

# VFogSim: A Data-driven Platform for Simulating Vehicular Fog Computing Environment

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

04-01-2022 / 27-04-2022

CITATION

Akgul, Ozgur Umut; Mao, Wencan; Cho, Byungjin; Xiao, Yu (2022): VFogSim: A Data-driven Platform for Simulating Vehicular Fog Computing Environment. TechRxiv. Preprint.  
<https://doi.org/10.36227/techrxiv.17829398.v2>

DOI

[10.36227/techrxiv.17829398.v2](https://doi.org/10.36227/techrxiv.17829398.v2)

# VFogSim: A Data-driven Platform for Simulating Vehicular Fog Computing Environment

Özgür Umut Akgül, Wencan Mao, Byungjin Cho, and Yu Xiao

**Abstract**—Edge/fog computing is a key enabling technology in 5G and beyond for fulfilling the tight latency requirements of emerging vehicle applications, such as cooperative and autonomous driving. Vehicular Fog Computing (VFC) is a cost-efficient deployment option that complements stationary fog nodes with mobile ones carried by moving vehicles. To plan the deployment and manage the VFC resources in the real world, it is essential to take into account the spatio-temporal variations in both demand and supply of fog computing capacity and the trade-offs between achievable Quality-of-Services and potential deployment and operating costs. Concerning the complexity and the economic load of real-world measurements, simulation becomes a better option at the early research phase to validate capacity and resource management solutions in various urban environments. The existing simulation platforms cannot provide a realistic techno-economic investigation to analyze the implications of VFC deployment options, due to the simplified network models in use, the lack of support for fog node mobility, and limited testing scenarios. In this paper, we propose an open-source simulator VFogSim that allows real-world data as input for simulating the supply and demand of VFC in urban areas. It follows a modular design to evaluate the performance and cost-efficiency of different deployment scenarios under various vehicular traffic models, and the effectiveness of the diverse network and computation schedulers and prioritization mechanisms under user-defined scenarios. Compared with the existing edge/fog computing simulators, such as iFogSim, IoTsim, and EdgeCloudSim, to the best of our knowledge, our platform is the first one that supports the mobility of fog nodes and provides realistic modeling of V2X in 5G and beyond networks in the urban environment. Furthermore, we validate the accuracy of the platform using a real-world 5G measurement and demonstrate the functionality of the platform taking VFC capacity planning as an example.

## I. INTRODUCTION

Increasing demand for high-complexity but low-latency computation driven by emerging vehicular applications stimulates mobile edge/fog computing (MEC) use, which brings cloud-like computing services closer to the data producers, i.e., client vehicles. Conventional MEC paradigms rely on stationary edge/fog computing nodes, i.e., cellular fog nodes (CFNs), to provide resource and service availability to the client vehicles. To ensure service continuity, the service providers have to plan and provision the resources considering the evolution of the demand. The inflexible deployment strategy and uncertainty over the return on the investment put an additional economic load on service providers and turn service provisioning into a non-profitable business model. Even with idealistically good planning and provisioning of the computational resources, the stationary nature of MEC limits its applicability on high mobility use cases, e.g., cooperative

and autonomous driving. Vehicular fog computing (VFC), motivated by the techno-economic pressure on MEC deployment, proposes complementing CFNs with fog nodes carried by moving vehicles, i.e., vehicular fog nodes (VFNs), to enable more flexible and cost-efficient deployment of computing resources [1]. For example, VFNs can be used to respond to excess demand at peak hours of the day and during special events [2].

Despite being a promising solution, developing and evaluating the capacity and resource management strategies for VFC remains a challenging topic. Minimizing costs while increasing the quality of service (QoS) requires tackling various research questions, including where and how much capacity to deploy [3], [4], whether and where to offload tasks [5], [6], or how to schedule the radio and computing resources jointly [7]. One common challenge of these research questions is how to assess the system performance for various conditions. Real-world experiments provide an accurate evaluation by demonstrating the advantages and the risks of the proposed systems. In a real-world test environment, as the primary research questions are investigated, the impacts of secondary aspects (e.g. the weather conditions) can be measured. However, real-world test environments are usually quite expensive and difficult to set up. Moreover, in most cases, it is not possible to obtain isolated and dedicated test environments to understand the value of VFC in different scenarios. The need to have a deeper understanding of the deployment decisions under extreme circumstances (e.g., ultra-dense, fast-moving, unbalanced loads), as well as the low economic and time requirements, have increased the attention on the simulators over the research community.

There are already several simulation environments for dynamic edge computing, e.g., iFogSim [8], EdgeCloudSim [10], and FogNetSim++ [11]. While these simulators can capture the general attributes of networking (e.g., QoS measurement or the mobility of client vehicles), they are missing several fundamental features, such as mobility of computing nodes [8], [10] or service differentiation [11]. Thus, the existing simulators could not provide an option to investigate the impacts of economic decisions (e.g., different pricing strategies) on the applicability of VFC. This paper aims at solving this challenge by developing an open platform that provides a realistic simulation of dynamic VFC environments and supports the simulation of various application scenarios. The platform proposed in this paper uses a data-driven approach where the real-world or synthetic data can be used. The input data consists of the signal to interference-plus-noise-ratio (SINR) and vehicular traffic locations (i.e., obtained from GPS). We

Simulator	CP	RA	QoS metrics	CM	FM
IFogSim [8]	No	Yes	Latency, network congestion	No	No
IoTsim [9]	Yes	No	Latency	No	No
EdgeCloudSim [10]	Yes	No	Latency, acceptance ratio	Yes	No
FogNetSim++ [11]	Yes	Yes	Latency, packet loss ratio, handover	Yes	No
EmuFog [12]	Yes	No	Latency	No	No
Fogbed [13]	Yes	No	Latency	No	No
FogTorch [14]	Yes	No	Latency, bandwidth	No	No
FogTorchII [15]	Yes	No	Latency, bandwidth	No	No
Veins [16]	No	Yes	Latency, packet loss ratio, throughput	Yes	Yes
<b>VFogSim</b>	<b>Yes</b>	<b>Yes</b>	<b>Latency, network congestion, acceptance ratio, economic aspects</b>	<b>Yes</b>	<b>Yes</b>
CP: Evaluation of capacity planning strategies. RA: Evaluation of resource allocation strategies.			CM: Integration of mobility model of client vehicles. FM: Support of mobility of fog nodes.		

TABLE I: Comparison between our work with the existing edge/fog simulators.

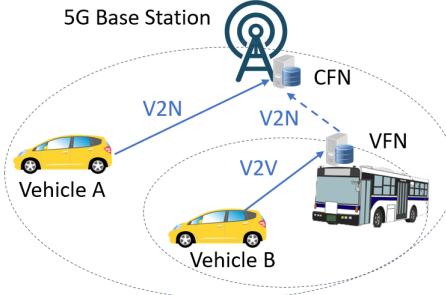


Fig. 1: An application scenario of VFC.

use SUMO [17] and WinProp [18] to generate synthetic data. Unlike the other simulation environments, VFogSim supports both stationary and mobile fog nodes and integrates vehicular traffic and network simulation covering physical and upper-layer protocols. It is embedded with both spectral and computational resource allocation policies and can be customized by user-defined algorithms. Finally, our simulator can be used for evaluating various aspects, including the QoS metrics (such as data rate, average delay), the techno-economic performance of the different network and task allocation policies, and the impacts of various factors such as inter-service prioritization, traffic load, and pricing strategy.

To the best of our knowledge, it is the first data-driven VFC simulation platform. Our key contributions are as follows.

