Driving in the Fog: Latency Measurement, Modeling, and Optimization of LTE-based Fog Computing for Smart Vehicles

Yong Xiao*, Marwan Krunz†‡, Haris Volos§, and Takashi Bando §
*School of Electronic Information and Communications, Huazhong Univ. of Science & Technology, Wuhan, China
†Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ
‡School of Electrical and Data Engineering, University Technology Sydney, Australia
§Silicon Valley Innovation Center, DENSO International America, Inc., San Jose, CA

***Abstract*—Fog computing has been advocated as an enabling technology for computationally intensive services in connected smart vehicles. Most existing works focus on analyzing and optimizing the queueing and workload processing latencies, ignoring the fact that the access latency between vehicles and fog/cloud servers can sometimes dominate the end-to-end service latency. This motivates the work in this paper, where we report a five-month urban measurement study of the wireless access latency between a connected vehicle and a fog computing system supported by commercially available multi-operator LTE networks. We propose *AdaptiveFog*, a novel framework for autonomous and dynamic switching between different LTE operators that implement fog/cloud infrastructure. The main objective here is to maximize the *service confidence level*, defined as the probability that the tolerable latency threshold for each supported type of service can be guaranteed. AdaptiveFog has been implemented on a smart phone app, running on a moving vehicle. The app periodically measures the round-trip time between the vehicle and fog/cloud servers. An empirical spatial statistic model is established to characterize the spatial variation of the latency across the main driving routes of the city. To quantify the performance difference between different LTE networks, we introduce the weighted Kantorovich-Rubinstein (K-R) distance. An optimal policy is derived for the vehicle to dynamically switch between LTE operators' networks while driving. Extensive analysis and simulation are performed based on our latency measurement dataset. Our results show that AdaptiveFog achieves around 30% and 50% improvement in the confidence level of fog and cloud latency, respectively.**

***Index Terms*—Fog computing, LTE, cloud computing, connected vehicle, low-latency, measurement study.**

I. Introduction

Low-latency, reliable communications, and processing are critical to newly emerging smart vehicular services such as congestion avoidance, accident prevention and active control intervention, autonomous driving, and intelligent driver assistance (e.g., route computation, searchable maps, etc.). Due to the limit of space, energy supply, processing and storage capabilities of the in-vehicle computer, connected smart vehicles supported by high-performance cloud data centers (CDCs) for data storage (e.g., high-definition map) and processing has been recently promoted by various industry consortiums and standardization bodies [1], [2]. The physical connectivity between vehicles and the CDC may span several wireless and wired links, each having its own traffic dynamics, medium access mechanisms, connection intermittency, etc. As a result, the end-to-end communication path may exhibit unacceptable latency and connection unreliability. This makes a strong case for seeking better solutions that are more suitable for low-latency and high-availability services. Fog computing has recently been introduced to enable network-edge computing, thus reducing the end-to-end latency [3]. Supporting smart vehicular applications via fog computing has the potential to significantly reduce the communication latency and improve service reliability [4], [5].

Fog computing-enhanced wireless system has recently been advocated by mobile network operators (MNOs) as a way to create new business opportunities, increase revenues, and reduce costs. Major MNOs, including AT&T, Verizon, and Deutsche Telekom have all announced plans to integrate fog computing into their network infrastructure to support emerging applications such as robotic manufacturing, autonomous cars, and augmented/virtual reality (AR/VR). LTE is readily available to support high-speed low-latency wireless solutions on a global scale, and therefore is in an excellent position to push for the maturity and large-scale deployment of fog computing. 3GPP recommends the round-trip-time (RTT) for user equipments (UEs) across LTE networks to be kept as low as 10 ms in optimal conditions [6], which is commonly considered to be negligible compared to other elements of latencies in the fog computing system such as processing and queueing latency. Unfortunately, recent reports as well as our own measurements suggest that the 10 ms latency requirement is too challenging to be achieved by most existing LTE networks. In fact, recent studies [7]–[9] observe that the wireless connection between moving vehicles and the LTE network can sometimes experience frequent disconnections, retransmission, and high wireless access latency that dominate the overall RTT between UEs and external servers at fog nodes as well as CDCs.

While there have been numerous studies on the wireless access latency throughout LTE networks, there is a noticeable lack of a long-term systematic study of latency modeling and optimization between moving vehicles and cloud/fog servers for practical LTE systems. In fact, due to the geographically varying network infrastructure deployment as well as different requirements and traffic dynamics of vehicular services, modeling and optimizing the latency in an LTE-based vehicular system is quite difficult.

This paper empirically analyzes the latency performance of vehicle-to-cloud/fog solutions for connected smart vehicular systems in a multi-operator LTE system. We propose a novel optimization framework, *AdaptiveFog*, for a vehicle to

dynamically switch between MNO networks that implement fog and cloud services on the move. We develop a smart phone app using Android API and place a Google Pixel 2 phone installed with our developed app in a vehicle to run a five-month measurement campaign on commercially available LTE networks deployed by two major MNOs throughout the main driving routes in a mid-sized city. These measurements are used to evaluate the impact of handover, driving speed, MNO network, fog/cloud server, and location on the service latency. We observe that the spatial variation (over different locations) of the latency performance across different MNO networks can be much more significant than the temporal variations (over different times of the day as well as days of the week). Accordingly, we investigate the confidence level of various connected vehicular service across a city-wide geographical area. An empirical spatial statistic model is established using the dataset collected in our campaign. We introduce the weighted Kantorovich-Rubinstein (K-R) metric to quantify the performance difference between MNO networks, taking into consideration of the heterogeneity of the demands and priorities of different services. We formulate the MNO selection and server adaptation problem as a Markov decision process and derive the optimal policy for a moving vehicle to switch between different MNOs' networks. Extensive simulations are also conducted to evaluate the performance of AdaptiveFog. Numerical results show that AdaptiveFog achieves around 30% and 50% improvement in the confidence level for fog and cloud latency, respectively, especially when being applied to vehicular applications with stringent latency requirement (e.g., active road safety applications) in existing LTE systems.

