Exploring Intelligent Service Migration in Vehicular Networks

Onyekachukwu A. Ezenwigbo, Vishnu Vardhan Paranthaman, Glenford Mapp, and Ramona Trestian

Middlesex University, Faculty of Science and Technology, London, NW4 4BT
A.Ezenwigbo, V.Paranthaman, G.Mapp, R.Trestian@mdx.ac.uk

Abstract. Mobile edge clouds have great potential to address the challenges in vehicular networks by transferring storage and computing functions to the cloud. This brings many advantages of the cloud closer to the mobile user, by installing small cloud infrastructures at the network edge. However, it is still a challenge to efficiently utilize heterogeneous communication and edge computing architectures. In this paper, we investigate the impact of live service migration within a Vehicular Ad-hoc Network environment by making use of the results collected from a real experimental test-bed. A new proactive service migration model which considers both the mobility of the user and the service migration time for different services is introduced. Results collected from a real experimental test-bed of connected vehicles show that there is a need to explore proactive service migration based on the mobility of users. This can result in better resource usage and better Quality of Service for the mobile user. Additionally, a study on the performance of the transport protocol and its impact in the context of live service migration for highly mobile environments is presented with results in terms of latency, bandwidth, and burst and their potential effect on the time it takes to migrate services.

Keywords: Edge Computing · Service Migration · Vehicular Ad-hoc Network.· Quality of Service.

1 Introduction

Over the years, the cloud-based mobile applications have seen a significant increase in popularity, making it apparent that the next move within the networking arena would be towards an intelligent edge environment. In this context, one of the primary issues is the provision of guaranteed Quality of Service (QoS) for a wide variety of services. The existing centralized structure of the cloud-based architecture has made a general large geographical separation between mobile users and the cloud infrastructure. In this scenario, end-to-end communication between the mobile user and the cloud infrastructure can involve a lot of network hops, thereby introducing high network latency. Additionally, the network bandwidth of the cloud may also depreciate because the cloud infrastructure is accessed on a many-to-one basis [12]. The new approach to resolve the above problems is to install computing infrastructures at the edge of the network.

Technological advances in personal computers, tablets, and smartphones have increased the demand and fabrication of applications and services to support these developments. This has necessitated a rising need for QoS and Quality of Experience(QoE). The processing, memory and storage capacity of these new mobile, portable devices are becoming more industrious but not in terms of supporting high processing application demand within an expected time. These high demanding applications also result in a high level of battery consumption reducing the duration of user device usage. It is on this premise that the Mobile Cloud Computing (MCC) concept was adopted to provide the Mobile Node (MN) with the opportunity of cloud computing [3]. The MCC utilises strong reserved centralized clouds through a core network of a mobile operator and the internet to provide computing and storage facilities to MNs. The MCC is highly beneficial [1] as among other advantages, it helps to offload computing, storage and memory of applications to the cloud which prolongs battery life; allowing the mobile user to access highly developed and demanding applications; as well as making increased storage facilities available to users. However, the traditional cloud for mobile users results in high latency since data is sent to a central cloud sever that is remotely located from its users in terms of network topology.

Furthermore, Edge Computing (EC) is an existing and capable approach to access large data locally and evade extensive latency [19, 24], especially in vehicular networks as the MN moves at a high speed and hence, requires low latency. Vehicular users requesting services through a core network from edge networks far from the cloud may cause extensive latency. The EC is established to overcome these disadvantages of traditional cloud computing [23, 17]. A lot of research have motivated Vehicular Edge Computing (VEC). Most of the recent research focused on the VEC architecture design still fail to look into mobility and how it can affect service migration in a highly mobile environment.

Moreover, the migration of the cloud services close to MNs helps to address the problem of high latency by moving services close to the MNs, i.e., to the edge of mobile network as considered in newly emerging EC paradigm as part of MCC. However, in the conventional MCC, the cloud services are called up via the Internet connection [16] whereas in the case of the EC, the computing resources are located in proximity of the MNs. Therefore, the EC can offer significantly lower latencies and jitter when compared to the traditional MCC. On the other hand, the EC provides only limited computational and storage resources with respect to the centralised cloud computing.

