Data-Driven Capacity Planning for Vehicular Fog Computing

Wencan Mao [1], Ozgur Umut Akgul [2], Abbas Mehrabidavoodabadi [2], Byungjin Cho [2], Yu Xiao [2], and Antti Ylä-Jääski [2]

[1]Aalto University
[2]Affiliation not available

October 30, 2023

Abstract

The strict latency constraints of emerging vehicular applications make it unfeasible to forward sensing data from vehicles to the cloud for processing. To shorten network latency, Vehicular fog computing (VFC) moves computation to the edge of the Internet, with the extension to support the mobility of distributed computing entities. In other words, VFC proposes to complement stationary fog nodes co-located with cellular base stations with mobile ones carried by moving vehicles. Previous works of VFC mainly focus on optimizing the assignments of computing tasks among available fog nodes. However, capacity planning, which decides where and how much capacity to deploy, remains an open and challenging issue. The complexity of this problem comes from the mobility of vehicles, the spatio-temporal dynamics of vehicular traffic, and the computing resource demand generated by varying vehicular applications. To solve the above challenges, we propose a data-driven capacity planning framework that optimizes the deployment of stationary and mobile fog nodes to minimize the installation and operational costs under the quality-of-service constraints, taking into account the spatio-temporal variation in computing demand. Through real-world experiments, we analyze the cost efficiency potential of VFC in long term and demonstrate that the performance loss of VFC is below 6% compared to stationary deployment with equal network capacity. We also analyze the impacts of traffic patterns on the potential cost saving. The results show when the traffic density is higher, more operational costs will be saved in the long run due to more dense deployment of mobile fog nodes.

Data-Driven Capacity Planning for Vehicular Fog Computing

Wencan Mao, Ozgur Umut Akgul, Abbas Mehrabi, Byungjin Cho, Yu Xiao, and Antti Ylä-Jääski

***Abstract*—The strict latency constraints of emerging vehicular applications make it unfeasible to forward sensing data from vehicles to the cloud for processing. To shorten network latency, Vehicular fog computing (VFC) moves computation to the edge of the Internet, with the extension to support the mobility of distributed computing entities. In other words, VFC proposes to complement stationary fog nodes co-located with cellular base stations with mobile ones carried by moving vehicles. Previous works of VFC mainly focus on optimizing the assignments of computing tasks among available fog nodes. However, capacity planning, which decides where and how much capacity to deploy, remains an open and challenging issue. The complexity of this problem comes from the mobility of vehicles, the spatio-temporal dynamics of vehicular traffic, and the computing resource demand generated by varying vehicular applications. To solve the above challenges, we propose a data-driven capacity planning framework that optimizes the deployment of stationary and mobile fog nodes to minimize the installation and operational costs under the quality-of-service constraints, taking into account the spatio-temporal variation in computing demand. Through real-world experiments, we analyze the cost efficiency potential of VFC in long term and demonstrate that the performance loss of VFC is below 6% compared to stationary deployment with equal network capacity. We also analyze the impacts of traffic patterns on the potential cost saving. The results show when the traffic density is higher, more operational costs will be saved in the long run due to more dense deployment of mobile fog nodes.**

***Index Terms*—Capacity planning, vehicular fog computing (VFC), spatio-temporal analysis, integer linear programming (ILP), techno-economic analysis, vehicular networks, intelligent transportation system (ITS), 5G.**

I. Introduction

CLOUD computing has long been the dominant solution for handling large scales of data generated from various sources [1]. However, the traditional cloud strategies are not feasible to the low latency requirement imposed by the emerging vehicular applications, such as cooperative intersection crossing [2] and lane change scheduling [3] for autonomous vehicles. Fog computing, as a promising alternative, moves the computation resource close to the edge of the network [4], and reduces network latency by its proximity to the end-users and dense geographical distribution [5].

In the scenarios of fog computing, distributed fog computing entities (a.k.a. fog nodes) can be installed in network infrastructures such as cellular base stations and road side units. We call the ones co-located with such as the cellular fog nodes (CFNs). In order to guarantee the quality of service (QoS) received by the end-users, the stationary deployment almost always leads to the over-provisioning of the resources, turning the service provisioning into a non-profitable business model. Motivated by this techno-economic pressure, vehicular fog computing (VFC) was proposed to enable mobility of fog nodes by installing fog nodes on moving vehicles, and to utilize the mobility to satisfy dynamic computing resource demand with lower costs [4]. We call the fog nodes carried by vehicles (e.g. buses, taxis and drones) as vehicular fog nodes (VFNs). The VFNs perform as local cloud servers and provide computing service to the surrounding vehicles [6].

Previous works on VFC focus on task assignment among available fog nodes. For example, Zhu et al. proposed a joint optimization solution to assign the tasks generated from vehicles across the stationary and mobile fog nodes under the constraints of service latency, quality loss, and fog capacity [7]. Capacity planning, which focuses on determining the locations and capacities of fog nodes, is different from the task allocation problem. Noreikis et al. proposed a capacity planning solution for edge computing that satisfies the QoS requirements while minimizing the number of required edge computing nodes [8]. However, their framework considers only stationary deployment of fog nodes, and therefore cannot be applied to VFC.

Capacity planning for VFC remains an open issue and it is challenging because of the following reasons. Firstly, vehicular traffic has high spatio-temporal diversity, where the traffic flow depends on the time-of-day and the geographic location [9]. The capacity planning for VFC requires a deep understanding of the spatio-temporal dynamics of the vehicular traffic. Secondly, in order to estimate the computing resource demand from the vehicular application users, the resource consumption pattern of various vehicular applications should be taken into consideration. Thirdly, VFNs are supposed to serve the vehicles within the one-hop communication range. The mobility of VFN adds a layer of complexity to the analysis of service availability and cost estimation.

In this paper, we propose a data-driven capacity planning framework that takes real-world traffic data and application profiles as inputs, and outputs a cost-optimal deployment plan of CFNs and VFNs using Integer Linear Programming (ILP). Our framework determines the number and types of fog nodes in divided region to satisfy the computing resource demand from the vehicular traffic environment.

The contributions of this work are listed as follows.

1) To the best of our knowledge, this is the first work on data-driven capacity planning for VFC. It provides methods for estimating the spatio-temporal distribution of computing resource demand from real-world traffic data and application profiles, and provides a mathematical model for minimizing the installation and operational costs under QoS constraints

through optimal deployment of fog nodes on cellular base stations and buses.

2) Through evaluation with real-world datasets, we show the potential of utilizing the mobility of fog nodes to fulfil dynamic computing resource demand with lower costs, compared with traditional stationary deployment on cellular base stations. Our evaluation also indicates that the deployment of mobile fog nodes would cause a performance gain up to 9.3% compared to stationary deployment with fixed network capacity and a performance loss less than 6% compared to that with extended network capacity, which makes VFC a promising solution for urban network deployment.

3) Our study provides deep insights on the impact of traffic patterns on the fog nodes deployment strategies by comparing the potential cost savings between areas with different traffic characteristics (e.g. downtown with dense traffic flows vs. suburb with low traffic volume) and between weekdays and weekends. The mobility of VFNs provides flexibility of adjusting the capacity deployment with varying demand.

The rest of this paper is organized as follows. Section II gives an overview of VFC. Section III introduces the data-driven methodology of the capacity planning platform for VFC. Section IV presents methods for estimating computing resource demand based on real-world traffic data and application profiles. Section V formulates the optimization problem for capacity planning. Experimental setup and results are discussed in Section VI and Section VII, respectively. Section VIII discusses the computational complexity of the model and the future directions. Section IX presents the related work before we conclude the work in Section X.

II. Vehicular Fog Computing

In this section, an overview of the VFC paradigm is presented. First, we demonstrate an application scenario of VFC. Then, we introduce the vehicular communication technologies to support the implementation of VFC.

A. VFC Application Scenario

Fig. 1 presents an application scenario of VFC. In this scenario, Vehicle A generates an object detection task in order to recognize the traffic signs. It is within the communication range of a bus which carries a VFN. Thus Vehicle A offloads the task to the bus. Meanwhile, Vehicle B generates a lane detection task, which is offloaded to a CFN co-located with the connected cellular base station.

In case more than one fog node is available within the communication range of a vehicle, task allocation algorithms are used to decide where to offload the tasks generated by the vehicle to improve the QoS and achieve better techno-economic performance. Capacity planning, on the other hand, focuses on planning where to deploy the fog nodes and how much computing capacity should be deployed in order to fulfill estimated computing resource demand with better techno-economic performance. In this paper, we focus on capacity planning for VFC, taking the mobility of vehicles including VFNs into account.