II. Related Work

The concept of the fog computing and its relation to other similar concepts such as cloud and mobile edge computing can be blurry in some contexts. For example, a cloud service provider can also deploy smaller-scale cloud computing infrastructure, i.e., fog servers, in some areas. In this paper, we use the term *fog computing* to refer to a generalized architecture that includes cloud, edge, and clients [3]. We also use the term *fog node* or *fog server* to denote the servers placed at the edge of the network. We use the term *cloud server* to denote the high-performance server installed at the CDC.

Fog Computing and Connected Vehicles: A fog node is considered as a cost-effective yet resource-limited computational device, especially compared to the CDC. Therefore, most existing works focus on developing new methods and architectures to improve the utilization of fog resources with reduced costs. For example, Tong *et al.* [10] proposed a hierarchical architecture to improve the resource utilization throughout a fog computing system. Yu *et al.* [11] considered the application provision problem with bandwidth and delay requirements in a fog computing-enabled Internet-of-Things (IoT) system. Garcia-Saavedra *et al.* [12] proposed an analytical framework, called FluidRAN, that minimizes the aggregated operator expenditure by optimizing the design of the virtualized radio access network. Inaltekin *et al.* [13] introduced an analytical framework to derive the optimal location of the virtual controller for balancing latency and reliability in a fog computing system.

Fog computing-supported connected vehicle has recently been promoted by both industry and standardization bodies as a key enabler for emerging smart vehicular applications, such as intelligent driver-assistance and autonomous driving [1], [14]. Premsankar *et al.* [4] studied the placement of edge computing servers for vehicular applications. An effective heuristic method was proposed to deploy fog servers based on the knowledge of road traffic within each deployment area. Lee *et al.* [15] proposed an in-kernel TCP scheduler to mitigate the network latency of connected vehicles with redundant transmission.

Performance Evaluation and Wireless Network Analysis: There have been quite a few studies on the performance of vehicular networks supported by a wireless infrastructure. For instance, Bedogni *et al.* [16] analyzed a real-world GPS trajectory dataset to investigate the temporal topology of vehicle-to-vehicle (V2V) networks. In [5], Asadi *et al.* studied beam selection for 5G mmWave-based vehicular-to-infrastructure (V2I) communications. An online learning algorithm with environment-awareness was developed and shown to approach the near-optimal performance.

In [8], Hameed Mir *et al.* compared the performance of IEEE 802.11p and LTE for vehicular networking using NS3 simulations. Simulation results show that LTE offers much better network capacity and mobility support compared to IEEE 802.11p. Xu *et al.* [9] conducted extensive real-world testing for multiple smart vehicular application scenarios. The results suggest that existing LTE systems are not recommended for active road safety applications with high-data rate and real-time requirements, such as collision avoidance. It is however, sufficient to support non-safety applications including traffic updates, file download, and Internet access. In [7], Hadzic *et al.* investigated the latency between a fixed mobile station and an LTE-based fog computing system. The authors conducted both in-lab testing using an isolated base station with controlled parameters as well as real-world evaluation on a commercial LTE system. The results reveal that the wireless connection between the UE and the base station introduces irreducible and non-negligible latency for delay-sensitive fog computing applications.

Our Contribution: To the best of our knowledge, this is the first work that focuses on modeling and optimizing the latency performance of LTE-based fog computing systems based on a long-term city-wide measurement. We introduce a novel distance metric, referred to as weighed K-R distance, to quantify the difference of latency probability distributions between different LTE networks. Accordingly, we derive the optimal policy for selecting an LTE provider and fog/cloud server when driving through different regions. Our solution is simple and comprehensive, and can be applied to more general scenarios with other choices of wireless access technologies and computational resources.

III. Architecture Overview

We consider a fog computing-supported connected vehicular system consisting of the following main elements:

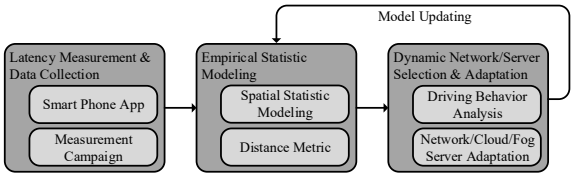

Fig. 1. Main components of AdaptiveFog.

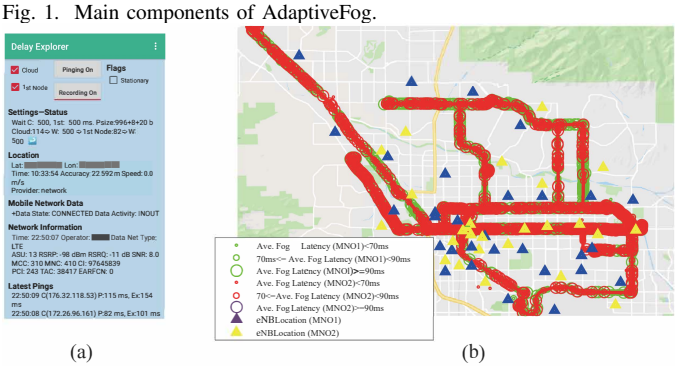

Fig. 2. (a) User interface of *Delay Explorer* developed specifically for our latency measurement and (b) measuring routes and traces in our study.

UE: corresponds to a moving vehicle installed with the over-the-top applications that can generate computational intensive workload requests that exceed the capability of the vehicle's onboard computers/processors. It can also be smart devices such as smart sensors, mobile phones, and laptops located in the vehicle.

LTE Networks: provide wireless links connecting the UE to fog nodes and the cloud server. In this paper, we consider multi-operator LTE connections in which the UE can switch to different MNO networks for submitting its workload request and receiving the processing result. For example, dual-SIM smart phones that can take two SIM cards from two MNOs are already on the market. In addition, Google's Project Fi-enabled smart phones also have the capability to switch between networks of multiple MNOs.

Fog Nodes: correspond to low-cost mini-servers deployed at the edge of the network to support low-latency services for connected smart vehicles.

Cloud Server: corresponds to the expensive high-performance servers deployed by the CDC to provide on-demand computational service for the UE.

IV. Methodology

We propose *AdaptiveFog*, a simple framework for the UE to dynamically switch between available MNO networks and cloud/fog servers on the move. It consists of three main components: trace collection, empirical modeling, and network adaptation, as illustrated in Figure 1.

