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 / 11-01-2022

CITATION

Akgul, Ozgur Umut; Mao, Wencan; Cho, Byungjin; Xiao, Yu (2022): A Data-driven Platform for Simulating Vehicular Fog Computing Environment. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.17829398.v1>

DOI

[10.36227/techrxiv.17829398.v1](https://doi.org/10.36227/techrxiv.17829398.v1)

A Data-driven Platform for Simulating Vehicular Fog Computing Environment

Ozgur Umut Akgul, Wencan Mao, Byungjin Cho, and Yu Xiao
Department of Communication and Networking
Aalto University, Finland
email: {firstname.surname}@aalto.fi

Abstract

Edge/fog computing is a key enabling technology in 5G and beyond for fulfilling the tight latency requirements of compute-intensive vehicular applications, such as cooperative driving. Concerning the spatio-temporal variation in the vehicular traffic flows and the demand for edge computing capacity generated by connected vehicles, vehicular fog computing (VFC) has been proposed as a cost-efficient deployment model that complements stationary fog nodes with mobile ones carried by moving vehicles. Accessing the feasibility and the applicability of such hybrid topology, and further planning and managing the networking and computing resources at the edge, require deep understanding of the spatio-temporal variations in the demand and the supply of edge computing capacity as well as the trade-offs between achievable Quality-of-Services and potential deployment and operating costs. To meet such requirements, we propose in this paper an open platform for simulating the VFC environment and for evaluating the performance and cost efficiency of capacity planning and resource allocation strategies under diverse physical conditions and business strategies. Compared with the existing edge/fog computing simulators, our platform supports the mobility of fog nodes and provides a realistic modeling of vehicular networking with the 5G and beyond network in the urban environment. We demonstrate the functionality of the platform using city-scale VFC capacity planning as example. The simulation results provide insights on the feasibility of different deployment strategies from both technical and financial perspectives.

1 Introduction

The emerging multi-access edge computing (MEC) and fog computing (FC) technologies share the objective of bringing cloud-like resources and services closer to the clients to reduce the response delay. One example is to

deploy vehicular applications, such as cooperative driving at the edge/fog computing nodes co-located with cellular base stations or road side units. In this case, one of the most significant concerns associated with the distributed architecture is the limited computing capacity and coverage area of any single edge/fog computing node. Such a constraint raises issues with economic viability and scalability of edge/fog computing services in satisfying the spatio-temporal variations in resource demand. Some computing nodes may be severely swamped by overwhelming demand in a particular region at a particular period of time. On the other hand, adding more capacity to the edge requires additional investment, and such static provisioning of computing capacity would result in a waste of resources if the demand does not always remain high.

Recently, vehicular fog computing (VFC) has been proposed as a new computing paradigm where fog nodes are installed also on moving vehicles to increase the spatio-temporal dynamics in supply of computing capacity [31]. We call these fog nodes as vehicular fog nodes (VFNs). VFNs are supposed to serve other vehicles within the range of one-hop communication [13, 15] to guarantee low transmission latency. With VFNs, the service migration between fog nodes occurs less often when client vehicles are travelling along VFNs [33]. This would lead to increased utility of computing resources, and further the decrease in deployment costs.

To provide reliable VFC services with low costs, a key requirement is to apply effective capacity planning and resource management strategies, i.e., where and how much capacity to deploy [22], whether and where to offload computing tasks [34], how to jointly schedule the radio and computing resources [1], and so on. These strategies must take into account the expected quality of service (QoS), the spatio-temporal variation in vehicle traffic, varying resource demand of vehicular applications, and the overall deployment and operating costs of fog nodes. While there have been many attempts to

implement new strategies toward greater intelligence and autonomy at the edge, one fundamental challenge is how to assess the practical performance of the planned system. This paper aims at solving this challenge by developing an open platform that provides realistic simulation of dynamic VFC environments and various application scenarios.

Compared with real-world experiment, simulation has the following advantages: i) less expensive than using a real test-bed, ii) repeatable and potentially scalable evaluation, and iii) timely and statistically rigorous manner with stochastic allowing for system performance and management of uncertainty across the heterogeneous environment. There exist already several simulators for dynamic edge computing environment, including iFogSim [12], EdgeCloudSim [26], and FogNetSim++ [24]. These simulators typically integrate networking, load generator, and resource management, and support mobility of clients. However, they still suffer from the lack of variants, subject to the simplifying assumptions that might not fully represent envisaged VFC environment. More specifically they do not support the mobility of fog nodes [12, 26], and do not consider the diversity in resource demand among applications [24] or the practical wireless channel characteristics [12, 24, 26]. As a consequence, the challenges brought by the mobility of fog nodes, such as availability and serviceability of VFC services, have not been considered. Moreover, the techno-economic implications of VFC in 5G and beyond networks have not been studied.

The platform proposed in this paper solves the above-mentioned issues by a data-driven approach where data reflecting real-world experiences are used to feed into the assessment of technical and economic potentials in VFC environment. To the best of our knowledge, it is the first data-driven VFC simulation platform that incorporates both stationary and mobile fog nodes, taking into account different building blocks that capture realistic environments in different physical features. The key contributions of this paper are summarized as follows.

- We develop a data-driven platform for simulating different VFC environments and application scenarios including the ones where stationary and mobile fog nodes (i.e. VFNs) coexist and collaborate to fulfill the spatio-temporally varying resource demand. The developed platform uses open-source data as input and is made open access to invite the research community for future developments.
- The platform provides realistic modeling of vehicle traffic in urban areas, resource consumption of different vehicular applications, and the related physical attributes of wireless networks, which is non-trivial to be addressed in a simulation platform.
- The functionality of the platform has been demonstrated through a case study on feasibility evaluation of different VFC deployment plans from both technical and financial perspectives. The platform follows the modular design principles. It can be also used for testing customized strategies of task offloading, resource allocation, service migration, and service prioritization.