- We develop a data-driven system-level simulation platform, VFogSim, for evaluating different VFC deployment options and resource schedulers. The developed platform is made open source<sup>1</sup> to invite the research community for future developments.
- The simulator follows the modular design principles, which allow it to be easily customized for particular use cases or input data.
- We evaluate accuracy of the wireless network simulation using the real-world measurements from a commercial network in Otaniemi, Finland.
- We demonstrate the functionality of VFogSim through a case study on VFC deployment in an urban area, the simulation results provide insights into the techno-economic implications of different capacity planning options.

<sup>1</sup>The current version of the simulation platform is available at [https://mobilecloud.aalto.fi/?page\\_id=1441](https://mobilecloud.aalto.fi/?page_id=1441)

The remainder of this paper is organized as follows. Section II gives an overview of the background and reviews the related works. Section III presents the platform architecture and an example scenario. While Section IV details the different blocks and how they are implemented, Section V evaluates the accuracy and the functionality of our simulator. Section VI discusses the key features of the platform and future work. Finally, Section VII concludes the work.

## II. BACKGROUND AND RELATED WORKS

In this section, we first introduce the motivation and exemplary scenario of VFC, then we compare our simulation platform with the state-of-the-art works.

### A. Vehicular Fog Computing

The conventional edge/fog computing strategy heavily relies on stationary fog node deployments, which are designed according to the peak demand on the computing resources [19]. However, this static allocation of the computing resources almost always leads to the over-provisioning of the resources, turning service provisioning into a non-profitable business model. One solution, motivated by the spatio-temporal variations in vehicle traffic in urban environments [4], [19], is the utilization of VFNs to complement the stationary fog nodes (i.e., CFNs in this case) at the peak demand hours. The VFC concept is built upon the collaboration between the CFNs and VFNs. Fig. 1 presents an application scenario of VFC, where Vehicle A and B can either offload their tasks to a CFN co-located with a 5G base station or a VFN carried by bus according to their performance metrics. VFC can be used for emerging vehicular applications involving time-critical and data-intensive computational tasks. Table II lists some exemplary applications and their corresponding latency requirements.

### B. Related Works

Current edge/fog computing simulators are mainly built on existing cloud computing simulators or network simulators. Table I presents a comparison of VFogSim with the state-of-the-art. Three well-known edge computing simulators are IFogSim [8], IoTsim [9], and EdgeCloudSim [10], all of which are built on CloudSim [22]. IFogSim models the fog environment where the fog nodes follow a hierarchical arrangement

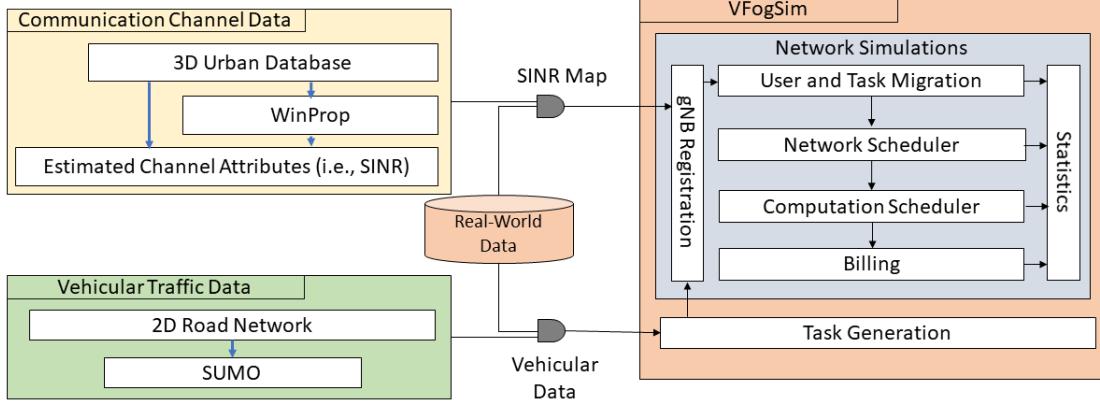


Fig. 2: System architecture of the VFogSim platform.

Application	Delay requirements	Type
Collision avoidance	Bound, 10ms	Safety
Vehicle platooning	Bound, 25ms	Safety
Collective perception	Bound, 100ms	Safety
Information sharing	Average, 250ms-500ms	Safety/Non-safety
Vehicle scheduling	Average, 1s	Non-safety
AR/VR	Bound, 10ms	Entertainment
Cloud gaming	Average, 100ms-1s	Entertainment

TABLE II: Exemplary vehicular applications and their delay requirements [20], [21].

from the sensors to the cloud and measure the impacts of resource management policies in terms of latency, network congestion, energy consumption, and cost [8]. IoTsim supports the simulation of IoT big data processing using the MapReduce model [9]. However, they simplify the network model and do not consider the communication channel attributes (e.g., SINR). EdgeCloudSim integrates multiple modules into an edge computing system, including the core simulation (i.e., the module responsible for loading and running the edge computing scenarios from the configuration files), networking, load generator, mobility, and edge orchestrator modules [10]. Despite considering the mobility of client vehicles, it does not support the mobility of fog nodes.

FogNetSim++ [11] is an edge computing simulator based on OMNeT++ [23], focusing on simulating the network characteristics of distributed edge computing devices and enabling users' customization of mobility models and fog node scheduling algorithms [11]. However, it cannot be used to estimate the network metrics (e.g., throughput) due to a lack of physical-layer protocols. Similarly, EmuFog [12] and Fogbed [13] are two edge computing simulators based on the network simulator Mininet [24] and its extended version MaxiNet [25]. EmuFog enables users to design the network topology with embedded fog nodes and run Docker-based applications on those nodes connected by an emulated network [12]. Fogbed enables the dynamic adding, connecting, and removing of virtual nodes via Docker containers and supports to perform real-world protocols and services [13]. Although they support the evaluation of cost and latency, they do not support other advanced functions (e.g., customizable scheduler, pricing strat-

egy, inter-service prioritization).

FogTorch is a simulation tool that supports QoS-aware deployment of IoT applications to fog infrastructures [14]. Extending FogTorch, FogTorchII exploits Monte Carlo simulations to take into account variations of the QoS and classifies deployments in terms of both QoS assurance and fog resource consumption [15]. Nevertheless, neither of the works have integrated the mobility model of client vehicles or taken the mobility of the fog nodes into account. To simulate the vehicular network, Veins couples OMNeT++ [23] with the mobility simulator SUMO [17] with the implementation of IEEE 802.11p [16]. It can also be used together with SimuLTE [26], which offers a detailed model of the Long-Term Evolution (LTE). However, it does not support the simulation of 5G and beyond networks. In this work, we integrate SUMO with WinProp [18]. While SUMO enables a fine-grained simulation of vehicular traffic, WinProp supports a variety of air interfaces including IEEE 802.11p, 3G/4G/5G, and even 6G in the future. Moreover, WinProp takes various real-world information such as the locations of the base stations, buildings, and trees into consideration, thus offering a more realistic and widely applicable vehicular network simulation.

### III. SYSTEM ARCHITECTURE & EXAMPLE SCENARIO

This section gives an overview of the modular and customizable platform, VFogSim. We introduce the system architecture following by an example use case.