A. Trace Collection

Smart Phone App Design: We begin by collecting traces to measure the network performance between a commercial off-the-shelf smart phone and the most likely fog node location as well as a CDC server. For some safety-related applications,

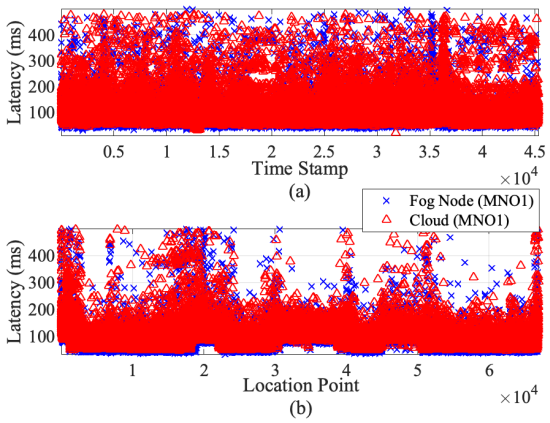

Fig. 3. Traces ranked by (a) different times in a fixed location throughout a full-week of measurement (b) different location points.

latency is a more important performance metric than the throughput. Instead of measuring the network bandwidth, we evaluate the RTT. We develop a smart phone app, called *Delay Explorer* using Android API to periodically ping the IP address of the most likely fog node location and an arbitrary IP address of the closest Amazon cloud server and record the resulting RTT at every 500ms. We have removed the measurements corresponding to the RTTs that are larger than 500ms. In addition to record the RTT, Delay Explorer also records other vehicle-related information such as time stamp, GPS coordinates, altitude, driving speed, as well as network-related information including network connection type, bearing, ASU, RSRP, RSRQ, etc., as shown in Figure 2(a). Delay Explorer only records RTT when it connects to the LTE networks and will stop recording if LTE connection is dropped.

Measurement Campaign: We ran a five-month city-wide measurement campaign with a Google Pixel 2 smart phone installed with Delay Explorer. For the first month, the phone has been placed at multiple fixed locations across a university campus and a residential area for continuous recording. It has then been placed at a vehicle for driving measurement for the rest 4 months (See Figure 2(b) for the measuring routes). For each MNO, the UE records two types of latency: 1) *Cloud latency*, that is the RTT recorded from pinging the IP address of the CDC server and 2) *fog latency*, which corresponds to the RTT recorded from pinging the first hop IP address in each MNO's LTE core network. For the driving measurements, the vehicle records latency traces on both working days and weekends and the driving time in each working day is approximately 2 hours. We have collected over 300,000 traces from each MNO's network when driving at the main routes throughout a mid-sized city.

Results and Discussion: In Figure 3(a), we present the traces recorded at a fixed lab location for a one-week continuous measurement to evaluate the impact of the time-of-measurement on the RTT. We found that there are generally no observable correlation between the RTT and the time of measurement. This result is consistent with a recent study in a similar sized city [15]. We then aggregate all the

TABLE I
LATENCY PERFORMANCE OF TWO MNO NETWORKS

Traces			L1 Fixed	L2 Fixed	All Fixed	R1(Drive) (6.1m/s)	R2(Drive) (15.7m/s)	All Drive
MNO1	Fog Latency (ms)	Mean	62	72	70	83	96	88
		STD	18	16	18	28	29	34
		Median	55	71	68	77	91	85
		Conf. 90%	85	86	85	115	121	120
	Cloud Latency (ms)	Mean	74	87	85	94	108	96
		STD	15	15	21	26	29	33
		Median	71	88	86	92	108	94
		Conf. 90%	88	100	104	124	129	128
MNO2	Fog Latency (ms)	Mean	72	64	72	85	80	83
		STD	14	17	15	52	46	51
		Median	71	93	71	69	67	66
		Conf. 90%	84	87	86	132	112	131
	Cloud Latency (ms)	Mean	87	74	88	119	125	124
		STD	13	13	17	50	47	54
		Median	88	71	90	108	117	109
		Conf. 90%	99	87	102	166	133	100

traces collected in a major route from the four-month driving measurements and rank the traces by the location points in Figure 3(b). We can observe noticeably different patterns in some locations than others. In other words, compared to the time of measurements, the geographical heterogeneity contributes more to the diversity of the statistics of RTT. We summarize the latency performance of traces collected from our measurement campaign in Table I. We present the mean, standard deviation (STD), and median values of all the traces for our fixed location and driving measurements as well as the RTT for two fixed locations (L1 and L2) as well as two major driving routes (R1 and R2 with average driving speeds $6.1m/s$ and $15.7m/s$, respectively). It can be observed that RTTs of different MNO networks can vary significantly in some locations/driving routes. When taking into consideration of all the traces, both MNOs exhibit similar latency performance in terms of mean and STD values. However, the driving traces of both MNO networks show more significant differences in terms of STD, mean, and median values. One of the main reasons causing this result is that, the eNB deployment densities and locations of our considered MNOs are quite different as shown in Figure 2(b). We will give a more detailed discussion about the issues that can affect the latency of a connected vehicular system in Section V.

B. Model Evaluation

Weighted Confidence: Most latency-sensitive applications do not differentiate the latency performance as long as the resulting RTT is below the a tolerable threshold. For example, it has been reported in [8] that for *active road safety applications* such as collision avoidance, emergency alert and active control intervention for crash prevention, the maximum tolerable service latency is 100ms. For *cooperative traffic efficiency applications* intended to provide additional information exchange and coordination for improving the traffic flow and enhancing the traffic coordination such as traffic congestion relief and flow control, less than 200ms of latency is considered as sufficient. For *infotainment applications* such as video/audio streaming, up to 500ms of latency is considered as tolerable.

We therefore consider the *proportionally weighed confidence level* as the main performance metric to evaluate the latency of each individual MNO network. More formally, suppose the UE can support a set of service types, denoted as $\mathcal{M}$, each has its own maximum tolerable latency denoted as r_i for service type i. The confidence level F_i of service type i is the probability that the maximum tolerable latency r_i can be satisfied, i.e., we have $F_i = \Pr(x \leq r_i)$.