The Vehicular Ad-Hoc Networks (VANETs) have emerged as a promising field of development and research, VANETs will enable the communication between vehicles and infrastructures on the road to provide services such safety, entertainment and infotainment. In VANET, vehicles and RSUs (Road-Side Units), i.e. network nodes, will be equipped with on-board computation and communication modules to make sure better communication is possible between them. It supports 802.11p which is required to support Intelligent Transportation Systems (ITS) applications because of available high bandwidth [4, 15]. Existing research on VANET focuses on communication between end nodes and the infrastructure and hence has failed to look into end to end communication with required QoS and security. Service migration is a technique which can helps in gaining high QoS and QoE by migrating the services closer to the user. Due to the high rate of vehicular mobility in a VANET environment, the highly dynamic topologies are frequently prone to network disconnection and fragmentation. Consequently, these inherent characteristics are bound to degrade the QoS provided by the VANET infrastructures. Therefore, the establishment of robust VANETs that could effectively support applications and services on a large geographical scale remains an open challenge.

There are lots of research efforts that look into migrating the services from the core cloud to the edge of the network but the focus of this paper is to explore service migration between nodes at the edge of the network in order to provide better QoS in vehicular networks. In this context, this paper investigates the benefits of integrating Mobile Edge Computing (MEC) within the Vehicular Ad-Hoc Network (VANET) scenario. A real experimental vehicular network test-bed is introduced. Results collected from this test-bed are used to better understand the impact of live service migration within a VANET environment.

The rest of the paper is structured as follows. Section 2 introduces the literature review while Section 3 looks at service delivery for mobile clients. Section 4 analyses the wireless coverage parameters for mobile networks and Section 5 shows edge to edge service migration. Section 6 looks at the experimental test-bed and results in detail. Section 7 concludes the paper.

2 Literature review

In order to analyse the effects of EC on reducing web response time authors in [6] derived a formula that reduces the response time of web pages by delivering objects from edge nodes. They investigated the effect of edge computing in different web categories such as sports and news. They were able to achieve this with their numerical evaluations using the data obtained by browsing about 1,000 web pages from 12 locations in the world.

Furthermore, authors of [5] proposed a model for system latency of two distributed processing scenarios by analysing the system latency of EC for multimedia data processing in the pipeline and parallel processing scenarios. They highlighted that both models can follow the actual characteristics of system latency. With regard to delay constrained offloading for MEC in cloud-enabled vehicular networks, the authors in [26] proposed a vehicular offloading framework in a cloud-based MEC environment. They were able to investigate the computation offloading mechanism. The latency and the resource limitations of MEC servers were taken into consideration which enabled the proposal of a computation, resource allocation and a contract-based offloading scheme. The scheme intends to exploit the utility of the MEC service provider to satisfy the offloading requirements of the task.

Given the significance of increased research in combining networking with MEC to support the development of 5G, the authors in [20] investigated the conceivable outcomes of engaging coordinated fiber-wireless (Fi-Wi) to get networks to offer MEC abilities. More predominantly, planned deployments of MEC over Fi-Wi networks for typical Radio Access Network(RAN) advancements were explored, representing both network architecture and enhanced resource management. Moreover, authors of [25] showed the architectural description of a MEC platform along with the key functionalities. They agreed that the RAN is enhanced by the computation and storage capacity provided using MEC. The primary benefit of MEC is to allow significant latency reduction to applications and services as well as reduced bandwidth consumption. The enhancement of RAN with the MECs capability can rely on its edge server cloud resources to provide the context-aware services to nearby mobile users in addition to conducting the packet forwarding.

For performance evaluation of edge cloud computing systems for big data applications, acceptable performance was revealed in [2] using Hadoop to build a visualisation machine for small clouds. In [22], [10] and [21], the intended functioning of the projected system has been presented in an attempt to determine if the migration of a service is required. The proposed model allows services to migrate from one cloud to another. Kikuchi et al. [7] proposed a MEC-based VM migration scheme whereby a VM migration is conducted to reduce congestion at the edge of the network. They addressed two QoS problems which were the congestion in a wireless access network and congestion in computing resources on an edge with the use of TCP throughput.

Kim et al. [8] did a study on service instance allocation algorithms to maintain the QoS for mobile user with the help of a service migration tool. They were able to show simulations and compared their result to a heuristic algorithm and reference algorithms. This led to the understanding that their proposed algorithm performed better in a larger user population imbalance.

Recently, the ability of using low cost devices that have virtualization services was regarded as a better alternative to support computational requirements at the edge of a network. Lertsinsrubtavee et al. [9] introduced PiCasso, which is a lightweight service orchestration at the edge of the network. They further analysed and discussed their benchmarking results which enabled them to identify important parameters PiCasso would need in order to be taken into considerations for use in future network architecture.