Fig. 1: Application Scenario of VFC.

B. Vehicular Communication Technology

Dedicated short range communications (DSRC) and cellular V2X (C-V2X) are the most widely used radio access technologies for vehicular communication. DSRC uses an orthogonal frequency division multiplexing (OFDM)-based physical layer with a channel bandwidth of 10 MHz [10]. C-V2X, which is developed by 3GPP, makes use of the widely distributed cellular infrastructure. Besides, it defines additional transmission modes that allow direct V2X communication using sidelink channels [10]. According to [11], DSRC and C-V2X can support basic safety applications as long as the vehicular density is not very high. The basic safety applications are mainly based on advertising driving alerts periodically about potentially dangerous situations. The latency requirements for these applications are 100ms [10].

Moreover, IEEE 802.11bd and 5G NR V2X are designed to support the advanced vehicular applications characterized by high-reliability and low-latency requirements [10]. These applications, so-called advanced vehicular applications, aim to increase driving safety and benefit traffic management. 3GPP has divided the advanced vehicular applications into four categories, i.e. vehicle platooning, advanced driving, extended sensor, and remote driving. Their latency requirements are 10-500ms, 3-100ms, 3-100ms, and 5ms respectively [12].

In this work, we consider 5G NR V2X as the communication module among the vehicles, which enables vehicular communications either within or out of the gNodeB coverage, and supports multiple communication types (i.e. broadcast, groupcast, and unicast) and message types (i.e. periodic and aperiodic). A system-level evaluation of the 5G NR V2X is provided by [13] using a 60 kHz sub-carrier spacing, a 20 MHz channel, and a transmission rate of 10Hz. The results show that under highway scenarios, the packet delivery rate (PDR) is around 99.7% to 99.8%; under urban scenarios, the PDR varies from 93% to 97% [13]. This means 5G NR V2X can provide high-reliable vehicular communication for VFC.

III. Data-driven Capacity Planning

We follow a data-driven methodology to plan for the deployment solutions of CFNs and VFNs. An overview of the data-driven capacity planning process is given in Fig. 2.

The capacity planning is implemented in three steps, namely demand estimation, cost minimization, and bus scheduling. Three types of data are used as the inputs. The vehicular traffic data and application profiles are used for demand estimation, and the bus mobility data are used for cost minimization and bus scheduling.

The first step is to estimate the computing resource demand generated by vehicles, which varies over time and between locations. The demand depends on the spatio-temporal distribution of vehicular traffic and the resource consumption profiles of vehicular applications. The latter describes the usage pattern of CPU and GPU resources for each application. Based on real-world traffic datasets, we propose to apply spatio-temporal analysis methods, such as clustering, traffic flow theory, and Gaussian Process Regression to model traffic flows (see Section IV-A). Meanwhile, we choose a set of representative vehicular applications as examples, and profile their resource usages under different latency constraints (see Section IV-B). The output of the demand estimation module defines the minimum amount of computing capacity (in terms of the number of fog nodes with fixed unit size) required in each cluster at each time slot with respect to the traffic flows and QoS requirements (see Section IV-C).

The second step is to find out a cost-optimal deployment plan based on the estimated demand and the potential supply (see Section V-B). We assume that VFNs would be installed on commercial fleets like buses, due to their predictable mobility patterns (e.g. schedules, driving routes). Accordingly, the supply of VFNs depends on the mobility pattern of buses, while the supply of CFNs depends on the deployment of cellular base stations. Based on real-world bus schedules, we divide a target area into clusters, and map bus journeys using a spatio-temporal availability matrix (see Section V-A). Here a bus journey defines the driving route as well as the time-of-day when the trip starts. The same journeys are typically repeated on a daily basis during weekdays, and on a weekly basis during weekends. The same bus journeys may be served by different buses on different days.

The outputs of the cost minimization module include the deployment plan of CFNs, the selection of bus journeys, and the minimized operational cost. In this module, we assume that VFNs are installed on all the buses in the study area. This may cause oversupply of VFNs. To solve this issue, the last step is to run the bus scheduling module to identify a minimal subset of buses for covering the selected bus journeys for VFN deployment. The bus journeys that are belong to the same bus line while having sufficient shifting time in between are chained together. This means the same bus can be reused for implementing different journeys. In this way, the installation cost of VFNs can be minimized.

IV. Demand Estimation

In this section, the approach of demand estimation is introduced. First of all, it is important to understand how the vehicular traffic vary over time and among locations, thus the spatio-temporal traffic model is established. Besides, we also need to know the consumption pattern of the vehicular applications, thus the process of application profiling is illustrated. With these inputs, the demand estimation problem is formulated.

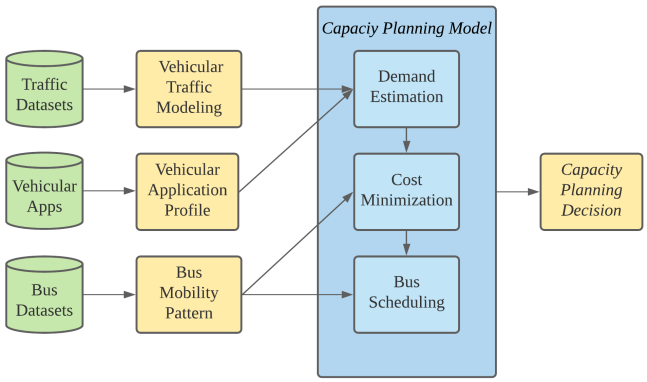


Fig. 2: Flowchart of data-driven capacity planning.

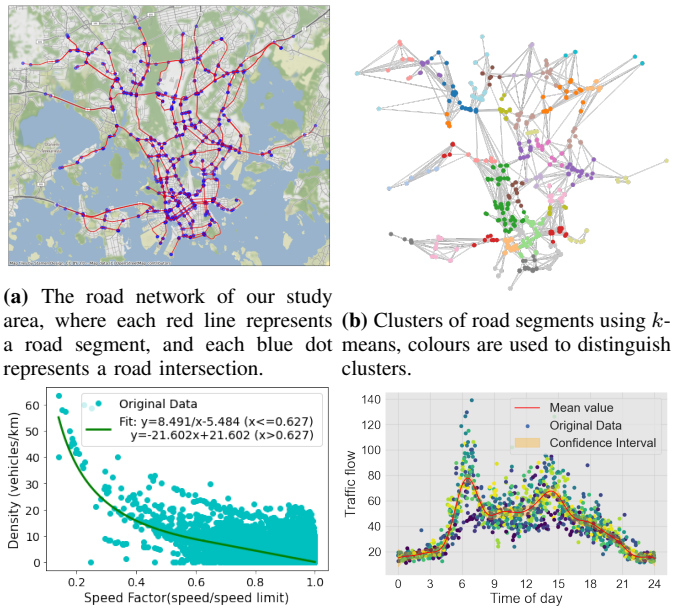
A. Spatial-temporal Traffic Modeling

The process of spatio-temporal traffic modeling is shown in Fig. 3. Based on the road network, clustering is used to group the road segments into clusters. Two types of traffic datasets are used to derive the traffic density according to traffic flow theory [14]. And Gaussian Process Regression is used to model the daily traffic flow as a distribution of time-of-day.

1) Road Network: A graph $G = (V, E)$ is used to represent a road network, where each vertex V represents a road segment, and each edge E represents a road intersection. The road segment is the basic unit of the road network, and the road intersections represent the topological relationship of the road segments.

2) Road Segment Clustering: The road segments are grouped into clusters based on the geographical relationship among them, and the traffic flow is accumulated in each cluster to estimate the demand. At each time slot, the CFNs and VFNs will serve the client vehicles that are within the same cluster. K-means is used for clustering the road segments, and there should be at least one base station inside each cluster.

3) Traffic Density Derivation: According to the traffic flow theory, the basic variables of traffic flow are the average speed, flow rate, and density; and if we know any two of these variables, we can always get the value of the last variable [14]. There are usually two types of traffic datasets. One type is the speed dataset, which samples the average speed of the vehicles on each road segment at each time slot. The other type is the flow rate dataset, which records the number of vehicles that pass through a site during a time interval. By applying traffic flow theory to the two types of dataset, the density of each road segment can be derived. More precisely, the scatter plot of the traffic density is calculated through dividing flow rate of each lane by the average speed. Then the traffic density is fitted using piece-wise regression with respect to the normalized speed (i.e. the ratio of average speed and speed limit). In this way, we are able to calculate the traffic density of all the road segments based on their normalized speed. The traffic density



(a) The road network of our study area, where each red line represents a road segment, and each blue dot represents a road intersection.