The remainder of this paper is organized as follows. Section 2 gives an overview of the background and motivation. Section 3 presents the platform architecture, with more detailed description of each building block included in Section 4. Section 5 demonstrates our framework's potential through a multitude of experiments. Section 6 compares the related work with this one. Section 7 discusses the key features of the platform and future work. Section 8 concludes the work.

2 Background and Motivation

We start with the application scenarios of VFC, and then move to the discussion about the potential use cases and desired features of a VFC simulator.

2.1 Vehicular Fog Computing

Previous works have proved the spatio-temporal variations in vehicle traffic in urban environments. Obviously, purely relying on stationary fog nodes whose capacities are planned according to the peak demand [30] would highly likely cause over-provisioning. One solution is to utilize the mobility of fog nodes to provide more cost-efficient fog computing services for connected vehicles. As illustrated in Figure 1, in such a dynamic VFC environment, there are at least two types of fog nodes, including cellular fog nodes (CFNs) and VFNs. VFNs and CFNs collaborate to provide computing services for vehicles within their communication ranges. For example, in Figure 1, Vehicle B offloads its task to a VFN that is carried by a bus. When Vehicle B is about to move out of the communication range of the VFN, the task will be migrated to the CFN co-located with Vehicle B's connected base station.

2.2 VFC Simulator

The computing and networking resources need to be rationally planned and allocated in order to meet stringent time constraints requested by real-time vehicular applications, such as cooperative intersection crossing [7] and lane change scheduling [2]. When the numbers of vehicles and fog nodes increase, resource management becomes more challenging, which requires the aid of control and data plane to gather the statistics, schedule the

Figure 1: Exemplary scenario of VFC.

network and computational resources, and handle the user and task migration. Though many researchers have been working on these issues, there still lack an open platform that would allow them to easily assess the solutions in a simulated but realistic VFC environment.

This paper aims at developing such a simulator that can be used for evaluating at least different capacity planning and task allocation strategies.

Capacity planning. Capacity planning is focused on planning where to deploy the fog nodes and how much computing capacity to deploy on each fog node in order to fulfill estimated computing resource demand with better techno-economic performance. For example, [22] proposed a QoS-oriented capacity planning solution for edge computing. Our platform can be utilized to validate such solution under the VFC environment.

Task allocation. Task allocation is focused on deciding where to offload the tasks generated by the users to improve the QoS and achieve better techno-economic performance. For example, [34] proposed a latency and quality optimized task allocation solution in VFC environment. Our platform can be used to validate such solution in the 5G network.

To support the evaluation of capacity planning and task allocation, the VFC simulation platform is expected to provide at least the following features.

Vehicle traffic simulation. The simulator must provide realistic simulation of the spatio-temporal variations in vehicle traffic. This requires jointly considering the macroscopic and microscopic characteristics. The macroscopic characteristics include speed, density, and flow rate of traffic flows, while the microscopic characteristics include the dynamics of individual vehicles and the effects of junctions, intersections, and traffic lights.

Deployment of VFNs and CFNs. The simulator should allow deployment of CFNs according to the locations of cellular base stations, and that of VFNs according to the

routes of vehicles carrying VFNs which can be either stochastic or

Physical network simulation. The simulator should provide network simulation covering physical and upper layers. Depending on the location of vehicles, the simulator is supposed to estimate the signal-to-interference-plus-noise ratio (SINR) they can receive from the network provider. Such physical layer information can be utilized for planning network handovers and service migration between fog nodes.

Computing and networking resource management. The simulator should support the planning and scheduling of both computing and networking resources. This would allow users to test different algorithms for joint scheduling of computing and network resources.

In addition to these functional features, the platform is supposed to be extensible to support new features and to be scalable. In practice, the complexity mainly comes from two perspectives. One is the mobility of vehicles including both client vehicles and VFNs. The availability and serviceability of VFC depends on the relative locations of vehicles and different types of fog nodes. The other is the joint scheduling of computing and networking resources in a dynamic environment with spatio-temporal variations in both demand and supply.

3 System Architecture

The system level architecture of VFC simulator is presented in Fig. 2. The complete model consists of two major parts, i.e. demand and mobility simulation and network simulation. Furthermore, the demand and mobility simulation is separated into vehicular traffic and user attributes modules, and the network simulation is divided into communication channel and upper-layer network modules.

Vehicular traffic. The vehicular traffic module is imple-

Figure 2: Vehicular fog computing platform's system level architecture.

mented by SUMO [18]. It generates a realistic vehicular movement through the considered region, taking the 2D road network as the input. It offers a fine-grained microscopic traffic simulation, including the vehicular movement and the traffic infrastructure (e.g., traffic lights). The mobility data is fed to the user attribute block to generate the request.

User attributes. The user attributes module involves application profile and request generation. The application profile is aimed to describe the use of the hardware components, e.g., CPU/GPU/NPU computing units, battery, and memory, by a given application. In a simulation platform, not only does application profiling help estimate the resource consumption of an application but also leads to strategic solutions that require resource consumption estimates. The request generation is responsible for generating any kind of offloading service in certain time intervals. The user attributes module provides both above to the upper-layer network module.