#### A. System Architecture

As illustrated in Fig. 2, the VFogSim platform requires the GPS coordinates of the vehicles and a SINR map for the given region. Therefore, the first step of running a simulation is to provide this input data. We design the VFogSim platform as a discrete-time optimizer, so the input data need to be discretized into transmission time intervals (TTIs). Once this data is inputted, the vehicles are registered to different base stations, i.e., gNBs. Note that in synthetic data, it is possible to associate the users with specific base stations automatically. At each base station, we first update the active user information

to determine the task migrations, i.e., when the vehicle moves to another base station before completing the active task. Once the active tasks are determined, each base station performs the network and computation scheduling and determines resource allocations to vehicles. The resulting system behaviours, such as resource allocation, delay, billing information, are stored in statics, respectively, to be analyzed at the end of the simulation.

**Input Data of VFogSim.** VFogSim can take as input either real-world measurement or synthetic data of SINR maps and vehicular locations. To generate the synthetic data, in this paper, we use WinProp [18] and SUMO [17]. The SINR map is generated in the following steps. To model the physical aspects, such as reflections and shadowing in the urban scenario, we first use the 3D map of the considered area extracted from OpenStreetMap [27] as input to generate the urban database in WinProp [18]. The database contains the locations and heights of the buildings and vegetation objects (e.g., trees) and calculated transmission loss caused by them. To simulate the sites and antennas realistically, we use the aggregated real-world data, including the location information, height, frequency, antenna type, and carrier parameters in the base stations. Finally, we consider the propagation model and air interface configuration to estimate the SINR map of the region. For tractability, we omit the secondary reflections (e.g., other vehicles) on the SINR map. Meanwhile, based on the 2D road network from OpenStreetMap, SUMO generates a fine-grained microscopic traffic simulation that includes the vehicular movement and the traffic infrastructure (e.g., traffic lights). Alternatively, if we use the vehicular GPS data, and the SINR data collected from the real-world experiments, both information needs to be associated with the TTIs. Particularly, we need the SINR information of the vehicle at every TTI. Apart from this association and time discretization, both synthetic and real-world data can directly be used.

**Task Generation.** The task generation block associates the vehicular traces with specific task requests and creates the network load. We discuss the specific association strategy in Section IV-B. It is possible to customize the tasks in terms of their CPU/GPU/NPU computing units, battery consumption, memory requirements, or inter-arrival times. The generated tasks, the vehicle ID, and the location of the vehicle are passed to the gNB registration block.

**Network Simulations.** The network simulations start with associating the vehicular traces with the SINR values at the gNB registration block. This block is also responsible for determining the load at different gNBs. If a client vehicle changes gNBs, the gNB registration block triggers the user and task migration block where the task migration tasks are handled and stored. Otherwise, the tasks are moved to the schedulers.

**Network and Computation schedulers.** The network scheduler and computation scheduler focus on the allocation of spectral and computational resources, respectively. As we perform the resource allocations, we assume that the priority of a service is determined by the requested resources, the price of the service, and the remaining service execution time. We have separated the network and computation schedulers

to ensure that different scheduling strategies can be covered using VFogSim. The information regarding the user and task migration, the resource (both spectral and computing) allocation decisions, and the billing information is stored in the statistics block for evaluation purposes, e.g., efficiency in network orchestration and resource management [28].

### B. Example Scenario

The modular and customizable structure of VFogSim enables simulating a large variety of testing scenarios. The blocks presented in Fig. 2 can be either used as the default mode (i.e., presented in the paper) or can be customized to test specific scenarios (i.e., by changing one of multiple blocks). In the default mode, the simulator stores various statistics, including the total delay, the data rate, and the total revenue accumulated by the operator.

Two testing scenarios for VFC are capacity planning and resource allocation. Capacity planning aims to determine where to deploy the fog nodes and how much should be deployed with cost-efficiency and the QoS guarantee. For example, Mao et al. proposed a capacity planning framework for VFC in order to minimize the costs while meeting the latency requirements [4]. Resource allocation aims to find the matching strategies between the fog nodes and the client vehicles in order to maximize the QoS. For example, Zhu et al. proposed a resource allocation algorithm for video crowd-sourcing tasks in a VFC environment in order to joint minimize video quality and latency [5]. To test different capacity planning strategies, the computation scheduler could be changed. On the other hand, to test the techno-economic performances of different resource allocation strategies, both the network scheduler and the computation scheduler could be replaced.

To demonstrate an example scenario in this paper, we consider the capacity planning for VFC in an urban area, as detailed in Section V. For the sake of simplicity, each user is assumed to be associated with one active service at each TTI. The SINR of user  $k$  is estimated based on the vehicle location and the SINR map. At every TTI, the simulator will first generate the active tasks and then perform the network scheduling per cell. Among the active services, the task scheduling algorithm is run for the computational resources. The user is queued if it fails to receive the computational resources. To ensure tractability, this work assumes that the respective containerized vehicular applications are active at all the fog nodes (including CFNs and VFNs). Consequently, the migration delay is the time it takes to move the user data from one fog node to another. This work assumes that the client vehicles are always connected to the cells with the highest SINR values.

## IV. SIMULATOR DESIGN

In this section, we detail the data-driven simulator design and the key attributes of the default mode.

### A. Generating Vehicular Traffic

We generate the synthetic vehicular trajectories in a region in two steps. In the first step, we feed the 2D map of the

TABLE III: Transmission data rate expectations  $R_1 - R_3$  (bps/Hz), utility values  $U_1 - U_3$ , Average task size  $C_{\text{avg}}$  (kB/task), delay expectations  $W_s$  (TTI), and price  $p$  (MU/task) for four exemplary service types. When the service differentiation is turned off, the  $R$  and  $U$  values for all the services are set according to  $A_1$ . When the pricing is turned off, the  $p$  values for all the services are set as 1 MU.

Service Type	$R_1$	$R_2$	$R_3$	$U_1$	$U_2$	$U_3$	$C_{\text{avg}}$	$W_s$	$p$
$A_1$ : Background	0.05	0.07	0.07	0	1	1	0	1	1
$A_2$ : Object detection	0.01	0.08	0.4	-1	0.7	1	500	1	2
$A_3$ : Lane detection	0.1	0.23	0.55	-0.5	0.7	1	1000	1	5
$A_4$ : Video transcoding	0	1.08	$\infty$	0	1	$\infty$	1500	3	10

region into SUMO. The road network generated by SUMO from the 2D map contains the information of the nodes (e.g., road intersections), the edges (e.g., road segments), and the relationship between the edges (e.g., junctions). While generating the road network, we include the road information (e.g., the number of lanes), road type (e.g., motorway), and traffic regulations (e.g., one-way or both). Additionally, the road network contains information on the traffic lights at each intersection and generates the corresponding time schedules.

In the second step, the trips of the client vehicles are generated using the built-in function *randomTrips*. The number of vehicles simulating at each time is controlled by the arrival and departure rates. More specifically, if there are  $n$  users in the setting, then the arrival rate is  $n$  vehicles/second. We control the arrival period to be very small (0.01 second) so that all the  $n$  vehicles appear at the same time. The *randomTrips* function will randomly choose the road segments as the origin and destination of each vehicle's trip, and route the vehicles according to the shortest path. The vehicle will disappear after it has arrived at its destination. The generation of the trips also considers the number of lanes as weight so that the traffic density on the primary roads is generally higher than on the secondary roads. Moreover, we use different arrival and departure rates to simulate various traffic conditions such as peak hours, off-peak hours, and extreme scenarios (e.g., when the request arrival rate is extremely high). In this work, VFNs are assumed to have regular routes and time schedules (e.g., carried by buses), where their routes are generated using the *origin – destination(OD)* matrix. More specifically, we set the origins and destinations of all the buses, and generate their routes by *dumarouter* which calculates the shortest paths.