A recent survey on architecture and computation offloading in MEC [11], explained that current research carried out regarding the MEC is basically how to guarantee service continuity in highly dynamic scenarios. They clearly state that this part is lacking in terms of research and is one of the stopping point to the use of the MEC concept in real world scenarios. Furthermore, they argued that recent validated research will not be acceptable due to their simplistic scenarios, simulations or analytical evaluations. Instead real tests and trials are further required in realistic scenarios.

With all the works mentioned above, the authors did not take into account the communication dynamics in highly mobile environments such as vehicular network i.e., handover in wireless network which is essential in order to provide a fully supported edge cloud computing environment and this is the focus of this paper.

3 Service Delivery for Mobile Clients

Current networks consist of three parts i.e., the core network, core end-point, and edge. The core network provides services to the users and has high computation, memory, and storage capacity. The core end-points are the end devices such as routers which enables communication and connectivity to other clouds and the edge network. Edge network consist of edge servers, access point, base station, etc., and has relatively low computation, memory, and storage capacity compared to the core network.

Providing services to the highly mobile system such as a vehicular network with high QoS is a key challenge in the future networks as the MN will be switching to different networks due to mobility. In order to address this challenge several works have proposed service migration as a solution. Service migration involves continuous migration of services closer to users as they move around resulting in reduced latency and better QoE [25]. For example, let us consider a scenario as shown in Fig. 1, where the cloud server is in the core network, routers are Core Endpoints, wireless access points are at the edge network and the MN as the client. There are four possible cases as described below:

- **Case 1:** A mobile user is travelling in a car and the user experiences a service offered by the core network i.e. the cloud that offers the service and the client receives it.
- **Case 2:** The service is running from the Core Endpoint, therefore, the delay between the Core Endpoint and the client should be lower than the case 1.
- **Case 3:** Services are running at the edge i.e., Edge server, Access point, etc. Therefore, the delay should be lower than the previous cases.
- **Case 4:** Here, the service are running within the client. This will be useful when the users have enough computing, memory and storage resources.

In this paper, we study the impact of service migration between the edge access points in a vehicular network which is called the Road-side Unit (RSU) in order to support better QoS for highly mobile nodes. There are several factors or functional requirements that have to be considered in migration of a service as explained below [21]:

- **Requirements 1:** A service needs to be recognisable by a unique ID and guaranteed to a set of parameters that can interoperate with the platform providers. The minimum required parameters must include CPU time, memory and storage, security protocols, network bandwidth and latency, and dependencies on other services.