(b) Clusters of road segments using k-means, colours are used to distinguish clusters.

(c) Traffic density versus normalized speed based on traffic flow theory.

(d) An example of traffic flow versus time-of-day during the weekdays.

Fig. 3: Process of spatio-temporal traffic modeling.

is further used to estimate the number of vehicles on each road segment at each time slot (i.e. spatio-temporal traffic flow) by multiplying the traffic density with the number of lanes and the length of the road segment.

4) Traffic Flow Modeling: Gaussian Process Regression is used to model the daily traffic flow in terms of vehicles per cluster, with the predicted mean and variance functions. Assume the mean function and variance function in a cluster have the value $\bar{X}$ and $\hat{\sigma}$ at a certain time. To get the upper limit of the confidence interval, we use $\bar{X}+\beta\times\hat{\sigma}$ to denote the spatio-temporal traffic flow, where β represent the coefficient of the variance in the confidence interval. The traffic flow is modeled separately during weekdays and weekends due to different time-of-day pattern (see Section VII-A).

B. Vehicular Application Profiling

Apart from the vehicular traffic, the demand of the fog computing system also depends on the resource consumption of the vehicular applications, which is reflected in the CPU and GPU consumption [8]. The vehicular applications are containerized into Docker Images, and a set of benchmark testing is designed for each containerized application. The application benchmark testing algorithm is shown in Algorithm 1. After getting the vehicular application profiles, nonlinear least squares regression is used to map the mathematical relationship among the CPU usage, the GPU usage, and the latency. Different standards of latency requirement are set for all the users, and the CPU and GPU consumption of each application under each latency requirement is calculated according to the regression results.

Algorithm 1: Application benchmark testing algorithm

Input: computing latency requirement $r_{compute}$
Output: mean of frame latency $\mu_{compute}$, variance of frame latency $\sigma_{compute}$, mean of CPU usage μ_{cpu}, mean of GPU usage μ_{gpu}

```
i = 1;
while μ_compute ≤ r_compute do
    start the docker service of i replicas of application;
    while service is running do
        record the frame latency;
        record the CPU and GPU usage per second;
    end
    calculate μ_compute, σ_compute, μ_cpu, μ_gpu;
    i = i + 1;
end
```

C. Formulation of Demand Estimation

The demand estimation problem aims to find the minimum amount of computing capacity required in each cluster at each time slot to meet the computing tasks generated from the users of vehicular applications. And the computing capacity is represented as the number of fog nodes with a fixed unit size. The set of computing tasks is represented by I. Assume each user will keep one active computing task at each time slot, then the number of the computing task is equal to the number of the users. This is denoted as $|I| = n$, where n is the number of users, and $|.|$ represents the cardinality of the set. The CPU and GPU consumption of each computing task p is represented by $c(p)$ and $g(p)$ respectively. The CPU and GPU consumption depend on the vehicular application type and the latency requirement. We assume the latency requirement is universal for the users at each time. We also need to know the maximum capacity of the CPU and GPU according to their configuration, which is represented by B_{CPU} and B_{GPU} respectively. In this work, all the fog nodes are homogeneous, so they have the same CPU and GPU configuration.

The demand estimation problem is given in (1a) to (1e). Our objective function is (1a), where $\lceil . \rceil$ represents the ceiling function. The object function reflects that a fog node will be needed if at least one computing task is packed inside, and it minimizes the required number of fog nodes to pack all the computing tasks generated from the users. Constraint (1b) is the *CPU configuration constraint*, which ensures that the computing tasks packed to each fog node does not exceed the CPU configuration of the fog node. Constraint (1c) is the *GPU configuration constraint*, which prevents the computing tasks packed to each fog node from exceeding the GPU configuration of the fog node. Constraint (1d) is the *non-repetitive assignment constraint*, which ensures each computing task is assigned to exactly one fog node. Finally, constraint (1e) defines the binary decision variable x_{pq}, which indicates if the task p is packed to the fog node q.

Since the objective of the above problem is to determine the minimum number of fog nodes, the problem can be solved in polynomial time using the algorithm as follows. We start by picking a random fog node and assigning the tasks to it until

it is full. We repeat this process of picking random fog nodes and assigning tasks until all the tasks are allocated or all the fog nodes are full.

$$\min_{x_{pq}} \quad \sum_{q=1}^{n} \lceil \frac{\sum_{p \in I} x_{pq}}{n} \rceil \tag{1a}$$

$$\text{s.t.} \quad \sum_{p \in I} c(p) x_{pq} \leq B_{\text{CPU}}, \forall q \in (1, 2, ..., n), \tag{1b}$$

$$\sum_{p \in I} g(p) x_{pq} \leq B_{\text{GPU}}, \forall q \in (1, 2, ..., n), \tag{1c}$$

$$\sum_{q=1}^{n} x_{pq} = 1, \forall p \in I, \tag{1d}$$

$$x_{pq} \in (0, 1), \forall p \in I, \forall q \in (1, 2, ..., n). \tag{1e}$$

V. Cost-Optimal Fog Nodes Deployment

This section focuses on finding out how to deploy CFNs and VFNs in order to fulfill the estimated computing resource demand from vehicular traffic environment. Different from CFNs which will be deployed at stationary cellular base stations, the locations of VFNs depend on the schedules and driving routes of the carriers, which are buses in this case. Therefore, we start by estimating the availability of VFNs based on the mobility pattern of buses, and then move to the problem of minimizing the operational and installation costs through optimal distribution of computing capacity on the cellular base stations and buses. The architecture of the capacity planning model is shown in Fig. 4.

A. Bus Mobility Pattern

We assume that bus journeys are planned beforehand. A bus journey m can be described with the following parameters, including the bus line l_m, the direction r_m, the departure time dp_m, and the one-way travel time tr_m. The bus line and direction together determine the driving route of the bus journey. During a journey, a bus may go through several road segment clusters.

1) Spatio-temporal Availability Matrix: To model the mobility of buses, we define a spatio-temporal availability matrix $\mathbf{A}(i, t)$ that describes the spatio-temporal distribution of bus journeys. For each cluster i and each time slot t, the spatio-temporal availability matrix is represented as a vector of size $(u \times 1)$, where u is the number of bus journeys in the study area. Each element in the vector is a binary indicating the availability of the bus taking the journey. The value is 1 if the bus taking the each journey is traveling in the cluster in question; and is 0 if the bus journey does not cover the cluster in question, or if the bus is traveling in another cluster. When a VFN is deployed on a bus, the VFN service becomes available along the bus journeys taken by the bus. When estimating the spatial distribution of VFNs in each time slot, we go through such bus journeys, and calculate the communication range of each VFN. Each VFN is associated with the nearest road segment cluster within its communication range. Tasks generated within a cluster are supposed to be executed only on the associated VFNs.

2) Adjacency of bus journeys: In addition, we use a parameter to indicate the adjacency relationship between each pair of journeys m and n. To be more precise, it indicates whether journey n can be covered after journey m by the same bus. A journey can be adjacent with another if both journeys belong to the same bus line, and the departure time of the upcoming journey is later than the arrival time of the last journey. Therefore, the adjacency between journey m and journey n is defined as:

$$c_{mn} = \begin{cases} 1, & \text{if } l_m = l_n \text{ and } r_m = r_n \\ \text{and } dp_m + 2tr_m \leq dp_n & \\ 1, & \text{if } l_m = l_n \text{ and } r_m \neq r_n \\ \text{and } dp_m + tr_m \leq dp_n & \\ -\infty, & \text{otherwise.} \end{cases}$$

B. Cost Minimization

The cost minimization problem aims to minimize the overall installation and operational costs of the fog computing system. The installation cost per fog node is represented by c_{cap}. The overall installation cost in this problem is represented by C'_{cap}, where $'$ indicates that the installation cost will be further minimized. The installation cost include the investment of purchasing and installing the fog nodes, which is paid only once. Assuming the VFNs are installed on all the buses in the study area, we need to know the overall number of buses in the area n_0.