### B. Modeling Communication Channel

We built an SINR map of the region in question using Altair WinProp [18], taking the 3D map of the area as input. The 3D map (i.e., from OpenStreetMap) is used for estimating the transmission loss caused by reflections and shadowing in the propagation model. We place the base stations according to their real-world locations. For example, the locations of the base stations are extracted from the street addresses in *CellMapper* [29] and exported to Universal Transverse Mercator (UTM) coordinate system using *geocoding*. The air interface is configured according to the 5G configuration in [18]. The network attributes, such as the cell coverage and SINR values, have the spatial granularity of 1 m<sup>2</sup>. The client vehicles can customize various physical network parameters such as the network type, multiple access, duplex type, frequency band, channel bandwidth, and transmission power.

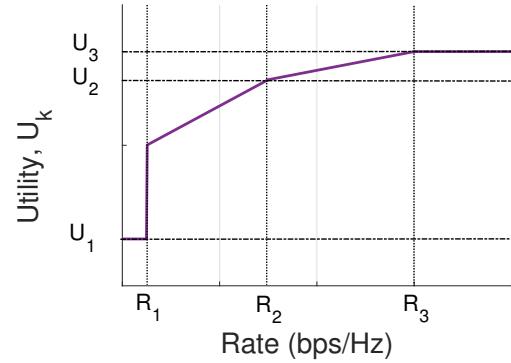


Fig. 3: Generic utility function from [7].

Combining the SINR map with the vehicular traces, we calculate the achievable rate of each vehicle at each time slot (i.e., TTI) over the simulation horizon using the Shannon Formula, i.e., presented in (2).

To represent the relationship between the transmission data rate expectations  $R$  and utility values  $U$ , we consider a piecewise linear utility function proposed in [7] (cf., Fig. 3). The utility values reflect the QoS received by the client vehicles. This utility function is determined by six parameters, namely  $R_1$ ,  $R_2$ ,  $R_3$ ,  $U_1$ ,  $U_2$  and  $U_3$ . The region between  $R_1$  and  $R_2$  is considered to be a standard quality region, to which the QoS is strictly tied. After  $R_2$ , the increase in the data rate has a slower effect on the achieved utility. We assume that  $R_3$  is the saturation point, namely further increasing the achievable rate above this value would not impact the QoS. The average data rate of a client vehicle is scaled with the delay requirement of the service, (i.e.,  $W_s$ ).

In the network site, the service attributes are coupled with a utility value which is proportional to the average data rate,  $R_k$  (bps/Hz). The average data rate is calculated by:

$$R_k = \sum_{i=n-W_s}^n x_k[i] r_k[i] / W_s. \quad (1)$$

In this equation, continuous variable  $x_k[i] \in [0, 1]$ , is a decision variable which represents the assigned wireless resource to a client vehicle  $k$  at time slot  $i$ .  $r_k[i]$  (bps) is the achievable transmission data rate of client vehicle  $k$  at time slot  $i$ , which is calculated by:

$$r_k[i] = B \log_2(1 + SINR_k[i]). \quad (2)$$

$B$  (Hz) represents the all available network bandwidth. The actual data rate of client vehicle  $k$  at time slot  $i$  is then given by  $x_k[i]r_k[i]$ .

### C. VFogSim Simulation

The vehicular traffic traces and channel simulation data as well as the service attributes (e.g., application profiles) are inputted to VFogSim.

**Task Generation.** The computing tasks can be classified based on their degree of demand for computation resources. Such resource demand is determined by average task size,  $C_{\text{avg}}$  (kB/task), and computation intensities,  $\gamma$  (cycles/kB). In our simulation platform, we consider each task as a basic unit for offloading with an average demand size  $C_{\text{avg}}$ . The computation intensity represents how many CPU cycles are required to process one-bit input data for a task mainly depending on the nature of the applications. This yields the mean number of CPU cycles per processed task,  $D$  (cycles/task), i.e.,  $D = C_{\text{avg}} \cdot \gamma$ . We use four exemplary services in Table III to model the service heterogeneity.

**Network Simulations.** On the network side, the service demand per cell is pooled. We assume that if the client vehicles cannot react to a certain threshold in their transmission data rate, they failed to transmit their tasks, and they need to re-transmit their requests in the next TTI. The vehicles that reach this threshold are considered to be active. We design the spectral resource allocation and the resource management of fog nodes as two separate problems.

**Network Scheduler.** The objective of the network scheduler is to maximize the total transmission data rate overall client vehicles at each TTI. The different slopes in the piece-wise linear utility function reflect the service priorities in the max-rate scheduler. More specifically, the scheduler would give the resources to the services, which can create the highest utility increase with a unit resource. Consequently, the applied utility function also determines the inter-service priorities. In addition to this inter-service priority, due to the definition of  $R_k$  in equation (1), the achievable rates of the client vehicles also affect the achieved utility and the network resource allocation. We focus on a simple scheduler at the spectrum side formulated in (3a)-(3b):

$$\max_{x_k[n]} \sum_{k \in K} U_k \quad (3a)$$

$$\text{s.t. } \sum_{k \in K} x_k[n] \leq 1 \quad (3b)$$

The objective function in equation (3a) maximizes the total achieved utility over all the vehicles,  $K$ . We calculate the achieved utility based on the piece-wise linear utility function and  $R_k$  in equation (1). The utility function in Fig. 3 is linearized using standardized methods. The constraint in equation (3b) limits the assigned spectral resources to the maximum available resources.

**Computation Scheduler.** We assume that the client vehicles are associated with the co-located fog node of the gNB to which they are connected. Therefore, in this work, we focus on fog node selection from a set of available ones rather than

using an explicit task allocation algorithm and consider the following task scheduling problem in (4a)-(4c):

$$\max_{q_k} \sum_{k \in K} p_k D_k \log^{-1} t_{\text{remaining}} q_k \quad (4a)$$

$$\text{s.t. } q_k \in \{0, 1\}, \quad (4b)$$

$$\sum_{k \in K} q_k D_k \leq AR_c \quad \forall c \in C. \quad (4c)$$

The objective function (4a) models the inter-service prioritization, based on the price of a service,  $p_k$ , the total demanded resources of service  $k$ ,  $D_k$ , and the remaining time to finish the execution of the task,  $t_{\text{remaining}}$ . The remaining time calculated based on the delay expectations  $W_s$  progressively decreases during the simulation.

The scheduler gives a higher priority to the services with lower remaining execution times, ensures the continuous execution of the services, and creates possibilities to interrupt the service execution if a higher priority service request is received (e.g., in case of emergency). As a design metric, the service requests with higher resource demand are prioritized over the smaller ones to minimize fragmentation. The binary variable  $q_k$  set in (4b) presents whether the client vehicle  $k$  is chosen for execution or not (i.e., 1 or 0). Finally, (4c) ensures that the total demand is not exceeding the available computing resource at the  $c$ -th cell in a set of cells,  $C$ , (i.e.,  $AR_c$ ).

The task scheduling algorithm in (4a)-(4c) is run at every TTI. This way, the computation scheduler can dynamically update the resource usage based on active services at each TTI. The current version of VFogSim supports only 5G vehicle-to-network (V2N) [30]. It can be extended to support vehicle-to-vehicle (V2V) in the future.

### D. Accuracy and Scalability

In this section, we evaluate how close the wireless network simulation results are to the real-world measurements. The evaluation includes two parts, namely accuracy and scalability.