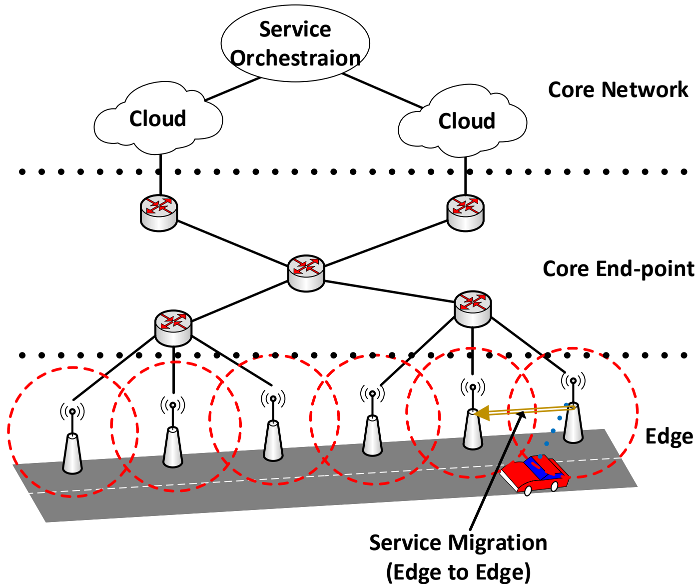

Fig. 1. Service Migration Scenario

- **Requirements 2:** The service should allow personalisation in terms of performance by the user, this would enable a user to make upgrades in terms of extra features or services. These parameters may include: maximum latency, a fixed allocated bandwidth to the user, security level, amount of storage, as well as taking into account the mobility of the user.
- **Requirements 3:** Platform providers (Clouds) can accept or reject services based on their set of requirements. They are required to bill service providers for processing, network and storage usage for any services and additional components running on their Cloud infrastructure. Furthermore, Cloud provider have the ability to decide which technology should be used for service migrations as long as it meets the service's minimum requirements.
- **Requirements 4:** To provide maximum benefits to their users, services have to be aware of their QoS level on a per-client basis. A server should stay alert in order to be aware of the client's network provider and current location. Such data can be gathered directly by the service and its processes, the client's device or a transport protocol that can report such information. This information will help to determine when and where to migrate.
- **Requirements 5:** If a service is requesting migration, it must pass information about the client's network provider to the platform provider to allow the finding of the best alternative Cloud to host the service. If possible a Cloud that is directly peering or local to the client's network.
- **Requirements 6:** Any Cloud offering resources for incoming services should report nominal values of network latency and bandwidth to the user's network. This helps to ensure that an incoming service will not only have suf-

ficient Cloud resources to run but also satisfactory network performance to deliver its content at the QoS demanded by the client.

- **Requirements 7:** The Serviced clients should have the ability to select the best possible network for handover via a querying mechanism which will confirm that the desired QoS level is deliverable through the new network. Therefore, clients will not rely on reported nominal values to determine the best network for service. If a handover to a network with less QoS is imminent, the service should migrate to an appropriate location (if it exists) to better the QoS.

Based on several requirements mentioned above the service migration time (ST) i.e, the time taken to migrate a service will be developed. In a highly mobile environment, ST is not the only parameter that can be used to decide whether a service can be migrated or not. In addition to the ST, the mobility aspects of the user has to be considered i.e, the time MN is expected to spend in the network coverage. If ST is greater than communication time (within the coverage region) then, by the time the service is migrated, the MN will be out of the coverage region, therefore, the service will not be successful. For example, Let us assume it takes 20s seconds to migrate a service from one RSU to another, when a MN is moving to the next coverage region and spends more than 20s then service can be migrated and the service can be received from the RSU. But if the MN is spending less than 20s in the new network then communication would be void because MN will be out of the coverage region. The following section will describe in detail the communication and mobility aspects in a highly mobile environments.

4 Wireless Coverage Parameters for Mobile Networks

In this section, we introduce a set of network coverage parameters that will be used in the following sections to demonstrate the service migration in highly mobile environment. The network coverage area is a region with an irregular shape where signals from a given Point of Attachment (PoA) i.e., Access Point or Base Station can be detected by a MN. The signals from the PoA are unreliable at the boundary and beyond the coverage area as the signals from the PoA cannot be detected. For seamless communication, handover should be finished before the coverage boundary is reached.

Therefore, a circle known as the handover radius (R_H) and exit radius (R_E) was defined in [13] to ensure smooth handover. The work states that the handover must begin at the exit radius and should be completed before reaching the handover radius boundary as shown in Fig. 2.

The exit radius will therefore be dependent on the velocity, ν, of the MN. If we represent the time taken to execute a handover by T_{EH}, then:

$$T_{EH} \leq \frac{(R_H - R_E)}{\nu} \tag{1}$$

Hence, exit radius can be given as shown in Equation (2)

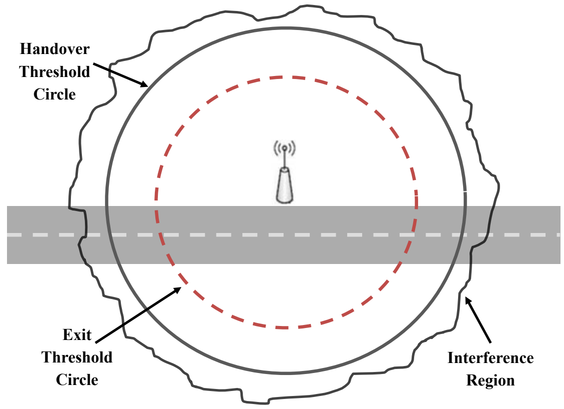

Fig. 2. Network Coverage

$$R_E \leq R_H - (\nu * T_{EH}) \tag{2}$$

So, the faster a MN moves, the smaller the R_E at which handover must begin. Given that we know the time taken to execute a handover, the velocity of the MN and handover radius, then we can calculate the exit radius which is dependent on the handover radius. A good estimation of the handover radius is required for the proposed approach which is dependent on the propagation models being used. The time taken to effect a handover was shown to be dependent on various factors such as Detection Time (t_{det}), Configuration Time (t_{con}), Registration Time (t_{reg}) and Adaptation Time (t_{adp}) as discussed in [13].