The operational cost of CFN and VFN per unit time is presented by c_{opc} and c_{opv} respectively, and the overall operational cost is represented by C_{op}. The operational cost include the rent, the fee of power consumption (e.g. fuel, electricity) and regular maintenance, which is proportional to the operating time. To calculate the operational cost, we also need to specify the operational time T in days, and record the duration t_j of each bus journey j ($j \in U$) in the study area, where U is set of bus journeys, and the number of the bus journey is denoted as u. Apart from above, the inputs also include the spatio-temporal demand estimation, represented by the demand d_{it} in cluster i ($i \in S$) at time slot t ($t \in W$). S is the set of clusters, and W is the set of time slots in a day. And we also need the spatio-temporal availability matrix $\mathbf{A}(it)$ from the bus dataset, represented by a vector of size $(u \times 1)$ in cluster i at time slot t.

The installation cost assuming the VFNs are installed on all the buses in the area can be written as:

$$C'_{\text{cap}} = c_{\text{cap}}(\sum_{i=1}^{S} n_i + n_0).$$

The operational cost can be written as:

$$C_{\text{op}} = T(c_{\text{opc}} \times W \sum_{i=1}^{S} n_i + c_{\text{opv}} \times \sum_{j=1}^{U} t_j x_j).$$

The cost minimization problem is given in (2a)-(2d). Our objective function, cf. (2a), minimizes the overall installation and operational costs. Constraint (2b) is the *spatio-temporal capacity constraint*, which ensures that the capacity provided by the cellular and vehicular fog nodes is equal or larger than

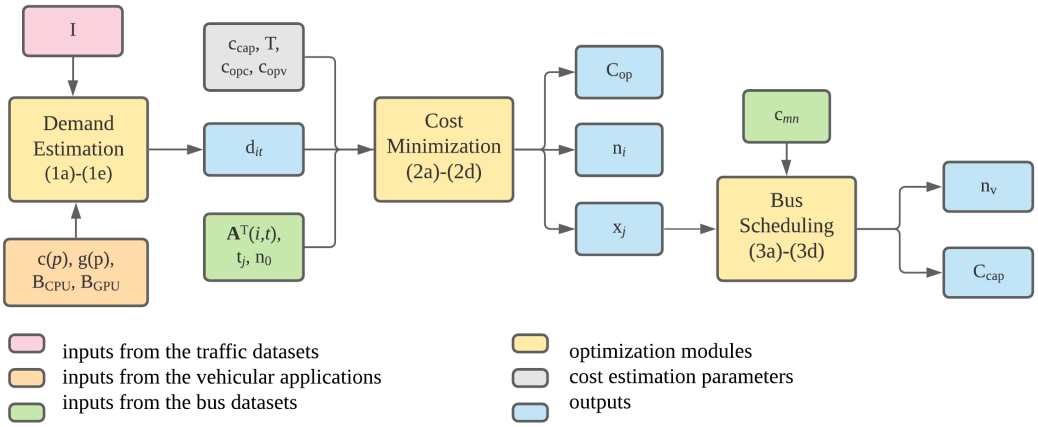


Fig. 4: Flowchart of the capacity planning model.

the demand estimation for each cluster at each time slot. To plan for the deployment of CFNs and VFNs, two decision variables are defined in Constraints (2c) and (2d). The first decision variable n_i is an integer variable to indicate the number of CFNs in cluster i. The second decision variable x_j is a binary variable to indicate whether the bus journey j is selected to serve as the vehicular fog nodes. The vector form of all the vehicular fog nodes decision is $\mathbf{X}$, with size of $(u \times 1)$.

$$\min_{n_i, x_j} \quad C'_{\text{cap}} + C_{\text{op}} \tag{2a}$$
$$\text{s.t.} \quad n_i + \mathbf{A}^T(i,t)\mathbf{X} \geq d_{it}, \forall i \in S, \forall t \in W, \tag{2b}$$
$$n_i \in \mathbb{Z}^+, \forall i \in S, \tag{2c}$$
$$x_j \in (0,1), \forall j \in U. \tag{2d}$$

C. Bus Scheduling

The bus scheduling problem aims to find the minimum number of buses to install VFNs so as to further minimize the installation cost. The minimal decomposition model [15] is used to schedule the buses. The set of selected journeys is represented by J. The number of the selected journeys $|J| = k$, where $|.|$ represents the cardinality of the set. In order to schedule the buses, we also need the adjacency relationship c_{mn} for each journey pair m and n.

The bus scheduling problem is given in (3a)-(3d). The objective function is (3a), based on the Dilworth Theorem of partial ordered sets [15]. It minimizes the number of buses to cover the selected journeys (i.e. the minimum decomposition of the bus journey set J [15]). Constraint (3b) and (3c) are the *non-repetitive scheduling constraints* in two directions, which guarantees each bus journey is either covered by an individual bus or covered in a sequence of bus journeys. Finally, Constraint (3d) defines the binary decision variable, which indicates whether a bus is scheduled to cover journey n after journey m.

$$\min_{b_{mn}} \quad k - \sum_{m \in J} \sum_{n \in J} c_{mn} b_{mn} \tag{3a}$$
$$\text{s.t.} \quad \sum_{n \in J} b_{mn} \leq 1, \forall m \in J, \tag{3b}$$
$$\sum_{m \in J} b_{mn} \leq 1, \forall n \in J, \tag{3c}$$
$$b_{mn} \in (0,1), \forall m \in J, \forall n \in J. \tag{3d}$$

With the minimum number of buses, the minimized installation cost can be written as:

$$C_{cap} = c_{cap}(\sum_{i=1}^{S} n_i + n_{\text{v}}), \tag{4}$$

where n_{v} equals to the value of objective function in the bus scheduling problem.

VI. Experimental Setup

Real-world datasets and applications are used to validate the capacity planning framework. This section introduces the datasets, applications, and the simulation settings.

1) Helsinki Speed Dataset: The Helsinki city map is extracted from latitude 60.222306, longitude 24.858754 to latitude 60.142211, longitude 24.993980. In the map, 869 road segments are included. Each road segment has its geographical information as a set of sequential coordinates with start and end intersections. the Helsinki speed dataset is collected using HERE Traffic API. The speed of all the road segments are sampled every minute from January 1^{st} to February 15^{th} 2020.

2) Helsinki Flow Rate Dataset: the Helsinki flow rate dataset is published by Traffic Monitoring System (TMS) of Finnish Transport Agency. The traffic monitoring stations are located at all the major roads in Finland, and the flow rate of these roads in the same study area are sampled during the same time period as the speed dataset. Combining the speed dataset with the flow rate dataset, we can get the spatio-temporal traffic models.

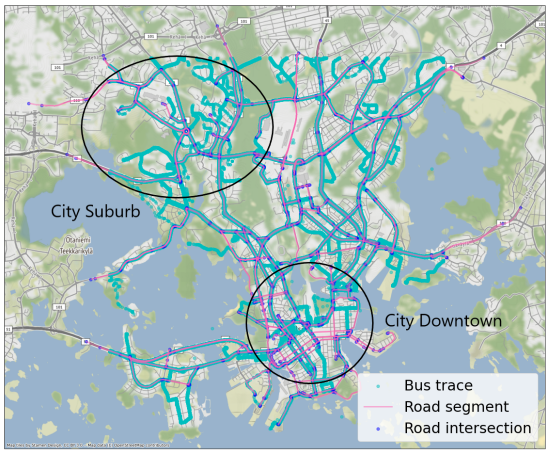


Fig. 5: Helsinki city map with the road network, bus traces, and the locations of downtown and suburb areas.

3) Helsinki Bus Position Dataset: the Helsinki bus position dataset is collected using HSL High Frequency Positioning (HFP) API. All the buses in Helsinki publish their status, including their bus line, vehicle id, direction, departure time, as well as real-time position around once per second. The bus position data is re-sampled into every minute to correspond with the traffic dataset. From the bus position dataset, we can get the bus mobility pattern.

4) Helsinki Bus Timetable: the Helsinki bus timetable is published by HSL General Transit Feed Specification (GTFS). It records the departure time and arrival time of all the bus journeys at every bus stop. From the bus timetable, we get the adjacency relationship of the bus journeys.

5) City Downtown and City Suburb: The city downtown and city suburb areas are extracted from the Helsinki city map by further clustering the road segment clusters based on their topological relationship, and they both consist of 6 clusters. The locations of the downtown and suburb areas are shown in Fig. 5. The city downtown includes the city center, central railway station, commercial area, and high-density residential area. The city suburb includes the highways, natural parks, and the low-density residential area.