It can be observed that the confidence level is a more realistic and useful performance metric, especially compared to the average and minimum latency because for most vehicular applications, it is critical to quantify the chances that a certain latency threshold can be guaranteed by the wireless system.

Different types of services can have different probability of being requested as well as priorities to be served. For example, cooperative traffic efficiency applications may be requested more often in low-speed traffic congestion area compared to the active road safety applications. Also, the active road safety applications should always be assigned with a higher priority compared to the infotainment applications. To include these factors into latency performance analysis, a weighting factor w_i can be assigned to each service type i and the proportionally weighed confidence level is the aggregated confidence levels with all the supported services being served at their corresponding tolerable latency thresholds given by

$$\hat{F} = \sum_{i \in \mathcal{M}} w_i F_i. \tag{1}$$

Note that (1) is a general performance metrics that can be applied to a wide range of applications under various scenarios. For example, suppose the probability of receiving type i service request is given by $\Pr(\lambda = i)$ for $i \in \mathcal{M}$. In this case, if we set $w_i = \Pr(\lambda = i)$, then $w_i F_i$ is equivalent to the probability that a service type being requested by the UE can also be served with the satisfied latency performance.

Distance Metric: To quantify the difference between the latency performance offered by different MNOs, we introduce the weighted Kantorovich-Rubinstein (K-R) metric which is defined as

$$K(F, G) = \sum_{i \in \mathcal{M}} w_i \left[F_i - G_i\right], \tag{2}$$

where F_i and G_i correspond to the two empirical cumulative distribution functions (CDFs) of latency traces recorded in two different MNO networks.

The weighted K-R distance in (2) corresponds to the weighted difference between the confidence levels of different services at their maximum tolerable thresholds. Generally speaking, the UE should always choose the LTE network that provides a higher confidence level to achieve a better service performance guarantee. However, there is a cost for switching between LTE networks. This cost can be caused by the price difference between MNO's networks, extra latency for the UE to disconnect from one MNO and reconnect to another, and/or extra energy and processing resource consumed during the switching. Therefore, the UE needs to not only consider the current performance of each MNO but also the performance that can be offered by the MNOs in the future, i.e., the UE should choose a single MNO or a sequence of MNOs to maximize the confidence of maintaining guaranteed services with the minimized cost incurred by switching back-and-forth between MNO's networks.

The weighted K-R distance is a useful metric for the UE to decide whether to switch to another MNO's network. We will give a more detailed discussion in Section VII.

Model Updating: The probability distribution of the latency in some specific locations can change over time, e.g., due to road work and/or traffic accidents. In this case, the UE should be able to detect the change and adjust the empirical PDF according to the updated latency traces. There are many existing approaches [17] can be applied to detect the change of empirical PDF using updated samples. Applying and comparing the model/statistic-changing detection methods into AdaptiveFog is out of the scope of this paper and will be left for our future research.

C. Network/Server Selection and Adaptation

Driving Behavior Modeling: In addition to the performance of the physical network infrastructure, the latency performance of the UE can also depend on many human-related factors such as the driving routine, habit, and behavior of the driver. It has been verified that the driving location and speed of a vehicle typically follow the Markov property, that is the future state of the vehicle including the location and speed only depends on the current state. We apply the driving location and speed data collected in our measurement campaign to calculate the empirical state transition probability of the UE when driving through different locations with different speeds.

Network Adaptation: The main objective is to maximize the long-term confidence level minus the possible cost incurred by switching between LTE networks while the UE is driving through different locations. We consider a slotted decision making process and assume in each time slot t, the UE can only choose one MNO's network. We abuse the notation and use k to denote both the selected MNO as well as its LTE network. We also use j to denote the fog or cloud server selected by the UE. As will be shown in Section VI, the cloud latency is generally lager than the fog latency. However, a cloud server has much more computational resources compared to the fog server and therefore can still be considered as the preferred choice of workload outsourcing if the latency requirement is not stringent. We write the utility obtained by the UE in time slot t as

$$u_t(k_t, j_t) = \sum_{i \in \mathcal{M}} w_i F_{i,t}\left(s_t, k_t\right) - \mathbf{1}\left(k_t \neq k_{t-1}\right) c \qquad (3)$$

where we use subscript t to denote the parameters in time slot t. $\mathbf{1}(\cdot)$ is the indicator function, c is the cost of switching between LTE networks, s_t is the state information including the location and speed of the UE, and $F_{i,t}(s_t, k)$ is the confidence level at r_i in state s_t with MNO k being selected by the UE.

We consider a slotted decision making process with infinite horizon. The optimal policy for the UE to select the optimal MNO and fog/cloud server for a given service time duration T is given by

$$\pi\left(\langle k, j\rangle\right) = \arg\min_{\langle k,j\rangle} E\left(\lim_{T\to\infty} \sum_{t=1}^{T} \gamma^t u_t\left(k, j\right)\right), \qquad (4)$$

where $0 < \gamma \leq 1$ is the discount factor specifying how impatient the UE is, i.e., the smaller the γ the more the UE cares about the latency performance in the current time slots than the future.

V. Latency Analysis in LTE-based Fog Computing

The RTT between the UE and the fog node for an LTE-supported fog computing network can be affected by the following factors:

Fog Node Placement: Most existing works assume that simply deploying fog servers at the eNB (LTE base station) location can achieve a negligible RTT between the UE and the fog server [1], [18], [19]. However, as observed in [7], eNBs are typically installed at inaccessible locations (e.g., the top of a hill or open spaces such as lamp posts and street cabinets) and therefore cannot offer sufficient space and resources (e.g., electric power and cooling load) for servers. In addition, allowing the workload submitted by the UE to be redirected to a co-located server at the eNB instead of being forwarded to the LTE core network, i.e., ePC, via S1 interface will also require a total redesign of LTE interfaces. In a real LTE system, data packets of the UE will be passing through many IP routing hops within the ePC. Unfortunately, these internal IP hops have been hidden from the public access. The UE can only get a private subnet IP address that is translated to a public address at the P-GW. In fact, in our measurement, we observe that, in each MNO's network, the first hop IP address identified by "traceroute" remains the same across different cities. This is typically for IPv4-based networks where IP addresses are scarce. In this case, an Internet-based application at the UE perceives the entire ePC as a single routing hop. To minimize the RTT between the UE and the fog node, the fog node should be placed close to the first public IP address also referred to as the first node in the ePC that can be identified.