Communication channel. The communication channel module is responsible for estimating the SINR values. In order to model the physical aspects, such as reflections and shadowing in the urban scenario, we use the 3D map of considered area as input to generate the urban database in WinProp [14]. The database contains the locations and heights of the buildings and vegetation objects (e.g. trees) and calculates the transmission loss caused by them. To simulate the sites and antennas in a realistic manner, we use the aggregated real-world data, including the information of the location, height, frequency, antenna type, and carrier parameters in the base stations. Finally, the propagation model and air interface configuration is jointly considered to estimate the SINR map of the region. The impact of vehicles and other time dependent objects on the SINR map is not considered.

Combining the SINR map with the vehicular traffic data, we calculate the achievable rates per vehicle and send them to the upper-layer network module.

Upper-layer network. On the upper-layer network module, both the spectral and computational resource allocations are performed as well as the necessary statistical investigations. gNB registration handles the subscription of the users to specific base station. This block is critical for the scenarios where the coverage area of the base station is dynamically managed in order to increase energy efficiency. As stated by ETSI [11], the gathered statistics enables higher efficiency in network orchestration and resource management solutions. In this work, we used this block for evaluation purposes. Billing is modeled especially for multi-tenancy scenarios and techno-economic analysis. The network scheduler and computation scheduler focus on the allocation of the spectral and computational resources. In this second phase, we assume that the priority of a service is determined by the requested resources, the price of the service and the remaining execution time of the service. The edge nodes are assumed to be connected to base station (gNB), therefore the tasks are executed either in the gNB or the VFN. Finally, user and task migration focus on offloading the users or the tasks to other cells. We assume that all the edges and VFNs has the necessary virtual machines (VMs). Therefore, user offloading would only cause migration delays.

In the simulator, first we generate the SINR map and the vehicular traces over the complete simulation horizon, N . The time horizon is discretized and divided into individual time slots, n . We assume that each user can only be associated with one active service at a time slot. The SINRs of user k is estimated per n based on the location of the vehicle and the SINR map. At every n , we

first generate the active tasks, then we perform the network scheduling per cell. If the user k reaches 80% of its expected achieved rate, we consider the service to be active and can be executed at the edge. Otherwise, the request is queued and executed in the upcoming time slots. Among the active services, we run the task scheduling algorithm and allocate the edge resources to the users. The user will be queued if it fails to receive the edge resources. We assume that the respective VMs are actively run at all the edges, and the user migration would cause a fixed amount of delay for all the services. In this work, the user associations with the base stations are done individually, thus the dynamic offloading strategies are not considered. Moreover, we only consider vehicle-to-network (V2N) mode in 5G V2X [20].

4 VFC Simulator Design

In this section, we detail the data-driven simulator design and the key attributes of the default model. The inputs of both vehicular traffic and communication channel modules are the 2D/3D map extracted from OpenStreetMap [23].

4.1 Modeling Vehicular Traffic

The 2D map is fed into SUMO [18] to generate the road network in the target area. The road network contains the information of the nodes (i.e. junctions) and the edges (i.e. road segments). The information of the nodes include their location, the associated edges, and the relationship of these edges. The information of the edges include their location, speed, priority level, number of lanes, road type (e.g. motorway, trunk, tertiary, residential), and the traffic regulations. Additionally, it contains the information of the traffic lights at each intersection and generates the corresponding time schedules. 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 rate. The generation of the trips takes the number of lanes into consideration so that the traffic density on the primary roads are generally higher than the secondary roads. Moreover, we use different number of vehicles to simulate various traffic conditions such as peak hours, off-peak hours, and extreme scenarios. In this work, we assume the VFNs are carried by buses, which have regular routes and time schedules. The routes of the buses are generated using Origin-Destination (OD) matrix. More specifically, we set the origins and destinations of all the buses, and generate their routes by the shortest path.

4.2 Modeling User Attributes

In general, the process of an application is divided into multiple but independently deployable services such as object detection and vehicle localization. When offloading a service instance, so-called a task, from a user to a nearby fog node, the decision of whether to offload the task is based on the prediction of resources consumed by the task. In addition to resource usage information, an application profile contains also task-specific constraints, e.g., data size, execution time, supported quality.

The offloading service user generates tasks and sends the request. Each user may have multiple types of service instances, each of which may follow some probability distribution with a certain mean value, e.g. periodic and exponentially distributed inter-task times, with a mean rate. One may feed different mean values for specific time intervals during a simulation time. For instance, the task generation rate can be increased or decreased in an hour-per-hour pattern or even in a minute-to-minute pattern depending on spatial and temporal features of user activities. In this work, each offloading service user is assumed to randomly choose a single task type at a given time, and to generate the task with an inter-arrival time proportion to the number of users in the vicinities, e.g., the number of task requesters within the communication coverage area of the same cell.

User attributes represent characteristics of the task of offloading service requests which can be divided into various resource requirement types based on their degree of demand for computation resource. Such resource demand is determined by average task size, C_{avg} (kB/task), and computation intensities, γ (cycles/kB). We assume that the computation resource demands of the users are modelled by the size and complexity of the task they holds. In general, computation tasks can be partitioned into sub-tasks at different granularity levels. The divisible multiple sub-tasks can be executed in parallel, series, or a mixture manner. Some of these sub-tasks must be performed locally and some of them can be both performed locally and offloaded to the external computation resources. In our simulation platform, each task is considered as a basic unit for offloading with an average demand size C_{avg} (kB/task). 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 bit, D (cycles/task), increasing with the task size or the computation intensity of the considered application, e.g., $D = C_{avg} \cdot \gamma$.