Our previous work on proactive handover in [14] showed that the above-mentioned coverage parameters can be segmented into communication ranges and presented an in-depth analysis of such segmentation and their importance in-order to achieve a seamless handover as shown in Fig. 3. This segmentation can be put into effective use for achieving proactive handover, resource allocation, and service migration for a highly mobile environment.

Time before handover (Υ) is the time after which the handover process should start and Time to handover ($\hbar$) is the time before which the handover to next coverage range has to be completed. Network Dwell Time ($\aleph$) is the time MN will spend in the coverage i.e., the Network Dwell Distance (NDD) of new network. Resource Hold Time ($\mathbb{N}$) is the resource usage time or when actual exchange of data is taking place. $\hbar$ and $\mathbb{N}$ are the two key parameters that has to be considered for service migration in highly mobile networks.

5 Edge to Edge Service Migration

Let's suppose the MN is travelling at a velocity, ν from one RSU's coverage region to the next one: with an estimate of the Υ, $\hbar$ and $\mathbb{N}$, it is possible to

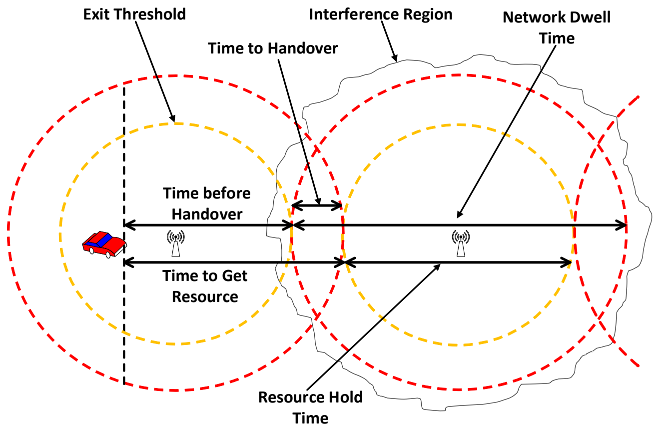

Fig. 3. Communication Range Segmentation

decide whether a service should be migrated with the knowledge of ST. Hence, ST should be less than the sum of $\hbar$ and $\mathbb{N}$ in order to have a seamless service. If the ST is greater than the sum of $\hbar$ and $\mathbb{N}$ then the MN will be out of coverage of the next RSU due to mobility by the time the service is migrated. Therefore, in order to get effective service

$$(\hbar + \mathbb{N}) > ST \tag{3}$$

Hence, for a seamless service to the MN the ratio of the communication times due to mobility and the service migration time has to be always greater than 1 as shown in the equation below. Here, $\hbar$ is usually very small as it is the time taken to handover.

$$\frac{(\hbar + \mathbb{N})}{ST} > 1 \tag{4}$$

The above equation denotes a reactive approach i.e., the service migration will only start after the MN reaches the next RSU's coverage. Therefore, when the MN reaches the next coverage range where $\Upsilon = 0$, i.e, during handover, the serivce will be migrated to the next RSU. In summary, communication and service handover takes place at the same time, this is called a reactive communication and service handover. The reactive migration approach might disrupt the service due to mobility for services with higher migration time. Hence, we need a proactive service migration to be adopted for better QoS and QoE.

In the proposed proactive approach the service migration will begin before the Υ i.e., before the communication handover as shown in Fig. 4. The point where the service is starting to migrate is called as proactive service migration time (X). When the service begins to migrate at point X before the communication handover, so the amount of time left is ST-X. This means that, X should be less than or equal to ST, which will ensure that the service is not migrated far ahead before the MN reaches the coverage. Therefore,