Uplink Latency: We consider the scenario that the UE submits its workload using the data-only best-effort service offered by the MNOs. In this case, the UE must first initiate the uplink data transmission by submitting a one-bit scheduling request (SR) to the physical uplink control channel (PUCCH) informing the eNB about the new packet arrivals. The UE will then wait for the eNB to schedule a grant that specifies the radio resources for uplink transmission. If the UE does not receive the uplink resources from the eNB, it will resend the SR on PUCCH based on the SR periodicity from 5ms to 80ms (In LTE Release 9, new 1ms and 2ms SR periodicities have been added.) [20].

Downlink Latency: The eNB will feedback the processing result to the UE when it is available. In LTE-FDD, a 1ms subframe is considered to be the typical wireless transmission time interval between the UE and the eNB. This pluses the frame alignment time (typically 0.5ms), UE processing latency (1.5ms). In case that the result delivery fails, the UE will feedback a negative acknowledgment (NACK) after 4 subframes and the Hybrid ARQ (HARQ) retransmission occurs 4 subframes after receiving the NACK resulting a total 8ms of delay.

Handover: One of the main factors that cause service interruption, drop of connection, and increased latency for the

UE when driving is the handover, i.e., the UE as well as its connected service is transferred from one cell/eNB to another. The handover decision is typically initiated by the UE via its connected eNB when its measured downlink signal power from its serving eNB is below a certain threshold. In particular, the UE starts measuring the signal strength of a neighboring eNB when the received signal power of the current eNB is below a threshold value. The UE will then report the result to the source eNB. Since the signal measuring and neighboring cell searching are made by the UE even in the idle state (during DRX periods), the latency of cell searching and identification is typically assumed to be negligible. Once the downlink measurement results reported from the UE satisfies a certain condition, the source eNB will initiate the handover process by sending radio resource control (RCC) reconfiguration message to the UE which specifies the identity of the target eNB. According to [21], the maximum allowed delay for RCC reconfiguration is 15 ms. The source eNB will also send a handover request message to the target eNB. Once received the request, the target eNB will allocate the resources in the target cell and allocate a new Radio Network Temporary Identifier (RNTI) to the UE. The handover can be based on the S1 interface between two eNB without requiring coordination through the higher level components such as MME and P-GW. When S1 interface is unavailable, the handover will be processed by the MME via X1 interface. From the UE's perspective, it is impossible to differentiate these two types of handover. In fact, it is generally impossible for the UE to tell which handover procedure has been executed.

VI. Empirical Modeling

A. Cloud vs. Fog Latency

Latency and Reliability Tradeoff: We present the histogram as well as the empirical PDF of cloud and fog latency measured in a fixed lab location in Figures 4 and 5, respectively. It can be observed that the PDF of fog latency follows the dual modal with the first and second peaks at around 54ms and 87ms, respectively. The 33ms difference between these two peaks is mainly caused by the SR retransmission periodicity (around 20 to 40ms) and the HARQ retransmission delay (around 1 to 8 ms). Note that, in [7], the authors observed a sawtooth RTT pattern caused by the SR retransmission periodicity at every 20ms with around 40ms amplitude in a fixed lab location. Since our latency traces are recorded at every 500ms, we did not observe any strong sawtooth pattern in our dataset. However, the SR retransmission still contributes to the second peak of the latency traces. From Figures 4 and 5, we can observe that the Internet connection between the LTE network and the cloud server contributes to approximately 10ms over the overall RTT of the UE. It is interesting to observe that for most of the latency traces, the standard deviation of the cloud latency is less than that of the fog node. This means that the extra delay and connection variation of the Internet compensates the latency variation of the wireless links between the UE and the ePC. The above observation also verifies the recent study reported in [13] where the authors suggest that although the cloud server normally has higher average latency compared to the fog node, the service latency between the UE and cloud offers lower uncertainty, i.e., less standard deviation, compared to that between UE and fog node.

In Figures 6 and 7, we compare the empirical PDF generated from the cloud and fog latency traces. We observe that the mobility of the UE contributes to around 10 to 20ms in average for the latency, compared to the fixed location. More importantly, the driving latency traces show a significantly increase in the variance of the RTT, i.e., around 30ms to 40ms increase for the 90 percentile of the empirical PDF for fog node and cloud latency, respectively. This is caused by handover, data loss, and reconnection which will be discussed in more details in the rest of this section.

Cloud/Fog Server Selection and Adaptation In Figures 8 and 9, we compare the CDFs of fog node and cloud latencies and compare their K-R distance under various latency thresholds, e.g., 50ms and 100ms, with the weighting factor set to be 1. We can observe that for fixed-location latency traces, the minimum K-R distance between cloud and fog latency is at 85ms in which the difference between two CDFs is only 0.23%. In other words, if the UE's applications cannot differentiate the service quality as long as the latency is controlled below 85ms, offloading the workload to cloud or fog node will not cause much noticeably different latency performance. However, for the applications that are sensitive to the latency below 85ms, the fog node will offer much better performance than the CDC. In particular, if the maximum tolerable latency of the UE is at 63ms, the difference between the confidence interval of cloud and first node to meet the required latency requirements will be as high as 58.6%.

For the driving latency traces, we observe that the difference between cloud and fog node becomes less compared to that of the fixed location. In particular, the minimum K-R distance is at 74ms with only 0.55% difference between the confidence level of cloud and fog latency. The maximum K-R distance is at 101ms where switching from cloud server to fog server can result in over 16.5% increase of confidence level. This means that the uncertainty of wireless connection plays a more dominant role in our driving latency traces, compared to the fixed location data set.