Various applications require different computing intensities which can be estimated from measurements by applying the methods, so-called application profiling, to analyze resource usage patterns of each application and

obtain resource requirements for each task category, described in [8, 10, 16, 21]. The demand takes the value from the application profiling with the input task for the next time slot. Once a task is generated and sent to an edge server, the edge server may make a pre-decision on resource management based on the existing application profile. The estimated values of the computation intensity of some test applications of practical interest are found to be 2339 cycles/bit for face recognition [16], 2640 cycles/bit for 400-frame video game [10], 8250 cycles/bit for seven-queenpuzzle [16], 36992 cycles/bit for virus scanning [8] and so on. Depending on the computing capability allocated to a fog node, f (cycles/sec), such profile values enable to obtain the execution time, $t_{\text{remaining}} = D/f$ (sec/task).

4.3 Modeling Communication Channel

The communication channel is simulated using Altair WinProp [14], and the urban database is converted from the 3D map. It contains the information of the urban environment such as the location and heights of the buildings and vegetation blocks. These information are used for estimating the transmission loss caused by reflections and shadowing in the propagation model. The sites and antennas are set according to the real-world data. For example, the locations of the base stations are extracted from the street addresses in CellMapper [3], and exported to Universal Transverse Mercator (UTM) coordinate system using geocoding. The air interface are set according to real-world configuration files. The users can customize various parameters such as the network type, multiple access, duplex type, frequency band, channel bandwidth, and transmission power. The network attributes, such as the cell coverage and SINR value are calculated for every pixel in the target area. Combing the SINR map with the vehicular traces, we can calculate the achievable rate of each vehicle at each time during the simulation.

4.4 Modeling Upper-layer Network

The physical attributes obtained from the vehicular traffic and channel simulation as well as the user attributes (e.g., application profiles, generated requests) are inputted to the upper-layer network simulation. In the network simulator, the user demand per cell is pooled and the resource allocations (both spectral and computational) are performed per transmission time interval (TTI). We design the spectral resource allocation and the edge resource management as two separate problems. We assume that in order for a service to become active, the user needs to have to achieve at least 80% of their maximum utility, indicating that the respective demand is

fully transmitted to the network. If the users fail to transmit their demands to the network, they will be queued to re-transmit their requests.

In the network site, the user attributes are coupled with a utility value which is proportional to the average achieved rate, R_k , where

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

The average achieved rate is scaled with the delay expectations, W_s of the services, therefore the delay is indirectly implemented in our model. We have considered a piece-wise linear utility function proposed in [1], cf. Fig. 3. This utility function is determined by six parameters, i.e., R_1 , R_2 , R_3 , U_1 , U_2 and U_3 , in order to assume a service is active. The region between R_1 and R_2 is considered to be standard quality region, to which the quality of experience (QoE) is strictly tied. After R_2 , the increase in the achieved rate has a decreased effect on the achieved utility. Finally, we assume that R_3 is the saturation point, namely further increasing achievable rate above this value would not impact the QoE. We used four exemplary services in [1] to model the service heterogeneity. The objective of the network scheduler is to maximize the total achieved rate over all users at each TTI. Therefore, the applied utility function also forces the inter-service priorities. More specifically, the slope of each utility function sets the initial priority of each service, i.e. the function with higher slope has higher initial priority. In addition to this initial slope, the achievable rates of the users affect the allocated resources at a given time slot, n . We focus on a simple scheduler at the spectrum side, as formulated in (2a)-(2b). Our objective function (2a) maximizes the total achieved utility over all the vehicles, K . The utility function in Fig. 3 is linearized using standardized methods. (2b) limits the assigned spectral resources to the maximum available resources.

We assume that the users are associated with the edge of the base station that they are connected to. Therefore, in this work, we did not used an explicit task allocation

Figure 3: Generic utility function from [1].

$$\max_{x_k[n]} \sum_{k \in K} U_k \quad (2a)$$

$$\sum_{k \in K} x_k[n] \leq 1, \quad (2b)$$

algorithm where we focus on the selection of edge nodes from a set of available edge devices. In this work we focus on the task scheduling problem, that we formulate in (3a)-(3c). The objective function (3a) models the inter-

$$\max_{w_k} \sum_{k \in K} p_k D_k \log^{-1} t_{\text{remaining}} w_k \quad (3a)$$

$$w_k \in \{0, 1\}, \quad (3b)$$

$$\sum_{k \in K} w_k D_k \leq AR_c \quad \forall c \in C. \quad (3c)$$

service prioritization, based on the price of a service, p_k , the total demanded resources, D_k , and the remaining time to finish the execution of the task, $t_{\text{remaining}}$. The scheduler gives higher priority to the services with lower execution times, ensuring the continuous execution of the services while allowing service interruption in case of an emergency. As a design metric, the larger requests are prioritized over the smaller ones, in order to minimize the fragmentation. The binary variable w_k is set in (3b) and presents whether the user k is chosen for execution or not, i.e. 1 or 0. Finally, (3c) ensures that the total demand is not exceeding the available resources at the cell, AR_c . In this work, we assumed that the buses act VFNs, and provide computational support. For the sake of simplicity, we assume that VFNs are associated to the cells that they are connected to. The task scheduling algorithm in (3a)-(3c) is run at every TTI. This way, the scheduler can dynamically update the resource usage based on service mix.

5 Evaluation

In this section, we analyzed the developed simulator platform's functionality and applicability to the urban area.

5.1 Experimental Setup

We evaluate the functionality of the simulation platform through a case study on data-driven feasibility analysis of different VFC deployment plans for a given geographical area. The input for the simulation platform in this case include a digital map of the area (see Fig. 4) and locations of cellular cells.

