ComSoc Ad Hoc and Sensor Network Technical Committee (AHSN TC) Newsletter. Among other conference involvements, he acted or is still currently acting as the TPC Chair of IEEE CCNC 2015, Symposium co-Chair in IEEE ICC 2009, 2010, 2012 and 2018 as well as IEEE GLOBECOM 2012 and 2015. He is a Member of IEEE.

Abderrahmane Lakas received his MS (1990) and PhD (1996) in computer systems from the University of Paris VI, Paris, France. He joined the College of Information Technology, United of Arab Emirates University in 2003. He is teaching various courses on computer networks and network security. Dr. Lakas had many years of industrial experience holding various technical positions in telecommunication companies such as Netrake (Plano, Texas, 2002), Nortel (Ottawa, 2000), and Newbridge (Ottawa, 1998). He spent two years (19941996) as a research associate at the University of Lancaster, UK. Dr. Lakas has been conducting research in the areas of network design and performance, voice over internet protocol, quality of service, and wireless networks. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and IEEE Communications Society. Dr. Lakas is in the editorial board of Journal of Communications (Actapress) and Journal of Computer Systems, Networks, and Communications (Hindawi). He is serving on the technical program committees of many international conferences Global Telecommunications (GlobCom), International Conference on Communications (ICC), Vehicular Technologies Conference (VTC), among many others.