**Experimental setup.** To test the accuracy of our network simulations, we set up a driving test with a Xiaomi Mi 10T Pro 5G phone [31] placed on a car. The phone is equipped with an Elisa commercial subscriber identification module (SIM) card with 100 Mbps 5G mobile broadband for business [32]. With regards to real-world performance measurement, we collect two metrics: i) cell ID representing which cell the vehicle is connected to, ii) data rate (i.e., throughput), and the data rate values are averaged over 10 repeated measurements. The above metrics are associated with the GPS coordinates of the vehicle, which are collected every second over a sample duration of 10 minutes. During the sample collection, the car drives two cycles at an average speed of 18.57 km/h on the test trail shown in Fig. 4 with a speed limit of 30 km/h. On a commercial network, determining the active number of client vehicles and their demand is not trivial from the client side. Therefore, we used the measured data rate during our experiment to determine the number of clients. In particular, we tried various numbers of client vehicles in



Fig. 4: Measurement setup in Otaniemi, a university campus in Helsinki. The white car used for the measurement (on the top), and the test trail (at the bottom).

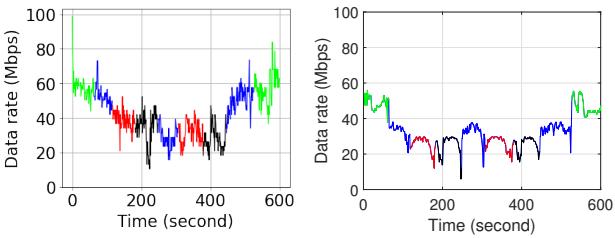


Fig. 5: Comparing the real-world measurements (on the left) and simulation results (on the right) in terms of data rate (i.e., throughput). The four cellular base stations which appear on the bottom of Fig. 4, with cell IDs #1 (red), #2 (black), #3 (green), and #4 (blue) have 8, 8, 4, and 6 client vehicles, respectively.

our simulations and selected the client number that minimizes the distance between the simulated result and the real-world measurement. For simplicity, we assumed that through the simulation horizon (i.e., 10 minutes in this case), the number of active client vehicles per cell and the active background services is constant, while the client vehicle's connected cell ID and the SINR value are changing in real-time.

**Simulator accuracy.** Fig. 5 shows the comparison between the measured throughput values in the real-world setup on the left and the simulator on the right. The actual throughput of a client vehicle depends not only on the SINR values but also on the scheduling dynamics such as the number of active client vehicles, the service mix at the given cell, or the scheduling policy. This dynamic nature of the resource scheduling causes the high fluctuations observed in real-world measurements. Fig. 5 shows that our network simulator performs close to the real-world measurements. In particular, both the simulation results and the real-world measurements have similar peak and minimum data rates. Although our assumption on the static number of client vehicles and their active applications in our simulation causes a more stable characteristic in Fig. 5, the

result shows a similar trend as the real-world measurements. Therefore, the network simulator can provide a realistic estimation of the data rates.

**Simulator scalability.** From the execution time perspective, as the number of client vehicles increases from  $|K| = 50$  to  $|K| = 100$ , we measure approximately 64% computation time increase, from 89.205 seconds to 139.705 seconds, in a commercially available computer with an I7 processor and 16 GB random access memory (RAM).

## V. CASE STUDY

We evaluate the functionality of VFogSim by demonstrating how to use it for analyzing the feasibility of different VFC deployment plans from a techno-economic perspective. In our analysis, we measure the data rate, delay, and cost of different deployment options. Moreover, we evaluate the impacts of inter-service prioritization, traffic loads, and pricing on the capacity plan, which can be used either in the default mode presented this section or substituted by the user-defined ones.

**Experimental setup.** In our simulations, we consider the  $1 \text{ km}^2$  area in downtown Helsinki city as shown in Fig. 6. To generate the SINR values using WinProp, we use a digital map of the area and locations of cellular cells from OpenStreetMap. Using SUMO, we generate the vehicular traces through this region for various numbers of client vehicles (i.e.,  $|K|$ ). Using the GPS coordinates of the client vehicles and the generated SINR map, we determine the SINR values of client vehicles at each TTI. This data is inputted into the task generation block of our simulator. To test the functionality of our simulator, we consider two task generation cases, i.e., high and low task generation. During the high task generation, the client vehicles generate a new task at each TTI. Therefore, this case is considered to be a worst-case scenario on the network. On the other hand, during the low task generation case, the client vehicles generate a task during a fraction of the simulation duration (i.e., 80% during the simulation), which is randomly distributed to the simulation horizon. In this paper, we consider a relatively short time horizon (i.e., in the order of hours), therefore long-term aspects such as the number of VFNs, communication technology, and network or edge capacity are considered to be constant. However, it is possible to test different communication technologies, changing bus routes, or capacity expansion strategies.

To map the service diversity in 5G networks, we model the four services with different QoS expectations, inter-service priorities, and prices. The different services are modeled by changing the  $R_1$ ,  $R_2$ ,  $R_3$ ,  $U_1$ ,  $U_2$ ,  $U_3$ ,  $C_{\text{avg}}$ ,  $W_s$  and  $p$  parameters.

- **Background (A<sub>1</sub>)**: This service runs in the background and requires relatively low data rate (e.g., connectivity service). It does not have any computational demand.
- **Object detection (A<sub>2</sub>)**: This service is latency-sensitive and rate-sensitive (i.e., non-elastic). We implement it through YOLOv5s [33] trained on COCO [34] dataset.
- **Lane detection (A<sub>3</sub>)**: This service is latency-sensitive and rate-demanding (i.e., non-elastic). We implement it through OpenCV [35] in a Python environment.

- **Video transcoding (A<sub>4</sub>):** This service have relatively soft latency and rate constraints (i.e., elastic). We implement it through HandBrake video transcoder [36] with x265 video encoder and mp4 container.

We profile the CPU resource demands of each service using an Intel Core i7-7700K 8-thread CPU with 4.2GHz frequency and detailed in Table III. For this scenario, we assume the network operator only supports one service.

Finally, we compare the following four deployment options based on the simulation results, i.e., estimated network connection and acceptance ratio.

- **NB Scenario:** In order to provide a base-level performance, we consider the scenario where the CFNs are used to serve all the demand.
- **WB Scenario:** To present our hybrid deployment solution, we consider the scenario where the VFNs are used to complement the CFNs.
- **InNB Scenario:** As an alternative to the WB scenario, we consider the case where the provider chooses to double the available CFNs instead of investing in VFNs.
- **B Scenario:** To provide another base-level performance, we consider the case no CFNs are deployed and all the computational tasks rely on VFNs.

The different scenarios are modeled by changing the available computing capacity at the buses and the base stations. To simulate NB and INB scenarios, we set the computing capacity of the buses equal to zero, whereas, for the B scenario, we set the CFN capacity equal to zero. For the WB, we consider both the CFN and VFN, therefore both computing capacities are non-zero. Due to the V2N scenario, from the network scheduler perspective, there are no differences among these scenarios. The VFogSim platform stores the resource allocations per user, the network data rates, the experienced delays per user (i.e., migration delay and queuing delay), and the total cost. In the remainder of this section, we detail how these results can be utilized to analyze the techno-economic dynamics.

**Average delay.** Fig. 7 presents the average delay experienced per client vehicle under two different deployment scenarios (i.e., NB and WB). We investigate two major components of delay, namely queuing delays and migration delays. Fig. 7

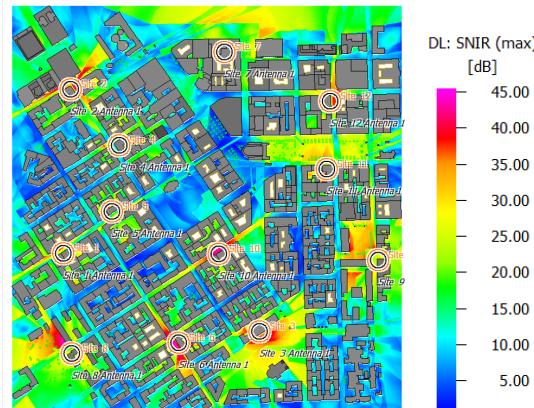


Fig. 6: The SINR map generated in WinProp.