A. Vehicular Applications

To exemplify the compute-intensive and latency-sensitive vehicular applications, four computing tasks are selected for testing, namely object detection, semantic segmentation, lane detection, and video transcoding.

1) Object Detection: the object detection application is implemented through YOLOv5s [16] trained on COCO dataset.

2) Semantic Segmentation: the segmentation application is implemented through Image Segmentation Keras [17] with VGG-UNET model and trained on Cityscapes dataset.

3) Lane Detection: the lane detection application is implemented by OpenCV in Python environment.

4) Video Transcoding: the video transcoding application is implemented through HandBrake video transcoder with x265 video encoder and mp4 container.

The CPU used for testing is the Intel Core i7-7700K. The CPU has 8 threads which can run in parallel. Assuming the computing capacity for each thread is 100%, the capacity of the CPU $B_{\mathbf{CPU}} = 800\%$. The GPU used for testing is the NVIDIA GeForce RTX 2080 Ti. The GPU resource is used as an integral, so the capacity of the GPU $B_{\mathbf{GPU}} = 100\%$.

TABLE I: Capacity planning simulation settings.

	Simulation 1	Simulation 2	Unit
c_{cap}	1000	500, 1000, 1500, 2000	MU/device
c_{opc}	0.02	0.02	MU/minute
α_{op}	0.5, 1.0, 1.5, 2.0	1.0	MU/minute
T	1300	780, 1040, 1300, 1560, 1820	day
r_{compute}	250, 150, 100	100	millisecond
β	3		/
p_{task}	1:1:1:1		/
W	1440		minute
S	6 in downtown, 6 in suburb		cluster
U	5189 in downtown, 5853 in suburb		journey
n_0	543 in downtown, 603 in suburb		bus

B. Simulation Setup

To validate the functionality of our model, we consider three deployment options in the experiment, as detailed below.
Option 1: deploying fog nodes on both cellular base stations and buses, and installing VFNs on minimum number of buses that can cover the selected bus journeys (i.e. where the number of VFNs equals to n_v).
Option 2: deploying fog nodes only on cellular base stations.
Option 3: deploying fog nodes on both cellular base stations and buses, and installing VFNs on all the buses in the study area (i.e. where the number of VFNs equals to n_0).

Among these three options, Option 1 is the output of proposed capacity planning framework, while the other two options are used for comparison. Option 2 corresponds to the traditional stationary fog node deployment model, and Option 3 corresponds to the output of the capacity planning model without bus scheduling block.

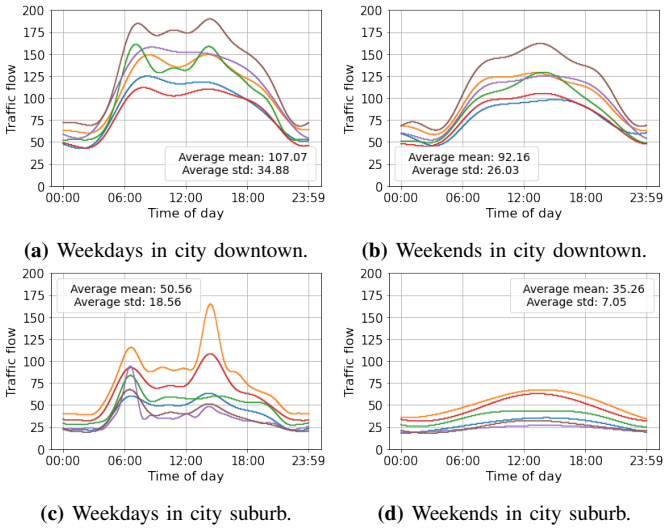
Two sets of simulations are listed in Table I. The given costs are purely designed for comparison purposes and given in Monetary Units (MU). However, we investigate the impacts of different cost ratios in Section VI-B. In both simulations, the confidence interval in traffic modelling is selected as three times of standard derivation. And for each user, the probability of selecting each type of computing tasks is regarded as equal.

The first simulation scenario analyzes the impacts of latency requirement and relative operational cost on the deployment of CFNs and VFNs. To analyze the implications of latency requirement, we change the computing latency requirements among 250ms, 150ms, and 100ms. We have defined the relative operational cost as the operational cost of CFNs per time unit divided by the operational cost if VFNs per time unit, which is denoted as:

$$\alpha_{\text{op}} = c_{\text{opc}}/c_{\text{opv}}. \tag{5}$$

To analyze the implications of the relative operational cost, we change it from 0.5 to 2.0, while keeping other parameters as constants.

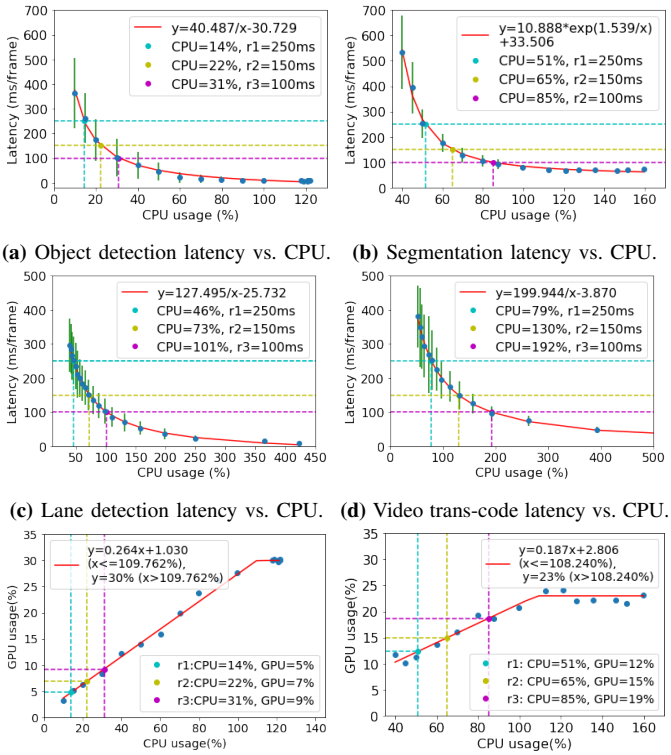
The second simulation scenario compares the installation and operational costs of deployment options under different unit installation cost and operational time. In this set of



(a) Weekdays in city downtown. **(b)** Weekends in city downtown.

(c) Weekdays in city suburb. **(d)** Weekends in city suburb.

Fig. 6: Spatio-temporal traffic models, where each color represents a cluster, and the traffic flow is represented by the number of vehicles in each cluster.



(a) Object detection latency vs. CPU. **(b)** Segmentation latency vs. CPU.

(c) Lane detection latency vs. CPU. **(d)** Video trans-code latency vs. CPU.

(e) Object detection GPU vs. CPU. **(f)** Segmentation GPU vs. CPU.

Fig. 7: Vehicular application profiles, where the red curves represent the regression results, and the dashed lines represent the computing latency requirements.

simulation, while keeping other parameters as constants and setting relative operational cost to 1.0, we change the unit installation cost from 500 MU/device to 2000 MU/device, and change operational time from 780 days to 1820 days (i.e. approximate working days from 3 years to 7 years). To estimate the overall costs in the long run, we assume that the vehicular traffic and bus mobility pattern remain unchanged during the operational time.

We compare the deployment decision as well as the cost estimation in city downtown versus suburb, on weekdays versus weekends. For the convenience of comparison, we set the time range of the weekend models equal to the weekday models. The proposed framework in Fig. 4 is formulated as separate ILP models. Among these models, the demand estimation module is solved using the heuristic method detailed in Section IV-C. The remaining modules are developed in Python 3.8 and solved using Gurobi [18] solver.

VII. Experimental Results

In this section, the results of traffic modeling, application profiling, fog node deployment, and cost estimation are presented, and the impacts of traffic pattern, latency requirements, and cost estimation parameters are analyzed. Furthermore, the service provision of different capacity planning strategies is evaluated through a VFC simulation.

A. Traffic Models

Fig. 6 shows the spatio-temporal distribution of the traffic flow, where each color represent a cluster in the area. The mean and standard deviation of the traffic flow are derived for each cluster, and the average values for all the clusters in the area are calculated for each traffic model. Despite having a smaller geographical coverage, downtown area accommodates a larger traffic volume with respect to suburb area. So the traffic density is higher in the city downtown, especially during the weekdays. The traffic flow has different time-of-day pattern between weekdays and weekends. During the weekdays, there are usually two peaks in the traffic flow due to the morning and evening commuting hours. However, on the weekends, we usually observe one peak around the noon. The traffic flow also shows difference between downtown and suburb areas through the day. In the downtown area, the variations of the daily traffic flow are generally larger than the suburb area.