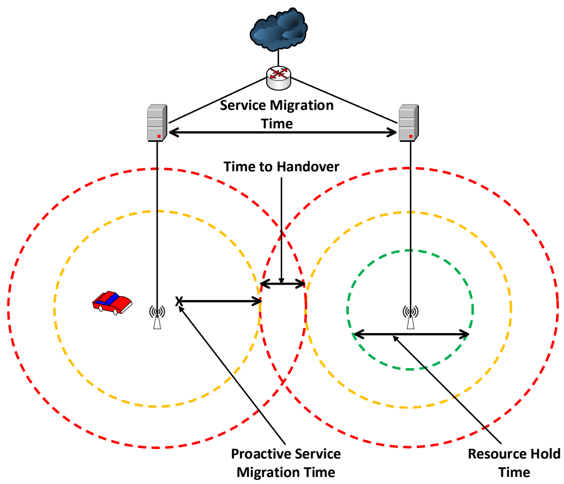

Fig. 4. Service Migration Segmentation

$$(\hbar + \mathbb{N}) > (ST - X) \tag{5}$$

As explained earlier for a seamless service to the MN the ratio of the communication times due to mobility and the service migration time has to always be greater than 1 as shown below:

$$\frac{(\hbar + \mathbb{N})}{(ST - X)} > 1 \tag{6}$$

6 Experimental Framework

In order to explore Equation 6, the values of the parameters $\hbar$ and $\mathbb{N}$ were obtained using a VANET test-bed. In addition, the Service Time (ST) was obtained for different applications and services which include No application, Game server, RAM Simulation, Video Streaming, and Face Detection. These applications were fully explored in a previous paper and accurate values of ST were obtained and used later in the paper to calculate service migration ratio as shown in Equation 6.

6.1 Experimental Test-Bed and Results

This section provides the details of the real experimental VANET test-bed. The Connected Vehicle Test-bed, was built by Middlesex University and the Department of Transport (DfT) using ETSI Intelligent Transport System (ITS) G5 (VANET) technology [15]. The test-beds were built on the Hendon Campus in London and alongside the surrounding roads. The test-bed uses seven RSUs and also extends to the A41 (Watford Way) behind the campus. Four RSUs, which were deployed on the MDX buildings, were backhauled directly to the university's gigabit ethernet network and the three RSUs deployed along the A41 were backhauled using LTE with a secure VPN tunnel service provided by Mobius Network as shown in Fig.5. They are now fully operational and trials have been held to fully understand the technology and concerns around its wide-scale deployment as well as communication dynamics needed to attain seamless communication for this environment.

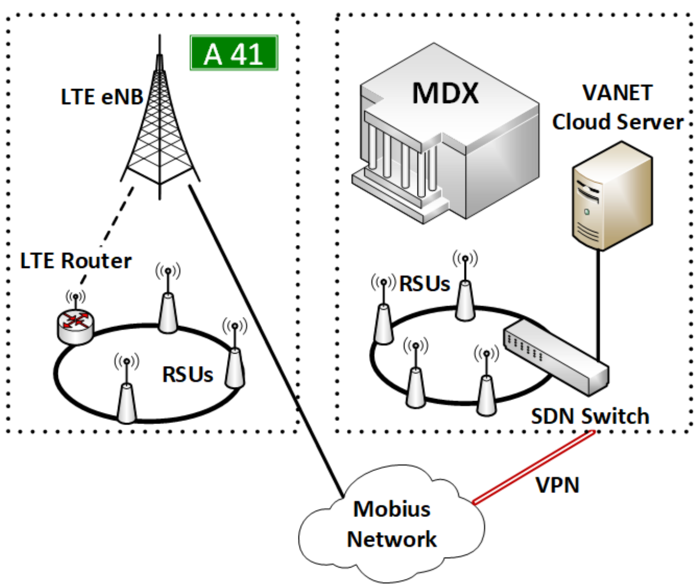

Fig. 5. MDX VANET Testbed Network Diagram

The coverage map of this testbed is as shown in Fig.6. The NDD for each RSU was measured from the coverage map and with the NDD, the $\aleph$ can be calculated if the velocity of the vehicle is known. Table 1 shows the $\aleph$ and $\mathbb{N}$ for two different velocities i.e., 30 Mph and 50 Mph for all the RSUs. We know from [13] that the handover execution time i.e., $\hbar$ is 4s and therefore, the $\mathbb{N}$ is:

$$\mathbb{N} = \aleph - \hbar \tag{7}$$

With the knowledge of the mobility, $\mathbb{N}$, $\hbar$, and in addition, if the ST can be estimated, then we can efficiently decide when and where to migrate a service. The following subsection will present the result of our approach.