TABLE IV: Estimated network connection ( $N \cdot C$ ) and acceptance ratio ( $A \cdot R$ ) for four different scenarios for  $|K| = 100$ .

Scenario	$N \cdot C$ (%)	$A \cdot R$ (%)
NB	99.97	82.15
WB	99.97	91.22
InNB	99.97	98.45
B	99.97	30.92

shows that in a high mobility environment, the main challenge is the migration delays, which cause approximately 70% of the measured delay. Deployment of the VFN affects mainly the queuing delays (i.e., up to 10% decrease) due to our task allocation strategy. More specifically, we do not reserve or direct any VFN resources to explicitly handle mobility challenges, but instead, we are demonstrating flexible capacity management with this periodically available computation capacity.

Fig. 8 demonstrates the congestion levels of base stations under two different deployment strategies. Here we measure the TTIs where the base station has at least one queued client vehicle over the complete simulation horizon. As envisioned, the simulation result demonstrates that the deployment of VFNs would create relaxation on the stationary fog nodes. However, unlike the InNB scenario, the relaxation in WB is affecting a subset of cells (e.g., cells 1, 4, 5, 8, 11 and 12 in Fig. 8). Therefore, the VFNs need to be chosen according to the objective to achieve the maximum impact.

**Techno-economic aspects.** The applicability of any technical solution in a real-world scenario depends not only on the QoS achievements but also on the economical implications posed by the solution. VFogSim provides the necessary statistics to perform a techno-economic analysis. We present the effects of various deployment options on the network connection and acceptance ratios in Table IV. In this scenario, we assume that the operator has sufficient network resources to guarantee spectral availability for all the client vehicles. Therefore, the application of VFNs does not impact the network connections. Consequently, we observe a connection of 99.97%, indicating that almost all the vehicles are connected to the cells at each time slot.

In terms of the acceptance ratio, we observe that the application of a hybrid platform can increase the efficiency by approximately 11%, while the InNB scenario provides the highest efficiency. On one hand, considering the doubled resources, the observed increase in performance can be ex-

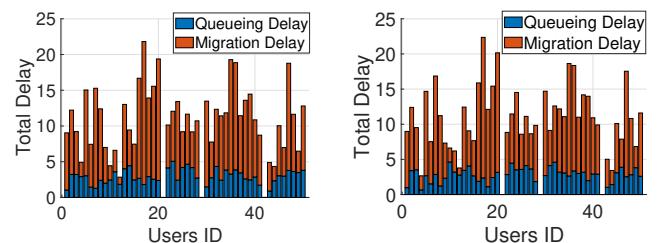


Fig. 7: The estimated total delay for deployment options NB (on the left) and WB (on the right) for  $|K| = 50$  and high task generation rate.

plained. On the other hand, the applicability of this strategy depends on the additional economical pressure it would create. We assume that the additional cost of deploying fog nodes and the respective operational costs are reflected in a proportional increase in the service price to the end client vehicles. We analyze the client vehicles' acceptance probability of the received service quality for the given price following the relation from [7]:

$$\left(\frac{U_{s1}}{U_{s2}}\right)^\mu \leq \left(\frac{P_{s1}}{P_{s2}}\right)^\epsilon \quad (4)$$

where the parameter  $\mu$  and  $\epsilon$  represent the sensitivities to the utility and the price, respectively, and we assume that they are equal. The left-hand side measures the increase ratio in the utility while the right-hand side measures the price increase by substituting an existing strategy (e.g., NB) with one with higher capacity. Considering the InNB scenario, doubling the existing resources also doubles the total price while the increment on the measured utility is relatively too low. Therefore, the inequality in equation (4) does not hold. However, when we analyze the hybrid computing solution (i.e., WB), as long as we ensure that the price of VFC is equal to or less than 12% of the standalone solutions, the client vehicles would accept the changes in the prices.

**Inter-service prioritization.** As detailed in the earlier sections, we map four different service types in our evaluations. In this part, we consider two task generation rates, namely low-rate and high-rate.

The background traffic is considered to have no computational requirements and is modeled as a live signal of the vehicles. Therefore, this service type has relatively low data rate expectations and strict latency requirements. The background acceptance ratios reported in Table V and Table VI indicate the connection status of the vehicles. Considering that this service has very high priority in the network scheduler, and it is well-covered in the urban area, more than 99% of the vehicles are always connected to a cell.

The remaining three service types are prioritized based on the previously detailed aspects. Among them, object detection and lane detection service types have higher priority than video transcoding services. This inter-prioritization is visible in the acceptance ratios. Both for WB and InNB scenarios, the prioritized services (i.e., lane detection and object detection) have an increase in their overall acceptance ratio between 10–20%, while the non-prioritized services (i.e., video transcoding)

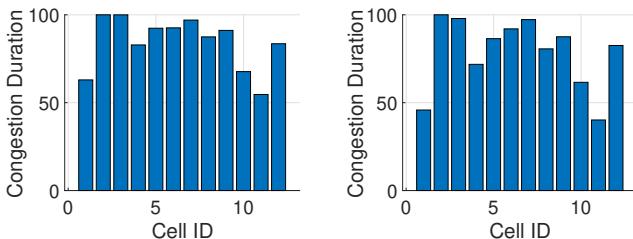


Fig. 8: The estimated cell congestion duration for deployment options NB (on the left) and WB (on the right) for  $|K| = 50$  and high task generation rate.

TABLE V: Estimated acceptance ratios for different services for four different scenarios for  $|K| = 100$  and high task generation rate.

Scenario	A <sub>1</sub> (%)	A <sub>2</sub> (%)	A <sub>3</sub> (%)	A <sub>4</sub> (%)
NB	99.64	71.7	72.76	38.37
WB	99.64	83.12	84.61	40.78
InNB	99.64	95.7	96.9	43.07
B	99.64	27.45	27.45	11.67

TABLE VI: Estimated acceptance ratios for different services for four different scenarios for  $|K| = 100$  and low task generation rate.

Scenario	A <sub>1</sub> (%)	A <sub>2</sub> (%)	A <sub>3</sub> (%)	A <sub>4</sub> (%)
NB	99.86	80.88	82.76	53.58
WB	99.86	88.9	90.12	54.78
InNB	99.86	98.95	99.33	56.52
B	99.86	32.19	32.7	19.29

report a negligible increase in their overall acceptance between 3 – 5%.

The acceptance ratio difference between the NB scenario and WB scenario indicates that the VFC mainly impacts the prioritized services. Note that this result is due to our deployed scheduler in (3a)-(3b). As detailed in the previous sections, it is possible to test different scheduling and task allocation models that can change the results.

**Impacts of traffic load.** In this part, we analyze the implications of increasing the number of vehicles on the service acceptance rate. To reflect the worst-case scenario, we assume the high task generation rate, and the results are presented in Table VII. First of all, the network connection of the vehicles, i.e., the background service, decreases to 97.27% for  $|K| = 150$ . The loss of approximately 2% is due to the coverage holes in the considered region and shows that the use cases that require high-reliability demand new network solutions. Regarding the considered deployment scenarios (i.e., NB, WB, InNB, and B), we can observe that for a few numbers of client vehicles the performance of WB and NB are comparable with each other. Therefore, when the traffic volume is low, the deployment of VFN provides a more flexible yet efficient network solution.