B. Application Profiles

The application profiles are shown in Fig. 7, represented by the latency versus the CPU and GPU consumption of each application. From the figure, we can see object detection and semantic segmentation are GPU-intensive applications, while lane detection and video transcoding are CPU-intensive applications. Generally speaking, within the appropriate range, the CPU usage is in inverse proportion to the mean latency, and the GPU usage is in direct proportion to the CPU usage. As a consequence, when the latency requirement become more stringent, the CPU and GPU usage will become higher until they have reached the maximum value.

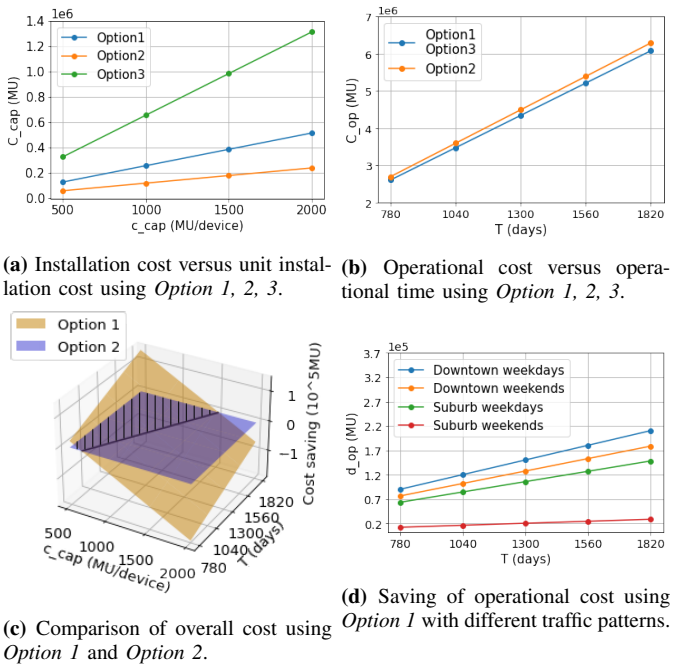
C. Fog node Deployment

Fig. 8 shows the impacts of latency requirement, relative operational cost, and the traffic pattern on the fog node deployment using *Option 1*. In the figure, the x-axis is the relative operational cost, and the y-axis shows the number of fog nodes. The number of CFNs and VFNs are stacked

(a) Weekdays in city downtown. **(b)** Weekends in city downtown.

(c) Weekdays in city suburb. **(d)** Weekends in city suburb.

Fig. 8: Effects of latency requirement, relative operational cost, and traffic models on fog node deployment decisions.



(a) Installation cost versus unit installation cost using *Option 1, 2, 3*.

(b) Operational cost versus operational time using *Option 1, 2, 3*.

(c) Comparison of overall cost using *Option 1* and *Option 2*.

(d) Saving of operational cost using *Option 1* with different traffic patterns.

Fig. 9: Cost analysis of different deployment options.

together in each bar, and the three bars in parallel represent the three settings of latency requirement.

Impacts of latency requirement: if we compare the three bars from left to right, we can see the latency requirement will influence the overall number of fog nodes. When the latency requirement becomes stricter, more computing resource are required, thus more fog nodes will be deployed.

Impacts of relative operational cost: comparing the x-axis from left to right, we can see the relative operational cost will influence the proportion of VFNs. When the relative operational cost increases, the percentage of VFNs in the downtown area will increase until it reaches the maximum value. This is because when the operational cost of VFNs becomes relatively cheaper, the model tends to select more bus journeys. However, not all of the bus journeys are suitable for deploying the VFN. In another word, at certain times and places, deploying CFN will be more cost-efficient no matter how much the operational cost is. The selection of the bus journeys will stop at the saturation point when none of the remaining bus journeys are suitable anymore, and the proportion of the VFNs will stop increasing. In the suburb area, the percentage of VFNs does not change with the relative operational cost. This is because the selection of the buses has already reaches the saturation point at the first setting.

Impacts of traffic pattern: if we compare the number of fog nodes from Fig. 8a to 8d, combined with the mean of traffic flow in Fig. 6a to 6d, we can see that the traffic density is the main aspect to determine the fog node deployment decision. The number of fog nodes increases with the traffic density, because the fog nodes are more required at the times and places with higher demand.

D. Cost Estimation

Fig. 9 shows the comparison of the installation and operational costs of the three options. Following it, it compares the overall cost of *Option 1* and *Option 2*, and the saving potential of *Option 1* with different traffic patterns.

Installation cost: Fig. 9a shows how the installation cost changes with the unit installation cost for three deployment options. The installation cost of *Option 2* is the lowest, and that of *Option 3* is the highest. The difference of the cost between the options increases when the unit installation cost becomes higher. Compared to *Option 2*, *Option 1* has higher installation cost. This is due to the fact that while the CFNs keep providing stable computing service at their corresponding clusters, VFNs can only provide the computing service when they pass through the clusters along their driving routes. In order to substitute the capacity supply of a CFN, much more VFNs are required to present at the times and places with higher demand. Compared to *Option 3*, *Option 1* significantly reduces the installation cost. Therefore, it is more feasible to install the VFNs on part of the buses and schedule for them, instead of installing them on all the buses.

Operational cost: Fig. 9b shows the operational cost versus operational time using the three options. The operational cost of *Option 1* and *Option 3* coincide with each other, since the selection of the bus journeys are the same. Compared to *Option 2*, *Option 1* has lower operational cost, and the difference of the cost increases when the operational time becomes longer. The deployment of VFNs reduces the idle state of the computing resources, thus saves the operational cost compared to using CFNs only.

Overall cost: Fig. 9c shows the overall cost using *Option 1* and *Option 2*. Considering *Option 3* has much higher cost compared to the other options, it is not plotted. In the figure, the x-axis is the unit installation cost, and the y-axis is the operational time. The cost estimation of *Option 2* is represented as the blue plane of $z = 0$, and the cost estimation of *Option 1* is represented as the orange plane. In the shaded area, the z-value of the orange plane is positive, meaning that *Option 1* is more cost-effective than *Option 2*. When the unit installation cost becomes cheaper, or when the operational time becomes longer, *Option 1* will have higher potential for cost saving compared to *Option 2*. So VFC is more suitable

TABLE II: Service rates of different deployment options under various latency requirements in peak and off-peak scenarios.

	Off-peak Scenario			**Peak Scenario**		
$r_{network}$	50ms	100ms	150ms	50ms	100ms	150ms
Option i	84.0%	92.0%	98.0%	69.3%	88.7%	95.3%
Option ii	80.0%	92.0%	98.0%	60.0%	87.3%	94.7%
Option iii	88.0%	92.0%	98.0%	75.3%	92.7%	96.0%

for the cases where the operational cost has greater weight compared to the installation cost.

Long-term saving of operational cost: from the above analysis, we already know that the deployment of VFNs saves the operational cost at the expense of adding installation cost. In order to estimate the saving of operational cost between different times and places, we compared them in Fig. 9d. The results show the saving of operational cost is lager in the city downtown than the city suburb, and larger during the weekdays than the weekends. Additionally, when the operational time becomes longer, the saving of operational cost becomes more significant. Therefore, we can conclude that the saving potential of operational cost is greater in the times and areas with higher traffic density in the long run, due to the dense deployment of VFNs.

E. Service Provision

In this subsection we analyse the actual service rate, i.e. the percentage of users that can be served with a given deployment option. The network dynamics (e.g. available bandwidth, communication range, etc.) play a major role on the actual service rate of a deployment option. We used a VFC simulator to measure the service rate of different deployment options.

In our simulation, we assume that the execution time for vehicular tasks is negligibly small. Therefore, the latency constraints are modeled to reflect only the network latency. The time is discretized and divided into time slots, i.e. TTIs. The total simulation horizon is set to be 2000 TTIs where 1 TTI set to be 10 milliseconds. During each TTI, the positions of the client vehicles and buses are updated. We assume that the vehicles can have at most one active task. The air interface used in the simulation is 5G NR n78 with a 3500MHz frequency band and a 20MHz channel bandwidth. The dominant path model is used for estimating the signal-to-interference-plus-noise ratio (SINR) of the users at each TTI. Each user is assigned to the cell with the maximum SINR. To simulate different traffic scenarios, we consider two scenarios, one with 50 client vehicles (off-peak scenario) and another with 150 client vehicles (peak scenario). Three network latency requirements are set, namely 50ms, 100ms, and 150ms, and the service rate is measured under each requirement.