Fog server typically has much less computing power compared to the cloud server. Therefore, most existing works suggest to only offload the most latency-sensitive applications to fog nodes and leave the more delay-tolerant service workload to the CDC. Our observation here suggests that the K-R distance offers more specific decision threshold for identifying the services that should be submitted to CDC or fog nodes. In particular, for a given LTE network and maximum tolerable delay r_i, we can write a simple threshold-based policy for selecting fog or cloud server to process each service type i as

$$j = \begin{cases} \{\text{Cloud Server}\}, & \text{If } K(G_i, G_i') \leq \theta_f, \\ \{\text{Fog Server}\} & \text{Otherwise,} \end{cases} \quad (5)$$

where G_i and G_i' is the empirical CDFs of cloud and fog latency at value r_i and θ_f is the threshold specifying the difference between tolerable confidence levels of fog node

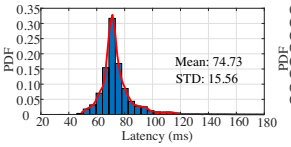

Fig. 4. Empirical PDF of cloud latency in a fixed location.

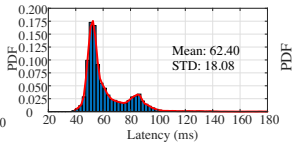

Fig. 5. Empirical PDF of fog latency in a fixed location.

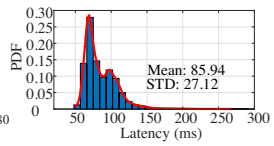

Fig. 6. Empirical PDF of cloud latency when driving.

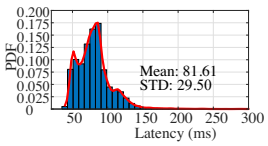

Fig. 7. Empirical PDF of fog latency when driving.

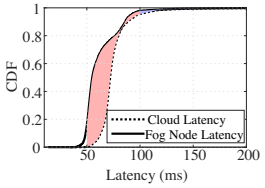

Fig. 8. Latency CDF in a fixed location with latency thresholds at 50ms, 100ms, 150ms.

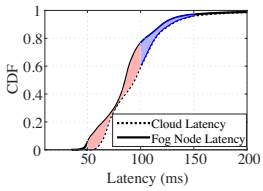

Fig. 9. Latency CDF when driving with thresholds 50ms, 100ms, 150ms.

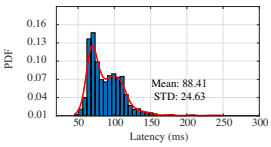

Fig. 10. Empirical PDF of MNO 2's cloud latency when driving.

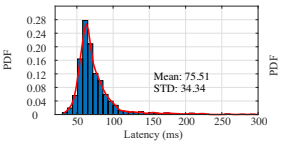

Fig. 11. Empirical PDF of MNO 2's fog latency when driving.

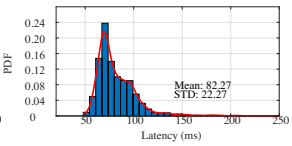

Fig. 12. Empirical PDF of cloud latency (MNO 1) in a multistory parking lot.

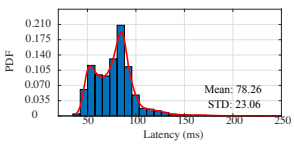

Fig. 13. Empirical PDF of fog latency (MNO 1) in a multistory parking lot.

and cloud latency that can be considered to be negligible for service type i.

B. Different MNOs

It is known that the service latency offered by different MNOs exhibit significant spatial variation depending on the location and eNB deployment densities at each area. To investigate the possible causes of the different latency performances of MNOs, we need to look into specific areas. In particular, in Figures 12-15, we present the empirical PDF of the RTT measured at the first level of a multi-story parking lot for both MNO's networks. We can observe that for MNO 1, the RTT in the parking lot results suffer around 10ms increase in the average compared to the RTT of the office in Figures 4 and 5. This can be caused by the higher chances of HARQ retransmission and in-synchronization. The RTT offered by MNO 2 however experience a much higher noticeable increase in both terms of average latency as well as the standard deviation due to the less dense deployment of the eNB in the surrounding area compared to the MNO 1. Another reason causing the performance degradation of MNO 2 is that the LTE network of MNO 2 in the local area operated at 1900 MHz. MNO 1 however operates at a lower frequency band (850MHz) which can have better penetration through concrete wall. This will also increase the chance of packet loss, in-synchronization, connection drop, and retransmission.

In Figures 16 and 17, we compare the CDFs of the cloud and fog nodes latency offered by the two MNOs. For the fog latency, we observe that if the latency constraint of the UE is at 88ms, then difference of the confidence levels offered by two MNOs reaches the maximum value at 25.79%. Also MNO 2 offers higher confidence level for services with the maximum tolerable latency below 131ms. The fog latency offered by two MNOs provide the same confidence level at 64ms and 125ms. The maximum difference between the fog latency confident level is at 80ms. In this case, MNO 2 offers 29.91% higher confidence level than MNO 1.

C. Handover

To investigate the impact of the handover on the latency performance, we present the empirical PDF of the RTT when the UE is driving between two eNB in a open straight route outside of the city center in Figures 18 and 19. Since it is impossible to identify the exact location/timing of the handover, i.e., handover can even happen after the UE drove pass the targeting eNB, the latency performance during the handover in practice will be much worse than the results presented in Figures 18 and 19. Also we consider the two eNB located outside of the city center in an open road. The handover process is expected to cause much higher latency increase in an urban environment. Even with these limitations, we can still observe that the average latency for both cloud and fog node increase around 40ms. According to our discussion in Section V, this means that most of the handover processes are successful in the first attempt.

D. Driving Speed

For a moving vehicle, it is expected that the fast driving speed will increase the chances of the Doppler effect resulting in much higher chances of packet drop or connection failure. To investigate the impact of the driving speed on the service latency in practical system, we analyze the latency traces at different driving speeds in Figure 24. We first present the mean and standard deviation of the latency traces in all the dataset collect from our driving measurement campaign. Surprisingly, we did not observe a significant increase of the RTT when the driving speed increases. For example, the average fog latency remains almost the same even when the driving speed approach 20m/s. The cloud latency increases for around 20ms at 20m/s of speed. This is because the driving speed can only become large when the vehicle is drove outside of the city center. The increased in-synchronization and disconnection probability will be compensated by the decrease of the reflection and blockage experienced inside the city. We compare the latency traces collected in an urban area with high eNB deployment density. We again observe a slight increase of the average cloud and fog latency, i.e., around 10-20ms of

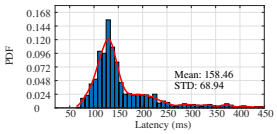

Fig. 14. Empirical PDF of cloud latency (MNO 2) in a multistory parking lot.