Although all the deployment options present a decrease in acceptance ratio when the traffic load increases, we can see that the standalone VFN deployments (i.e., deployment case B), present a more than 25% acceptance rate. This result indicates that although this standalone deployment option is not feasible for urban scenarios, it has the potential to be applied in rural areas.

**Impacts of pricing.** Finally, we analyze the implications of different prices for different services. In our task scheduling algorithm, the price of a service proportionally increases the priority of the service. In this part, we demonstrate the impact of different pricing options on resource allocations. In particular, we assume that the service provider charges higher for elastic services and cheaper for non-elastic services. Therefore, despite the under-prioritization of video transcoding services, the service provider would have a higher incentive to serve this client vehicle service. To analyze this scenario,

TABLE VII: Estimated acceptance ratios for different services for four different scenarios and number of vehicles,  $|K|$ .

Scenario	$ K $	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>
NB	50	99.60	87.31	88.43	69.38
NB	100	99.64	71.7	72.76	38.37
NB	150	97.27	64.5	63.43	29.29
WB	50	99.6	90.97	91.9	70.24
WB	100	99.64	83.12	84.61	40.78
WB	150	97.27	75.49	76.25	31.53
InNB	50	99.60	99.58	99.76	71.57
InNB	100	99.64	95.7	96.9	43.07
InNB	150	97.27	89.31	91.22	34.48
B	50	99.60	32.23	32.47	25.83
B	100	99.64	27.45	27.45	11.67
B	150	97.27	25.58	25.16	9.66

TABLE VIII: Estimated acceptance ratios for different services for four different scenarios for  $|K| = 100$  and high task generation rate with price differentiation.

Scenario	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>
NB	99.64	49.87	74.92	43.33
WB	99.64	69.31	86.76	43.47
InNB	99.64	90.63	99.09	43.36
B	99.64	17.47	28.6	13.96

we assume that the prices of video transcoding services are 10 Monetary Units (MU), while the object detection and lane detection services are prices of 2 MU and 5 MU respectively. Because the background services are modeled as live signals, we assume that they are priced the least, i.e., 1 MU. Table VIII presents the measured acceptance ratios for different deployment options.

Comparing Table V and Table VIII, we can see that the background services are not affected by the lower price. This is due to the fact that the prioritization mechanism in the network scheduler is not directly affected by the service price. However, we can observe that this new pricing strategy mainly affects the object detection services, which have the second least price. On the other hand, in terms of the lane detection services, we do not observe any major drop even with the relatively low price. This is because the prioritization mechanism we use relies on a multitude of aspects apart from the service prices. With this new pricing strategy, the video transcoding service gains a higher acceptance rate, showing that the additional incentive produced by the higher price is effective in decision-making.

Fig. 9 visually presents the impact of pricing strategy on the acceptance ratio of each service type, where the risk of unstable pricing can be observed. More specifically, the different prices for services cause an approximately 10% decrease in object detection services, while the other two services report a less than 5% increase. This result demonstrates that, even though the pricing strategy can affect the service-based acceptance rates, the price should reflect the supply and demand aspects of a particular service.

## VI. DISCUSSION

In this section, we summarize the key characteristics of our simulation platform and detail our future directions.

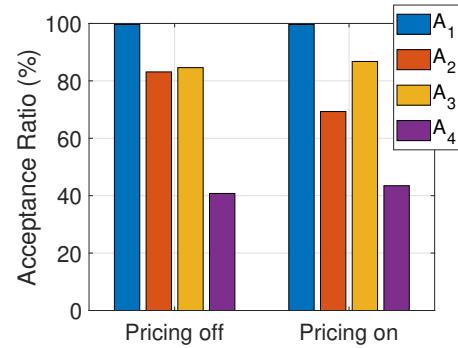


Fig. 9: The acceptance ratio's per service type for different pricing strategies for WB deployment scenario and  $|K| = 50$ .

**Functionality.** The results in Section V demonstrate that VFogSim provides the necessary functionality to model and test various deployment options. Furthermore, the simulator output provides both the perceived quality and the total cost to the end-user, which allows us to conduct a deep analysis of the techno-economic implications of the hybrid fog computing environment.

**Input Flexibility.** VFogSim can use either synthetic data or real-world measurements as inputs (cf. Fig. 2). This versatile input option enables testing of a large variety of use cases, including real-world implementations and future potential deployment options.

**Modularity.** VFogSim is designed and developed in a modular structure to allow customization. It is possible to integrate different schedulers, prioritization mechanisms, or pre-trained AI models (see in Section V). Since the task scheduler and network scheduler are separated in our model, they can be customized individually based on the user-defined scenario.

**Extendability.** VFogSim is extendable to respond to the evolution of the network and edge computing technologies in the intelligent transportation system (ITS). Although we performed our analysis for 5G, it is possible to extend it for 6G and beyond networks [37]. For example, we can change base station characteristics by substituting the network configuration file in WinProp and adjusting the traffic behavior by adding more autonomous vehicles in SUMO [38]. Moreover, the simulator can work with different resource schedulers (see further details in Appendix).

**Scalability.** VFogSim can accommodate more realistic system scenarios with a larger set of client vehicles, VFNs, and base stations. The time complexity of the overall simulator depends on the used scheduling algorithms and the simulation scenario. We demonstrated the increase in the time complexity with the number of client vehicles in Section IV.

**Future Directions.** The modular design of VFogSim simplifies the integration of new features into our simulator. We are extending the current version to support the multi-tenancy and network slicing as proposed by 3GPP [28]. Finally, to increase our range of scenarios, we will integrate Vehicle-to-vehicle (V2V) communication into future versions.

## VII. CONCLUSION

This work presents VFogSim, a modular and a customizable simulation platform to test and evaluate techno-economic implications of different VFC deployment options and use cases. It supports the mobility of fog nodes and contains realistic modeling of vehicle traffic, resource consumption, and wireless communications. We evaluate the accuracy and scalability of our platform by comparing the network simulation results with the real-world measurement in Otaniemi. Furthermore, we demonstrate the functionality of the simulator by using it for evaluating the technical and financial feasibility of different VFC capacity plans for an urban area in Helsinki. In practice, our simulator can also be used for testing customized resource and task allocation strategies, and prioritization mechanisms in diverse VFC environments and application scenarios. In this version, we support V2N communication but if needed it can be extended to support V2V communication.

## ACKNOWLEDGMENT

This project has received funding from Academy of Finland under grant number 317432 and 318937.