We compare the service rate of three deployment options:
Option i: Using both CFNs and VFNs with fixed network capacity (i.e. $cap_{network}$).
Option ii: Using only CFNs with fixed network capacity (i.e. $cap_{network}$).
Option iii: Using only CFNs with extended network capacity (i.e. $1.67 \times cap_{network}$).

Among the above options, *Option i* corresponds to VFC, while the *Option ii* and *Option iii* are used for comparison. The CFNs are co-located with 12 base stations, and the VFNs are carried by 8 bus journeys, so the overall network capacity is equal in *Option i* and *Option iii*.

The service rates of different deployment options under various network and traffic scenarios are shown in Table II. The presented results are averaged over 20 independent instances. Table II shows that the service rate is generally higher during the off-peak scenario than the peak scenario. The increasing latency constraint causes an average 20.6% drop in the service rate. Comparing different deployment options, *Option i* performs better (up to 9.3%) compared to *Option ii* and slightly worse (less than 6%) than *Option iii*.

Considering the relatively small performance difference between *Option i* and *Option iii*, it is possible to argue that the deployment strategy's applicability depends on the economical feasibility. From the economic perspective, upgrading the network capacity in both ways means additional investment in the infrastructure. However, the mobility of VFC allows smart scheduling of the resources. Therefore, VFC can save long-term operational costs and thus more cost-efficient than stationary deployment.

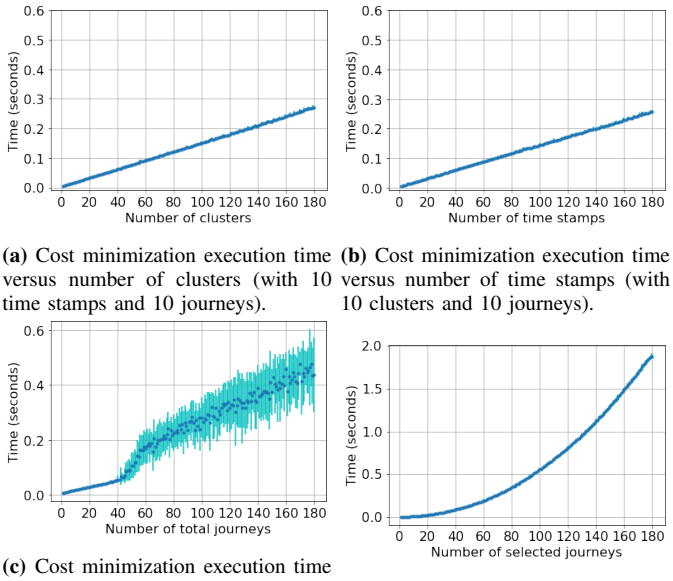
VIII. Discussion

In this section, we will discuss the computational complexity of the capacity planning model, the limitations of the current work and the future directions.

A. Computational Complexity

The demand estimation module can be solved in polynomial time using the algorithm explained in Section IV-C. The computational complexity of the cost minimization and the bus scheduling modules are evaluated for various data sizes. We use a commercially available computer equipped with an Intel Core i7-7700K at 4.2 GHz frequency CPU, where one thread out of eight is used during the measurements. The measurements are done for 20 independent instances and the presented results in Fig. 10 are averaged over this 20 instances. It can be seen that for the cost minimization module, the execution time increases linearly with the number of clusters and time stamps. And the execution time increases a bit faster than the linear relationship with the number of total journeys. For the bus scheduling model, the execution time increases quadratically with the number of selected journeys.

These linear and quadratic growth in complexity can be acceptable in real-world scenarios due to two reasons. First, the capacity planning is a long-term decision problem (i.e., in the order of months or years), therefore, there is no real-time constraints in the model. Secondly, even under stricter time constraints, it is possible to use high-power computing resources to decrease the execution time. In our experiments for Helsinki downtown area, the average execution time of demand estimation for each vehicular application combination is about 3 seconds. The average execution time of cost minimizing and bus scheduling are around 283 seconds and 9 seconds respectively when we consider 6 clusters, 1440 time stamps, and 5189 total journeys.



(a) Cost minimization execution time versus number of clusters (with 10 time stamps and 10 journeys).

(b) Cost minimization execution time versus number of time stamps (with 10 clusters and 10 journeys).

(c) Cost minimization execution time versus number of total journeys (with 10 clusters and 10 time stamps).

(d) Bus scheduling execution time versus number of selected journeys.

Fig. 10: Execution time versus size of data, where blue points represent the mean values, and the cyan lines represent the variations.

B. Limitations and Future Directions

In this work, we assume that the traffic in near future follows the same spatio-temporal distribution. Since the distribution may change in the future, especially when more autonomous vehicles are taken into use [19], we will update our traffic models with the ones that take into account the new changes in traffic patterns. For example, the regression-based solution we use can be replaced with deep learning based algorithms for predicting future traffic flows, such as Long Short-Term Memory network (LSTM) [20] and Graph Convolutional Network (GCN) [21]. And the capacity planning model will be updated accordingly to taken the uncertainty of the demand into consideration. Apart from this, we consider commercial fleets like buses to be the VFNs. In the future, the concept of VFN will be generalized. In another word, we will consider using taxis, drones, or other vehicles to serve as VFNs.

IX. Related Work

In this section, we listed the recent works of spatio-temporal traffic modeling, resource management and capacity planning in edge/fog environment, and compare them with our work.

The current research in traffic modeling focuses on finding the spatio-temporal patterns (i.e. traffic status, interaction among road segments, changing trend) of the daily vehicular traffic. Zhang et al. employed the dictionary-based compression theory to detect the anomaly behavior in road networks by analyzing the multi-dimensional traffic data [22]. Zhang et al. proposed a multi-agent system to analyze the spatio-temporal characteristics of the traffic data, as well as the cooperation and workflow among them [23]. Our work aims to quantify the spatio-temporal traffic flow and establish the traffic models that are embedded with the above-mentioned patterns for demand estimation.

For resource management in edge/fog computing environment, Sahni et al. proposed a data-aware multi-stage greedy adjustment algorithm to jointly schedule task and network flows to achieve low latency [24]. Gu et al. designed a distributed and context-aware task assignment mechanism to reduce overall energy consumption while satisfying the heterogeneous delay requirements [25]. Wang et al. proposed a latency-aware heterogeneous mobile edge computing system where the data that cannot be timely processed at the edge are allowed be offloaded to the upper-layer servers, and finally to the cloud center [26]. Mai et al. proposed a reinforcement learning approach that utilizes the evolution strategies for real-time task assignment among fog nodes to minimize the total latency during a long-term period [27]. These works are focused on managing the stationary resources to meet the computing demand with lower latency. However, in VFC, the problem becomes more challenging due to the mobility of the vehicles including the ones generating computing demand as well as the ones carrying the computing resources.

For resource management in VFC environment, Zhu et al. proposed a latency and quality optimized task allocation solution where a dynamic task allocation framework is built to adapt to the mobility of VFNs [7]. Shi et al. developed a deep reinforcement learning based algorithm for maximizing both the expected reward and the entropy of policy, while simultaneously evaluating the service availability of the vehicles that are incentivized to be the VFNs [28]. Zhou et al. proposed a two-stage VFC framework which consists of a contract theory based vehicular computational resource management mechanism, and a matching-learning based task offloading mechanism [29]. The works above are more focused on task allocation strategies based on latency or quality requirements, whereas our work is more focused on where to deploy the fog nodes and how much computing capacity is required to meet the demand.

For the capacity planning in edge/fog computing environment, Chiu et al. proposed an ultra-low latency cooperative task computing algorithm to simultaneously decide the number of fog nodes with proper communication resource allocation and computing task assignment [30]. Zhang et al. proposed the planning of fog computing networks that incorporate fog nodes planning, resources allocation, and offloading strategies to optimize the trade-off between the capital expenditure and the network delay [31]. Haider et al. proposed a mathematical model to simultaneously determine the optimal location, the capacity, the number of fog nodes, as well as the connection between the fog nodes and the cloud to minimize the delay in the network and the traffic to the cloud [32]. Stypsanelli et al. proposed an optimal capacity planning solution of fog computing infrastructures under probabilistic delay guarantees aiming to save the energy and operations costs [33]. Noreikis et al. proposed a capacity planning solution for edge computing that satisfying the QoS requirements while minimizing the number of required edge computing nodes [8]. The works above are also focused on the fog nodes deployment to meet the QoS requirement while minimizing the cost or amount of resources. However, they cannot be applied to VFC, since they did not consider the spatio-temporal dynamics of vehicular traffic.