We present two main simulation scenarios to test both the functionality of the simulator and gain some insights for VFC. Analysis on Simulator (AoS) scenarios focus on observing the functioning of the simulator and its

Figure 4: The considered urban area.

scalability with the number of vehicles and event rates. In the evaluations, we consider two event rates, 'High' and 'Low'. The arrival rate of high event rate is 20% higher than the low event rate. For AoS scenarios, the service and price differentiation are turned off. In this paper, we consider 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. In these analysis, we also assume that there is only one operator. The second part of our investigation focus on the Evaluation of VFC (EoV) where the service is differentiated through various parameters. EoV-1 and EoV-2 focus on this service differentiation (S.D.) based on QoS expectations and service requirements. In EoV-3 scenario, we analyze the impacts of price differentiation.

We evaluate the performance of VFC from two perspectives. Firstly, we analyze the implications of the hybrid topology on **user experience and the long-term**

Figure 5: The SINR map generated in WinProp.

Table 1: Simulator configurations for different scenarios.

Scenario	Vehicle #	Bus #	S.D.	Pricing	Technology	Expansion	Multi-tenancy	Event Rate
AoS-1	50/100/150	8	Off	Off	Cellular 5G	Off	Off	Low
AoS-2	50/100/150	8	Off	Off	Cellular 5G	Off	Off	High
EoV-1	50/100/150	8	On	Off	Cellular 5G	Off	Off	Low
EoV-2	50/100/150	8	On	Off	Cellular 5G	Off	Off	High
EoV-3	100	8	On	On	Cellular 5G	Off	Off	High

economics. The user experience is estimated based on the average achieved rate and average delay measured per user, while the long-term economic dynamics are evaluated based on the re-investment cost. For this scenario, we assume the network operator only supports one service. Following this first scenario, we extend our investigations to **service heterogeneity**. In particular, we evaluated the impact of service differentiation on the techno-economic dynamics. Four types of services, i.e. background (A_1), machine to machine (A_2), Inelastic (A_3) and Elastic (A_4), are modeled with different QoS expectations and inter-service priorities as detailed in Table 2.

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 cellular base stations are used to serve all the demand.
- **WB Scenario:** In this scenario, we consider a hybrid deployment which is built upon the NB scenario, where VFNs are used to support CFNs.
- **InNB Scenario:** As an alternative to the hybrid deployment, we also consider the case where the provider chooses to double the available CFNs instead of investing in VFNs.
- **B Scenario:** In this scenario, no CFNs are deployed and all the computational tasks rely on VFNs.

We detail the simulation configurations for the scenarios as follows. In the target area, $|C| = 12$ cells are distributed over the coverage area and using Winprop [14],

Table 2: Achieved rate expectations (bps/Hz), the utility values and average demand sizes (kB) for four exemplary service types.

	R_1	R_2	R_3	U_1	U_2	U_3	C_{avg}
A_1	0.05	0.07	0.07	0	1	1	0
A_2	0.01	0.08	0.4	-1	0.7	1	500
A_3	0.1	0.23	0.55	-0.5	0.7	1	1000
A_4	0	1.08	∞	0	1	∞	1500

we estimate the respective SINR map, cf. Fig. 5. A set of vehicular users, $|K| = 50$, is randomly positioned and their mobility is modeled using SUMO [18]. The simulation horizon is discretized and divided into time slots. The simulation is run for $N = 60$ time slots and the presented results are averaged over 50 independent instances. Further details on simulator configurations for different scenarios are presented in Table 1.

5.2 Analysis on Simulator

We analyze the functionality in terms of the achieved rates, experienced delays, and the acceptance ratio. We also investigate the impact of vehicles and the techno-economic constraints on resource pricing.

Average achieved rate. To test the limits of the considered network deployment, we assume that each user tries to reach the maximum achieved rate, i.e. 100 Mbps. Therefore, this scenario is a limit case where the users compete for the spectral resources in extreme levels. We test the average achieved rates for three different scenarios as presented in Fig. 6, namely $|K| = 50$, $|K| = 100$, $|K| = 150$. Note that we also assume a high event rate, more specifically, the vehicles are 20% more likely to send a service request at each TTI.

The increased number of vehicles decreases the average achieved rate per user since we assume that the operators do not change the available bandwidth per cell. For lower number of users, $|K| = 50$, the users achieve between 60 Mbps and 100 Mbps. However, as the number of users increases, we observe a more standard speed among users. More specifically, the achieved rate varieties between 40 Mbps and 60 Mbps for $|K| = 100$, while for $|K| = 150$, this variation decreases to 25 Mbps to 35 Mbps. This is due to the guaranteed fairness within service expectations and the observed resource scarcity. In particular, the resource scarcity imposed by the number of users forces our scheduler to provide a more strict scheduling, and ensure that all the users can reach their standard quality region (i.e. 1 Mbps for this scenario) before allocating further resources to other vehicles.

Average delay. The average delay experienced per user under two different deployment scenarios are presented in Fig. 7. We investigate two major elements of de-

lay, namely the queuing delays and migration delays. As demonstrated in Fig. 7, the main problem of vehicular users is the migration delays, which causes approximately 70% of the measured delay. Deployment of the VFN affects mainly on the queuing delays (up to 10% decrease) due to our task allocation strategy. More specifically, in our task allocation process, we do not reserve or direct VFN resources for the users with high mobility.

Fig. 8 demonstrates the congestion levels of base stations under two different deployments. Here we measure the TTIs where the base station has at least one queued user over the complete simulation horizon. As envisioned, the simulation result demonstrates that the deployment of VFNs would create a relaxation on the stationary fog nodes. However, unlike the InNB scenario, the relaxation in WB is affecting to 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 impacts.