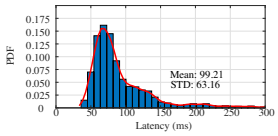

Fig. 15. Empirical PDF of fog latency (MNO 2) in a multistory parking lot.

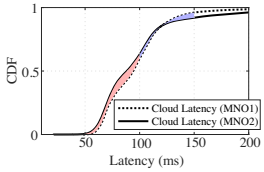

Fig. 16. CDF of cloud latency.

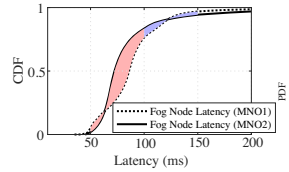

Fig. 17. CDF of fog latency.

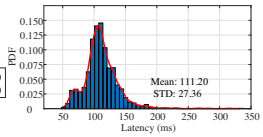

Fig. 18. Empirical PDF of cloud latency (MNO1) when driving between two eNBs.

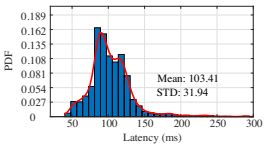

Fig. 19. Empirical PDF of fog latency(MNO1) when driving between eNBs.

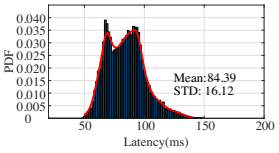

Fig. 20. Empirical PDF of cloud latency with AdaptiveFog.

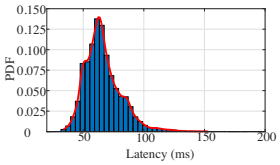

Fig. 21. Empirical PDF of fog latency with AdaptiveFog.

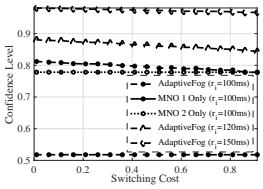

Fig. 22. Confidence level of cloud latency under different switching costs.

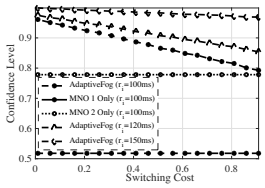

Fig. 23. Confidence level of fog latency under different switching costs.

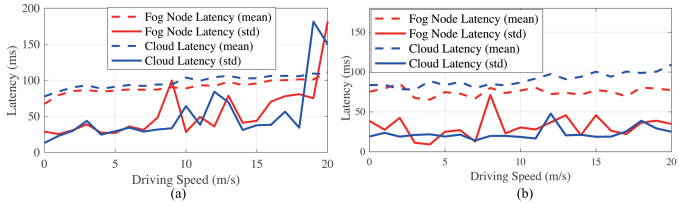

Fig. 24. Latency performance of (a) traces collected in an small urban region and (b) all the driving traces ranked by driving speeds.

increase for both latency. However, we observe a significant increase in the standard deviation of both latency when the speed become large (e.g., over 10m/s).

VII. Optimal Network/Server Selection and Adaptation

In this section, we derive the optimal policy for the UE to dynamically switch between two LTE networks. As mentioned earlier, in addition to the physical setup of the network, the latency of the UE is also closely related to the driving routine of the vehicle which can be affected by driver's intended destinations (e.g., home and office locations), driving habit, timing, and traffic conditions. This further increases the complexity of deriving the optimal policy. Fortunately, existing works as well as our measurement confirmed that the vehicle's future location and speed mainly depend on its current location and speed. In the rest of this section, we formulate the network adaptation and fog/cloud server selection as a Markov decision process (MDP). A simple threshold policy can then be derived for the UE to make an autonomous decision about whether or not to switch to another LTE network.

We formulate the network adaptation and fog/cloud server selection as a MDP with infinite horizon consisting of the following elements:

State: The state of the UE includes its driving speed v, location x, and the connected LTE network l. Typically, driving speed and location are continuous variables. However, in real world measurement study, the empirical PDF of the driving speed and location are generated from the histogram with only finite numbers of states. We can therefore define the state space $\mathcal{S}$ as a finite set of possible intervals of speed, location regions, and LTE network choices of the UE. We write each instance of state as $s = \langle v, x, l \rangle$ for $s \in \mathcal{S}$.

Action: The UE can decide whether or not to switch to another LTE network labeled as k for $k \neq l$ or stay with the current choice $k = l$. It can also choose to offload its workload to fog or cloud server. We assume each service can only be submitted to one LTE network at a time. We define the action space $\mathcal{A}$ of the UE as all the possible choices of LTE networks and fog and cloud servers. We also write the action of the UE as $a = \langle k, j \rangle$ for $a \in \mathcal{A}$.

State Transition Function: The probability of transitioning from one possible location and driving speed to another location and speed can be calculated from our driving measurement dataset. We observe that the driving speed as well as its probability of transitioning to another possible speed is closely related to the driving time. In particular, the driving speed and the probability of transitioning from one speed to another during rush hours is generally different from those during the non-peak hours. We therefore generate a set of different state transition probabilities at different time slots throughout a day. To simplify our description, we assume the state transition probability can be considered as fixed during the time slot of consideration and write the probability of state transferred from state s to s' when taking action a as $\Gamma(s', s, a) = \Pr(s'|s, a)$.

Utility Function: The main objective is to maximize the confidence level for the UE to have all the services being successfully served within the required latency. We assume the UE can receive requests from a set of services defined as $\mathcal{M}$ each has a fixed probability of arrival in each time slot denoted as p_i for service type i, for $i \in \mathcal{M}$. Let r_i be the maximum tolerable delay for type i service. To avoid the UE to switch back-and-forth between different MNOs, we assume a fixed cost for switching from one LTE network to another. We consider the instantaneous utility function in (3).

To maximize the long-term utility, the best action for the UE to take is to maximize both its utility in the current time slot as well as the expected utility in future time slots. This problem can be solved by employing the standard dynamic programming approach. We omit the details due to the limit

of space.