For capacity planning in Vehicular environment, Hussain et al. proposed an Integer Linear Programming model for calculating the optimal location and capacity of fog nodes towards minimal overall network delay and energy consumption [34]. It focused on capacity planning of network resources instead of computing resources, and considered only stationary deployment of fog nodes. Premsankar et al. proposed a mixed integer linear programming formulation to minimize the deployment cost of edge devices by jointly satisfying a target level of network coverage and computational demand of vehicular applications in smart cities [35]. Our work differs from their work from two perspectives. Firstly, our work considers the deployment of both stationary and mobile fog nodes, rather than stationary ones alone. Secondly, we follow the data-driven approach and use real-world data for capacity planning, whereas they used random vehicular traces generated by a traffic simulator and synthetic application profiles.

X. Conclusion

This work proposes a data-driven capacity planning framework that optimizes the deployment of stationary and mobile fog nodes. Taking into account the spatio-temporal changes of demand, the installation and operational costs are minimized under the QoS requirements. The spatio-temporal distribution of vehicular traffic is modeled, the dynamic computing resource demand is estimated based on the traffic model and the resource consumption of the vehicular applications, and integer linear programming is used to find the cost-optimal solution. Real-world vehicular traffic data and vehicular applications are used to validate the proposed framework. Compared with the solution that only deploys fog nodes on base stations, the experimental results prove the potential to reduce costs by deploying fog nodes on cellular base stations and buses. The results show the deployment of mobile fog nodes saves operational costs at the expense of additional installation costs. Moreover, in the long run, more operational costs will be saved in the times and areas with higher traffic density due to the dense deployment of VFNs.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825496 and No. 815191, and Academy of Finland under grant number 317432 and 318937.

References

[1] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *2017 Global Internet of Things Summit (GIoTS)*, 2017, pp. 1–6.

[2] L. Castiglione, P. Falcone, A. Petrillo, S. Romano, and S. Santini, "Cooperative intersection crossing over 5g," *IEEE/ACM Transactions on Networking*, vol. 29, no. 01, pp. 303–317, Jan. 2021.

[3] M. Atagoziyev, K. Schmidt, and E. Schmidt, "Lane change scheduling for autonomous vehicles," May 2016.

[4] M. Chiang, B. Balasubramanian, and F. Bonomi, *Fog for 5G and IoT*, 1st ed. Wiley Publishing, 2017.

[5] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[6] 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.

[7] 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.

[8] M. Noreikis, Y. Xiao, and A. Ylä-Jaäiski, "Qos-oriented capacity planning for edge computing," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[9] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *2017 IEEE International Conference on Pervasive Computing and Communications: Workshops (PerCom Workshops)*. Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2017, pp. 6–9. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/PERCOMW.2017.7917508

[10] 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.

[11] R. Molina-Masegosa and J. Gozalvez, "Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.

[12] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.331, April 2017, version 14.2.2.

[13] H. Huawei, "R1-1812210: System-level evaluations on sidelink for nr v2x." 3GPP TSG RAN WG1 95, Spokane, USA, Nov. 2018.

[14] L. Elefteriadou, *An Introduction to Traffic Flow Theory*, Jan. 2014, vol. 84.

[15] S. Bunte and N. Kliewer, "An overview on vehicle scheduling models," *Public Transport*, vol. 1, pp. 299–317, Nov. 2010.

[16] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

[17] D. Gupta and R. J. wala, "Image Segmentation Keras: Implementation of Segnet, FCN, UNet, PSPNet and other models in Keras," Dec. 2020.

[18] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[19] B. Friedrich, *The Effect of Autonomous Vehicles on Traffic*, May 2016, pp. 317–334.

[20] X. Di, Y. Xiao, C. Zhu, Y. Deng, Q. Zhao, and W. Rao, "Traffic congestion prediction by spatiotemporal propagation patterns," in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, 2019, pp. 298–303.

[21] Q. Xie, T. Guo, Y. Chen, Y. Xiao, X. Wang, and B. Zhao, ""how do urban incidents affect traffic speed?" a deep graph convolutional network for incident-driven traffic speed prediction," Dec. 2019.

[22] Z. Zhang, Q. He, H. Tong, J. Gou, and X. Li, "Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 284–302, Oct. 2016.

[23] H. Zhang, Y. Zhang, Z. Li, and D. Hu, "Spatial-temporal traffic data analysis based on global data management using mas," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 267–275, 2004.

[24] Y. Sahni, J. Cao, and L. Yang, "Data-aware task allocation for achieving low latency in collaborative edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3512–3524, 2019.

[25] B. Gu, Y. Chen, H. Liao, Z. Zhou, and D. Zhang, "A distributed and context-aware task assignment mechanism for collaborative mobile edge computing," *Sensors*, vol. 18, p. 2423, July 2018.

[26] P. Wang, Z. Zheng, B. Di, and L. Song, "Hetmec: Latency-optimal task assignment and resource allocation for heterogeneous multi-layer mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4942–4956, 2019.

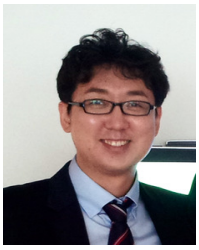
[27] L. Mai, N.-N. Dao, and M. Park, "Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latency minimization in fog computing," *Sensors*, vol. 18, p. 2830, Aug. 2018.

[28] J. Shi, J. Du, J. Wang, J. Wang, and J. Yuan, "Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.
[29] Z. Zhou, H. Liao, X. Wang, S. Mumtaz, and J. Rodriguez, "When vehicular fog computing meets autonomous driving: Computational resource management and task offloading," *IEEE Network*, vol. 34, no. 6, pp. 70–76, 2020.
[30] T. Chiu, W. Chung, A. Pang, Y. Yu, and P. Yen, "Ultra-low latency service provision in 5g fog-radio access networks," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2016, pp. 1–6.
[31] D. Zhang, F. Haider, M. St-Hilaire, and C. Makaya, "Model and algorithms for the planning of fog computing networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3873–3884, 2019.
[32] F. Haider, D. Zhang, M. St-Hilaire, and C. Makaya, "On the planning and design problem of fog computing networks," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2018.
[33] I. Stypsanelli, O. Brun, S. Medjiah, and B. J. Prabhu, "Capacity planning of fog computing infrastructures under probabilistic delay guarantees," in *2019 IEEE International Conference on Fog Computing (ICFC)*, 2019, pp. 185–194.
[34] M. M. Hussain, M. Alam, and M. M. Beg, "Vehicular fog computing-planning and design," *Procedia Computer Science*, vol. 167, pp. 2570–2580, 2020.
[35] G. Premsankar, B. Ghaddar, M. Di Francesco, and R. Verago, "Efficient placement of edge computing devices for vehicular applications in smart cities," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–9.

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.

Ozgur Umut Akgul is currently a Postdoctoral researcher with the Department of Communications and Networking, Aalto University. He received B.S. in electronics enginnering 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.

Abbas Mehrabi Abbas Mehrabi is currently a Lecturer in the department of computer and information sciences at Northumbria University, Newcastle upon Tyne, UK since May 2020. He was a Lecturer in computer science at Nottingham Trent University, UK during 2019-2020. From 2017 until 2019, he was a Postdoctoral researcher in distributed and mobile systems research group in the computer science department at Aalto University, Finland. He received his Ph.D from the school of electrical engineering and computer science at Gwangju Institute of Science and Technology, South Korea in 2017 and his M.Sc and B.Sc both in computer engineering from respectively Azad University, Tehran and Shahid Bahonar University of Kerman, Iran. His research interests include mobile cloud/edge computing, Internet of Things, vehicular networking and smart grid communications. He is a member of IEEE and Associate Fellow of UK higher education academy (HEA).



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.

Antti Ylä-Jääski Antti Ylä-Jääski received the Ph.D. degree from ETH Zürich in 1993. From 1994 to 2009, he was with Nokia in several research and research management positions, with a focus on future Internet, mobile networks, applications, services, and service architectures. Since 2004, he has been a Tenured Professor with the Department of Computer Science, Aalto University. His current research interests include mobile cloud computing, mobile multimedia systems, pervasive computing and communications, indoor positioning and navigation, energy efficient communications and computing, and Internet of Things.