Techno-economic aspects. The applicability of any technical solution in a real-world scenario depends on the economical implications posed by it. Table 3 presents the effects of various deployment options on the network connection and acceptance ratios. In this scenario, we assume that the network connections are ensured by the base stations. Therefore, the application of VFNs does not impact the network connections. Consequently, we observe a connection of 99.97% connection, 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 hybrid platform can increase the efficiency approximately 11%, while InNB scenario provides the highest efficiency. On one hand, considering the doubled resources, the observed increase in the performance can be explained. On the other hand, the applicability of this

Figure 6: Cumulative Distribution Function (CDF) of achieved rate over all the user for different number of users, $|K|$. We assumed 20 MHz bandwidth to estimate the actual achieved rate.

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

Table 3: Estimated network connection (NC) and acceptance ratio (AC) for four different scenarios for $|K| = 100$.

Scenario	NC (%)	AR (%)
NB	99.97	82.15
WB	99.97	91.22
InNB	99.97	98.45
B	99.97	30.92

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 reflecting to a proportional increase in the service price to the end users. We analyze the users' acceptance probability using (4) from [1].

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

In this equation, the left-hand side measures the increase ratio in the utility while the right-hand side measures the increase in price. Considering InNB scenario, doubling the existing resources also doubles the total price while the increment on the measured utility is relatively too low. Therefore, this inequality 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 or less than 12% of the standalone solutions, the users would accept the changes in the prices.

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

Table 4: Estimated acceptance ratios for different services for four different scenarios for $|K| = 100$ and high event rate.

Scenario	A ₁ (%)	A ₂ (%)	A ₃ (%)	A ₄ (%)
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 5: Estimated acceptance ratios for different services for four different scenarios for $|K| = 100$ and low event rate.

Scenario	A ₁ (%)	A ₂ (%)	A ₃ (%)	A ₄ (%)
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

5.3 Evaluation of VFC

This subsection presents the performance of different deployment options under service differentiation constraints.

Inter-service prioritization. As detailed in the earlier sections, we considered four different service types in our evaluations. In this part, we consider two event arrival rates, namely low-rate and high-rate, where the inter-arrival rate for events has increased 20% between two scenarios. The background traffic is considered to have no computational requirements and modelled as a live signal of the vehicles. Therefore, this service type has relatively low achieved rate expectations and strict latency requirements. The background acceptance ratios reported in Table 4 and Table 5 shows the connection status of the vehicles. Considering this service has very high priority in 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, M2M and Inelastic service types have higher priority than the Elastic services. This inter-prioritization is visible in the acceptance ratios. Both for WB and InNB scenarios, the prioritized services (i.e. Inelastic and M2M) have an increase in their overall acceptance ratio between 10–20%, while the non-prioritized services (i.e. Elastic) report a negligible increase in their overall acceptance between 3–5%.

The acceptance ratio difference between NB scenario and WB scenario indicates that the VFC mainly impacts the prioritized services. Note that this result is due to our deployed scheduling strategy. As detailed in the pre-

Table 6: Estimated acceptance ratios for different services for four different scenarios and number of vehicles, $|K|$.

Scenario	$ K $	A ₁	A ₂	A ₃	A ₄
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

vious sections, it is possible to test different scheduling and task allocation strategies that can change the results.

Impacts of traffic load. In this section we analyzed the effects of vehicle numbers in the traffic on the service acceptance rate. In order to reflect the worst case scenario, we assume the high event rate and the results are presented in Table 6.

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 requires high-reliability demand new network solutions. Regarding the considered deployment scenarios, i.e. NB, WB, InNB and B, we can observe that for a low number of users 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, we can see that the standalone VFN deployments, i.e. deployment case B, presents a more than 25% acceptance rate. This results indicates, although this standalone deployment option is not feasible for urban cities, 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 the 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 elastic services, the service provider would have a higher incentive to serve this

Table 7: Estimated acceptance ratios for different services for four different scenarios for $|K| = 100$ and high event rate with price differentiation.

Scenario	A ₁	A ₂	A ₃	A ₄
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

user type. To analyze this scenario, we assume that the price of elastic services are 10 Monetary Units (MU), while the M2M and Inelastic services are prices with 2 MU and 5 MU respectively. Considering the fact that the background services are modeled as live signals, we assume they are priced the least, i.e. 1 MU. The measured acceptance ratios for different deployment options are presented in Table 7.

Comparing Table 4 and Table 7, 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 M2M services, which have the second least price. However, in terms of the Inelastic services, we do not observe any major drop even with relatively low price. This is due to the fact that the prioritization mechanism that we use relies on a multitude of aspects apart from the service prices. With this new pricing strategy, the elastic services 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 specifi-

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

cally, the different prices for services causes an approximately 10% decrease on M2M services, while the other two services report a less than 5% increase. This results demonstrates that despite the pricing strategy can affect the service-based acceptance rates, the price should reflect the supply and demand aspects of a particular service.

6 Related Works

Current edge/fog computing simulators are mainly built on existing cloud computing simulators or network simulators. Build on a cloud computing simulator, CloudSim [6], three well-known edge computing simulators are IFogSim [12], IoTSim [32], and EdgeCloudSim [26]. IFogSim models the fog environment in the Internet of Things (IoT) system where the fog nodes follow a hierarchical arrangement and measures the impact of resource management policies [12]. IoTSim supports simulation of IoT big data processing using MapReduce model. In both works, the network model is simplified. EdgeCloudSim integrates multiple modules in edge computing system, including the core simulation, networking, load generator, mobility, and edge orchestrator modules [26]. Despite considering the user mobility, [26] does not focus on the resource mobility, i.e. VFC.