VIII. Numerical Results

In this section, we evaluate the performance of AdaptiveFog using our driving dataset. In Figures 20 and 21, we present the empirical PDFs of both fog node and cloud latency when the UE can use AdaptiveFog to dynamically switch between MNOs. We can observe that AdaptiveFog provides significant benefit to the fog latency with almost 15ms and 9ms reduction on the average latency compared to the case that the UE can only access a single MNO's LTE network. More importantly, AdaptiveFog reduces the standard deviation of the latency by almost a half compared to the scenario that the UE is stuck with a single MNO. For the cloud latency, the improvement on the average latency is relatively limited. However, we can again observe a significant reduction on the standard deviation of the cloud latency especially compared to the single MNO case.

It is obvious that the performance of AdaptiveFog is closely related to the cost for the UE to switch between MNO's networks. In Figures 22 and 23, we present the confidence level under different switching cost for both fog node and cloud latency with and without using AdaptiveFog. We compare confidence level of three latency thresholds, 100 ms, 120 ms, and 150 ms, corresponding to vehicular applications with different levels of stringent latency requirement. Note that confidence level of the single MNO will not change with MNO switching cost. We observe that, when the switching cost is low, AdaptiveFog achieves almost 30% improvement in confidence level of cloud latency, compared to the single-operator case. For the fog latency, AdaptiveFog achieves almost 50% improvement in the confidence level for supporting the active road safety applications. Note that these results are simulated by applying all of our driving data set to evaluate the performance improvement of AdaptiveFog. In some specific local area such as the one MNO has much higher eNB deployment density than the other, the performance improvement achieved by AdaptiveFog should be even higher.

IX. Conclusion

This paper has reported a city-wide measurement of the wireless access latency between a moving vehicle and a fog computing system connected through a multi-operator LTE network. A novel networking and server adaptation framework, called AdaptiveFog, has been proposed for vehicles to autonomously and dynamically connect with different LTE networks and fog or cloud servers. We have developed a smart phone app running on a moving vehicle to periodically measure the RTT of the UE when connecting with fog/cloud servers though different LTE networks. An empirical spatial statistic model is established to characterize the spatial variation of latency performance across various locations of the city. We introduce the weighted K-R distance to quantify the performance difference between different LTE networks. An optimal policy has been derived for a moving vehicle to sequentially switch to the optimal LTE networks. Extensive simulations have been performed. Our results show that AdaptiveFog achieves around 30% and 50% improvement in the confidence level for fog node and cloud latency, respectively.

Acknowledgment

The work of Y. Xiao is supported in part by the Fundamental Research Funds for the Central Universities under Grant No. 2019KFYXJJS180. The work of M. Krunz was supported in part by NSF (Grants # IIP-1822071, CNS-1409172, CNS-1563655, and CNS-1731164) and by the Broadband Wireless Access & Applications Center (BWAC).

References

[1] 5GAA, "Toward fully connected vehicles: Edge computing for advanced automotive communications," White Paper, Dec. 2017.
[2] NGMN Alliance, "V2X white paper," Jun. 2018.
[3] M. Chiang, B. Balasubramanian, and F. Bonomi, *Fog for 5G and IoT*. John Wiley & Sons, Apr. 2017.
[4] G. Premsankar, B. Ghaddar, M. D. Francesco, and R. Verago, "Efficient placement of edge computing devices for vehicular applications in smart cities," in *Proc. of IEEE/IFIP Network Operat. and Manage. Symp.*, Taipei, Taiwan, Apr. 2018.
[5] A. Asadi, S. Mullery, G. H. Sim, A. Kleiny, and M. Hollick, "FML: Fast machine learning for 5g mmwave vehicular communications," in *Proc of IEEE INFOCOM*, Honolulu, HI, Apr. 2018.
[6] 3GPP, "LTE; requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN)," 3GPP TR 25.913, Feb. 2010.
[7] I. Hadžić, Y. Abe, and H. C. Woithe, "Edge computing in the ePC: A reality check," in *Proc of ACM/IEEE Symposium on Edge Computing*, San Jose, CA, Oct. 2017.
[8] Z. Mir Hameed and F. Filali, "LTE and IEEE 802.11 p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 89, 2014.
[9] Z. Xu, X. Li, X. Zhao, M. H. Zhang, and Z. Wang, "DSRC versus 4G-LTE for connected vehicle applications: a study on field experiments of vehicular communication performance," *Journal of Advanced Transportation*, vol. 2017, 2017.
[10] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc of IEEE INFOCOM*, San Francisco, CA, Apr. 2016, pp. 1–9.
[11] R. Yu, G. Xue, and X. Zhang, "Application provisioning in fog computing-enabled internet-of-things: A network perspective," in *Proc of IEEE INFOCOM*, Honolulu, HI, Apr. 2018.
[12] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis, "FluidRAN: Optimized vRAN/MEC orchestration," in *Proc of IEEE INFOCOM*, Honolulu, HI, Apr. 2018.
[13] H. Inaltekin, M. Gorlatova, and M. Chiang, "Virtualized control over fog: Interplay between reliability and latency," *arXiv:1712.00100v2*, Feb. 2018. [Online]. Available: http://arxiv.org/abs/1712.00100
[14] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.
[15] H. Lee, J. Flinn, and B. Tonshal, "RAVEN: Improving interactive latency for the connected car," in *Proc. of ACM Mobicom*, New Delhi, India, Nov. 2018, pp. 557–572.
[16] L. Bedogni, M. Fiore, and C. Glacet, "Temporal reachability in vehicular networks," in *Proc of IEEE INFOCOM*, Honolulu, HI, Apr. 2018.
[17] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proc of Intern. Conf. Very Large Data Bases*, Toronto, Canada, Aug. 2004, pp. 180–191.
[18] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proc of IEEE INFOCOM*, Atlanta, GA, May 2017.
[19] ——, "Distributed optimization for energy-efficient fog computing in the tactile internet," *IEEE J. Sel. Area Commun.*, vol. 36, no. 11, pp. 2390–2400, Nov. 2018.
[20] 3GPP, "3GPP radio resource control (RRC) (Release 10)," 3GPP PS 36.331, v10.14.0, Sep 2014.
[21] ——, "Handover procedures," 3GPP TS 23.009, Jan. 2015.
[22] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 2014.