## REFERENCES

- [1] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.
- [2] C. Zhu, G. Pastor, Y. Xiao, and A. Ylajaaski, "Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 58–63, 2018.
- [3] M. Noreikis, Y. Xiao, and A. Ylä-Jääski, "Qos-oriented capacity planning for edge computing," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [4] W. Mao, O. U. Akgul, A. Mehrabi, B. Cho, Y. Xiao, and A. Ylä-Jääski, "Data-driven capacity planning for vehicular fog computing," *IEEE Internet of Things Journal*, pp. 1–1, 2022.
- [5] C. Zhu, J. Tao, G. Pastor, Y. Xiao, Y. Ji, Q. Zhou, Y. Li, and A. Ylä-Jääski, "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4150–4161, 2019.
- [6] B. Cho and Y. Xiao, "Learning-based decentralized offloading decision making in an adversarial environment," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11 308–11 323, 2021.
- [7] O. U. Akgul, I. Malanchini, and A. Capone, "Dynamic resource trading in sliced mobile networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 220–233, 2019.
- [8] H. Gupta, A. Dastjerdi, S. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, 06 2016.
- [9] X. Zeng, S. K. Garg, P. Strazdins, P. P. Jayaraman, D. Georgakopoulos, and R. Ranjan, "Iotsim: A simulator for analysing iot applications," *Journal of Systems Architecture*, vol. 72, pp. 93–107, 2017, design Automation for Embedded Ubiquitous Computing Systems. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1383762116300662>
- [10] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 39–44.
- [11] T. Qayyum, A. W. Malik, M. A. Khan Khattak, O. Khalid, and S. U. Khan, "Fognetsim++: A toolkit for modeling and simulation of distributed fog environment," *IEEE Access*, vol. 6, pp. 63 570–63 583, 2018.
- [12] R. Mayer, L. Graser, H. Gupta, E. Saurez, and U. Ramachandran, "Emu-fog: Extensible and scalable emulation of large-scale fog computing infrastructures," in *2017 IEEE Fog World Congress (FWC)*, 2017, pp. 1–6.
- [13] A. A. Coutinho, F. Greve, C. Prazeres, and J. Cardoso, "Fogbed: A rapid-prototyping emulation environment for fog computing," 05 2018, pp. 1–7.
- [14] A. Brogi and S. Forti, "Qos-aware deployment of iot applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, 2017.
- [15] A. Brogi, S. Forti, and A. Ibrahim, "How to best deploy your fog applications, probably," 05 2017.
- [16] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.
- [17] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: <https://elib.dlr.de/124092/>
- [18] R. Hoppe, G. Wölflé, and U. Jakobus, "Wave propagation and radio network planning software winprop added to the electromagnetic solver package feko," in *2017 International Applied Computational Electromagnetics Society Symposium - Italy (ACES)*, 2017, pp. 1–2.
- [19] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 6–9.
- [20] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [21] 3GPP 22.886 Release 15, "Study on enhancement of 3GPP support for 5G V2X services," 2018.
- [22] R. Calheiros, R. Ranjan, C. De Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," 04 2009.
- [23] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," 01 2008, p. 60.
- [24] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. New York, NY, USA: Association for Computing Machinery, 2010.
- [25] P. Wette, M. Draxler, A. Schwabe, F. Wallaschek, M. Zahraee, and H. Karl, "Maxinet: Distributed emulation of software-defined networks," 06 2014, pp. 1–9.
- [26] A. Virdis, G. Stea, and G. Nardini, "Simulte - a modular system-level simulator for lte/lte-a networks based on omnet++," in *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*, 2014, pp. 59–70.
- [27] OpenStreetMap contributors, "Planet dump retrieved from <https://planet.osm.org/>," 2017.
- [28] ETSI GS ZSM 003 V0.22.0, "Zero-touch Network and Service Management (ZSM); End to end management and orchestration of network slicing," 2020.
- [29] B. Nelms, L. Waldron, L. Barrera, and et al., "Cellmapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types," 2016.
- [30] G. Naik, B. Choudhury, and J. Park, "Ieee 802.11bd 5g nr v2x: Evolution of radio access technologies for v2x communications," *IEEE Access*, vol. 7, pp. 70 169–70 184, 2019.
- [31] "Mi 10t pro 5g," 2022, last accessed 11 January 2022. [Online]. Available: <https://www.mi.com/global/mi-10t-pro/specs/>
- [32] "Elisa 5g mobile broadband for businesses," 2022, last accessed 11 January 2022. [Online]. Available: <https://yrityksille.elisa.fi/mobiililaajakaista>
- [33] G. Jocher, A. Stoken, J. Borovec, and etc., "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4154370>
- [34] "Coco - common objects in context," 2021, last accessed 31 August 2021. [Online]. Available: <https://cocodataset.org/>
- [35] "Opencv," 2021, last accessed 31 August 2021. [Online]. Available: <https://opencv.org/>
- [36] "Handbrake," 2021, last accessed 31 August 2021. [Online]. Available: <https://handbrake.fr/>
- [37] X. You, C. Wang, J. Huang, and etc., "Towards 6g wireless communication networks: vision, enabling technologies, and new paradigm shifts," *Science in China, Series F: Information Sciences*, vol. 64, no. 1, Jan 2020.
- [38] M. Guérin and I. Dusparic, "Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.

**Ozgur Umut Akgul** is currently a Postdoctoral researcher with the Department of Communications and Networking, Aalto University. He received B.S. in electronics engineering and electrical engineering and M.S. in computer engineering from Istanbul Technical University, Turkey, in 2011 and 2014, respectively, and a Ph.D. in information technology from Politecnico di Milano in 2019. His research interests focus on optimization models, mathematical programming, and machine learning, with the application of these techniques to wireless network problems such as wireless resource allocation, edge computing, anticipatory network optimization, infrastructure and resource sharing, and network slicing.

**Wencan Mao** Wencan Mao is currently pursuing the Ph.D. degree in the Department of Communications and Networking, Aalto University, Espoo, Finland. She received the B.E. degree in vehicle engineering from Wuhan University of Technology, Wuhan, China in 2017, and the M.S. degree in mechanical engineering from Aalto University, Espoo, Finland in 2019. Her current research interests include edge computing, Internet of Things, vehicular networking, resource allocation and capacity planning.

**Byungjin Cho** received the doctoral degree in communications engineering from Aalto University, in 2016. He is currently a postdoctoral researcher with the Department of Communications and Networking, Aalto University. His research interests include resource managements in networked systems using algorithmic decision theory.

**Yu Xiao** Yu Xiao received the doctoral degree in computer science from Aalto University, in 2012. She is currently an assistant professor with the Department of Communications and Networking, Aalto University. Her current research interests include edge computing, wearable sensing and extended reality. She is a member of the IEEE.

## APPENDIX

### *Applied WinProp and SUMO Parameters*

Parameter	Value
Air interface	5G NR n78
Multiple access	OFDM/OFDMA
Duplex separation	TDD
Frequency band	3500MHz
Channel bandwidth	100MHz
Minimum required SINR	-10dB
Number of base stations	12
Height of base stations	30m
Transmission power	46dB
Propagation model	Dominant path model
Simulation area	1km × 1km
Simulation granularity	1m × 1m
Simulation horizon	200s
Simulation time slot	1s

### *Customizable Functions*

Function	Parameters	Functionality
preprocess()	datasource, source	Processes the cvs file from 3rd party programs and returns the user data
User_driven_event()	Event Rate, User Matrix	Generates the requests based on application profiles
Scheduler()	Requests, Cell and VFN information, Application Profiles	Main scheduler function
Default_CellUpdate()	User Matrix, Resource allocations	Updates the user and cellular information and determines the blocked or rejected applications
Default_Scheduler()	SINR values, Achieved rate Limit, User List, Application Profiles, Utility function parameters	Performs the network resource scheduling
Default_EdgeScheduler()	VFN Capacity, Active services, Application Profiles, Bus Schedules	Performs the computational resource scheduling

### *Input Data*

Data Name	Location	Definiton
Input.csv	Data	The user data obtained from SUMO and Winprop
bus.csv	Data	VFN locations per TTI obtained from SUMO and Winprop

### *Input Parameters*

Data Name	Definition
EventRate	Manages the density of request arrival
TimeFilter	Limits the duration of the simulation
UserFilter	Limits the maximum number of users
ServiceRequirements	Contains the achieved rate requirements per service
ServiceUtilities	Contains the utility values per service type
appData	Contains the computational requirements of a service