FogNetSim++ [24] is an edge computing simulator based on OMNet++ [27]. FogNetSim++ focuses on simulating the network characteristics of distributed edge computing devices and enables users' customization of mobility models and fog node scheduling algorithms [24]. However, there is a lack of physical layer protocols. Similarly, Fogbed [9] and EmuFog [19] are two edge computing simulators based on the network simulator Mininet [17] and its extended version MaxiNet [29]. 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 [19]. However, the placement policy is based on latency in the default mode, which can be extended to other evaluation metrics. Fogbed enables the dynamic adding, connecting, and removing of virtual nodes via Docker containers, and supports to perform real world protocols and services [9]. However, the simulator is designed only to target low costs.

Apart from above, FogTorch proposed a simulation tool to support QoS-aware deployment of IoT applications to fog infrastructures [4]. On this basis, FogTorchII exploits Monte Carlo simulations to take into account variations of the QoS and classifies deployments both in terms of QoS assurance and of fog resource consumption [5]. Nevertheless, neither of the works have taken the mobility of the fog nodes into account.

To simulate the vehicular network taking the mobility

into account, Veins couples OMNeT++ [27] with the mobility simulator SUMO [18] with an implementation of IEEE 802.11p [25]. It can also be used together with SimuLTE [28], which offers a detailed model of the Long-Term Evolution (LTE). However, it does not support the simulation of 5G and beyond networks. In our work, we integrate SUMO with WinProp [14]. 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. Besides, It takes various real-world information such as the base stations, buildings, and trees into consideration, thus offering a more realistic and widely applicable vehicular network simulation.

To sum up, our work is different from all the works above. Our simulator supports both stationary and mobile fog nodes, and integrates vehicular traffic and network simulation covering physical and upper layer protocols. Moreover, it is embedded with both spectral and computational resource allocation policies, and can be customized by user-defined algorithms. As demonstrated in the Section 5, the simulator we propose can be used for evaluating various aspects, such as the achieved rate, average delay, and techno-economic performance of different network and task allocation policies, as well as finding the impacts of various factors such as inter-service prioritization, traffic load, and pricing strategy.

7 Discussion and Future Work

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

Simulator analysis. The results in Section 5 prove that the platform provides the necessary functionality to model and test various deployment options and resource management models. Furthermore, our investigation gives a deeper understanding of the techno-economic implications of hybrid fog computing and the practical implications of using a hybrid fog computing environment.

Physical attributes. We extract the physical aspects of the considered network from public-available databases in 3D map format. We choose the base station locations on this map to reflect the actual base station locations. Due to the limitation of an academic license, we have considered a fixed number of base stations. However, it is possible to model larger network topologies by modeling separately at the 3rd party software.

Modular structure. The simulation platform is designed and developed in a modular structure to allow customization. It is possible to integrate different schedulers or pre-trained AI models. Since the task scheduler and network scheduler are separated in our model, they can be customized individually based on the scenario.

Extendability. Our simulator platform is extendable to respond to the evolution of the network and edge technologies. Although we performed our analysis for 5G, it is possible to change base station characteristics and traffic behavior. Moreover, the simulator can work with different resource schedulers. We present further details in the appendix.

Scalability. The designed simulation platform can accommodate more realistic system scenarios with a larger set of vehicles, VFN, or base stations. The time complexity of the overall simulator depends on the used scheduling algorithms and the simulation scenario. Each instance of the EoV-3 case takes 139.705 sec. in a commercially available computer with an i7 processor and 16 GB ram.

Future works. The modular design of the proposed simulator simplifies the integration of new features to our simulator. We are extending the current version to support the multi-tenancy and network slicing as proposed by 3GPP. Moreover, we are also investigating the possible extensions to support long-term (i.e., in the order of months and years) analysis. Finally, to increase our range of scenarios, we will integrate Vehicle-to-vehicle (V2V) communication into future versions.

8 Conclusions

This work presents a simulation environment to model and analyze different VFC environments and application scenarios, with the support for the mobility of fog nodes and realistic modeling of vehicle traffic, resource consumption, and wireless communications. We demonstrate the functionality of the simulator by using it for evaluating the technical and financial feasibility of different VFC deployment plans made for an urban area. Despite being built upon a simple resource management strategy, the results give insights regarding the limits where the application of VFNs is both efficient and profitable. 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.

References

- [1] AKGUL, O. U., MALANCHINI, I., AND CAPONE, A. Dynamic resource trading in sliced mobile networks. *IEEE Transactions on Network and Service Management* 16, 1 (2019), 220–233.
- [2] ATAGOZIYEV, M., SCHMIDT, K., AND SCHMIDT, E. Lane change scheduling for autonomous vehicles**this work was supported by the scientific and technological research council of turkey (tubitak) [award 115e372]. *IFAC-PapersOnLine* 49 (12 2016), 61–66.
- [3] BD, N., L, W., LA, B., AW, W., JA, G., G, G., RK, M., MR, N., DT, B., SB, S., SH, O., ML, B., C, H., AND WI, L.

- Cellmapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types, 2016.
- [4] BROGI, A., AND FORTI, S. Qos-aware deployment of iot applications through the fog. *IEEE Internet of Things Journal* 4, 5 (2017), 1185–1192.
 - [5] BROGI, A., FORTI, S., AND IBRAHIM, A. How to best deploy your fog applications, probably.
 - [6] CALHEIROS, R., RANJAN, R., DE ROSE, C., AND BUYYA, R. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services.
 - [7] CASTIGLIONE, L. M., FALCONE, P., PETRILLO, A., ROMANO, S. P., AND SANTINI, S. Cooperative intersection crossing over 5g, 2019.
 - [8] CHUN, B.-G., IHM, S., MANIATIS, P., NAIK, M., AND PATTI, A. Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems* (2011), pp. 301–314.
 - [9] COUTINHO, A. A., GREVE, F., PRAZERES, C., AND CARDOSO, J. Fogbed: A rapid-prototyping emulation environment for fog computing. pp. 1–7.
 - [10] CUERVO, E., BALASUBRAMANIAN, A., CHO, D.-K., WOLMAN, A., SAROIU, S., CHANDRA, R., AND BAHL, P. Maui: making smartphones last longer with code offload. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (2010), pp. 49–62.
 - [11] ETSI GS ZSM 003 V0.22.0. Zero-touch Network and Service Management (ZSM); End to end management and orchestration of network slicing), 2020.
 - [12] GUPTA, H., DASTJERDI, A., GHOSH, S., AND BUYYA, R. ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments. *Software: Practice and Experience* 47 (06 2016).
 - [13] HASSAN, N., YAU, K.-L. A., AND WU, C. Edge computing in 5g: A review. *IEEE Access* 7 (2019), 127276–127289.
 - [14] HOPPE, R., WÖLFLE, G., AND JAKOBUS, U. 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.
 - [15] HUAWEI. Five trends to small cell 2020, 2016.
 - [16] KOSTA, S., AUCINAS, A., HUI, P., MORTIER, R., AND ZHANG, X. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *2012 Proceedings IEEE Infocom* (2012), IEEE, pp. 945–953.
 - [17] LANTZ, B., HELLER, B., AND MCKEOWN, N. A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks* (New York, NY, USA, 2010), Hotnets-IX, Association for Computing Machinery.
 - [18] LOPEZ, P. A., BEHRISCH, M., BIEKER-WALZ, L., ERDMANN, J., FLÖTTERÖD, Y.-P., HILBRICH, R., LÜCKEN, L., RUMMEL, J., WAGNER, P., AND WIESSNER, E. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems* (2018), IEEE.
 - [19] MAYER, R., GRASER, L., GUPTA, H., SAUREZ, E., AND RAMACHANDRAN, U. Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures. In *2017 IEEE Fog World Congress (FWC)* (2017), pp. 1–6.
 - [20] NAIK, G., CHOUDHURY, B., AND PARK, J. Ieee 802.11bd 5g nr v2x: Evolution of radio access technologies for v2x communications. *IEEE Access* 7 (2019), 70169–70184.
 - [21] NETO, J. L. D., YU, S.-Y., MACEDO, D. F., NOGUEIRA, J. M. S., LANGAR, R., AND SECCI, S. Uloof: A user level online offloading framework for mobile edge computing. *IEEE Transactions on Mobile Computing* 17, 11 (2018), 2660–2674.
 - [22] NOREIKIS, M., XIAO, Y., AND YLÄ-JÄÄSKI, A. Qos-oriented capacity planning for edge computing. In *2017 IEEE International Conference on Communications (ICC)* (2017), pp. 1–6.
 - [23] OPENSTREETMAP CONTRIBUTORS. Planet dump retrieved from <https://planet.osm.org>. 2017.
 - [24] QAYYUM, T., MALIK, A. W., KHAN KHATTAK, M. A., KHALID, O., AND KHAN, S. U. Fognetsim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access* 6 (2018), 63570–63583.
 - [25] SOMMER, C., GERMAN, R., AND DRESSLER, F. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing* 10, 1 (2011), 3–15.
 - [26] SONMEZ, C., OZGOVDE, A., AND ERSOY, C. 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.
 - [27] VARGA, A., AND HORNIG, R. An overview of the omnet++ simulation environment. p. 60.
 - [28] VIRDIS, A., STEA, G., AND NARDINI, G. 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.
 - [29] WETTE, P., DRAXLER, M., SCHWABE, A., WALLASCHEK, F., ZAHRAEE, M., AND KARL, H. Maxinet: Distributed emulation of software-defined networks. pp. 1–9.
 - [30] XIAO, Y., AND CHAO ZHU. Vehicular fog computing: Vision and challenges. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (2017), pp. 6–9.
 - [31] YOUSSEFPOUR, A., FUNG, C., NGUYEN, T., KADIYALA, K., JALALI, F., NIAKANLAHIJ, A., KONG, J., AND JUE, J. P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture* 98 (2019), 289–330.
 - [32] ZENG, X., GARG, S. K., STRAZDINS, P., JAYARAMAN, P. P., GEORGAKOPOULOS, D., AND RANJAN, R. Iotsim: A simulator for analysing iot applications. *Journal of Systems Architecture* 72 (2017), 93–107. Design Automation for Embedded Ubiquitous Computing Systems.
 - [33] ZHU, C., PASTOR, G., XIAO, Y., AND YLAJAASKI, A. Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges. *IEEE Communications Magazine* 56, 10 (2018), 58–63.
 - [34] ZHU, C., TAO, J., PASTOR, G., XIAO, Y., JI, Y., ZHOU, Q., LI, Y., AND YLÄ-JÄÄSKI, A. Folo: Latency and quality optimized task allocation in vehicular fog computing. *IEEE Internet of Things Journal* 6, 3 (2019), 4150–4161.

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.cvs	Data	The user data obtained from SUMO and Winprop
bus.cvs	Data	VFN locations per TTI obtained from SUMO and Winprop

Input Parameters

Data Name	Definiton
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