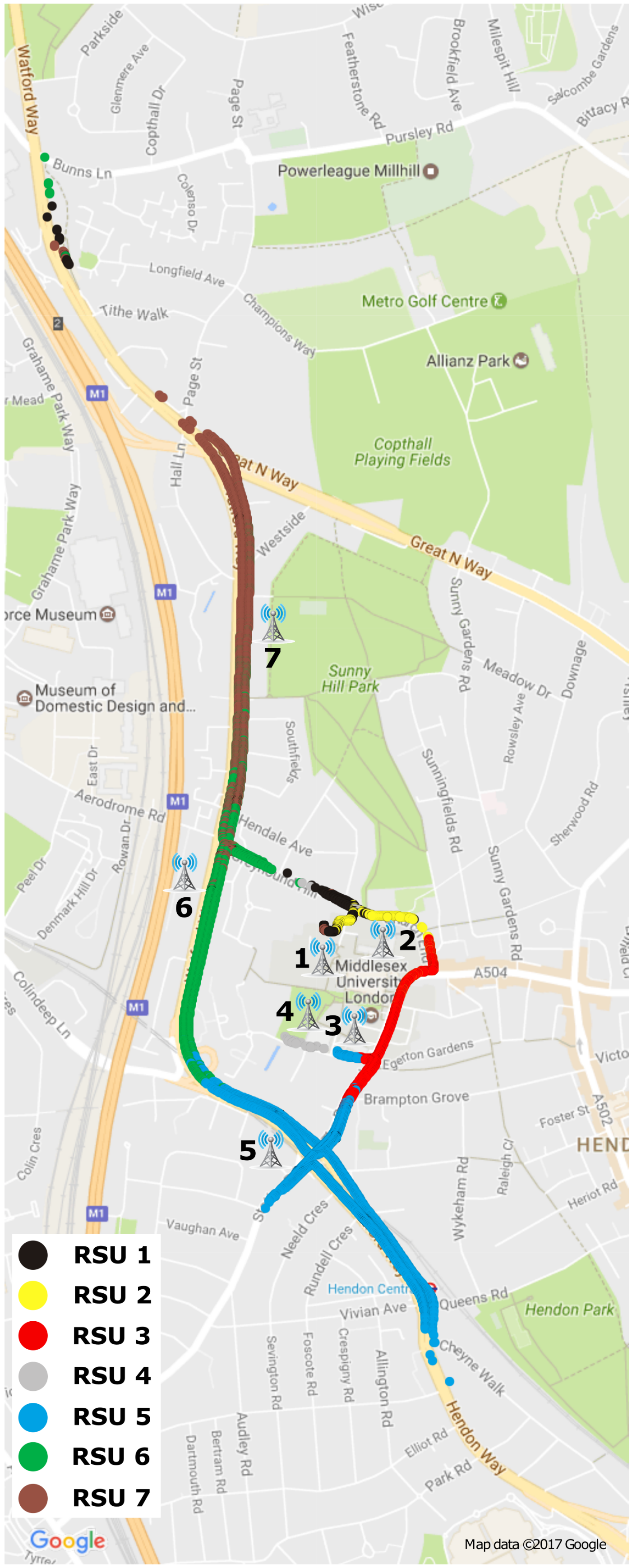

Fig. 6. Full Coverage and Overlapping Map for A41, Watford Way, Hendon, London

Table 1. Communication Coverage Segmentation Distance and Time ($\hbar = 4s$).

RSU No.	NDD	30 Mph		50 Mph	
		$\aleph$	$\mathbb{N}$	$\aleph$	$\mathbb{N}$
RSU 1	300 m	22.37 s	18.37	13.42 s	9.42 s
RSU 2	456 m	34.00 s	30.00 s	20.40 s	16.40 s
RSU 3	517 m	38.55 s	34.55 s	23.13 s	19.13 s
RSU 4	248 m	18.49 s	14.49 s	11.09 s	7.09 s
RSU 5	974 m	72.63 s	68.63 s	43.57 s	39.57 s
RSU 6	1390 m	103.64 s	99.64 s	62.19 s	58.19 s
RSU 7	1140 m	85.00 s	81.00 s	51.00 s	47.00 s

6.2 Live Service Migration Use Case Scenario Results

Moving a service across the edge nodes is an essential factor with regards to ensuring that users have good performance with proximity especially in a highly mobile network like vehicular networks. This section investigates the impact of live service migration within a VANET environment by making use of the results collected from the real experimental VANET test-bed as well as the results for ST presented in [12] for different services. We have considered two RSUs; RSU 2 located at Williams building and RSU 3 located at Hatchcroft building. The scenario considered was a MN travelling at two different velocities i.e., 30Mph and 50Mph which is being handing over from RSU 3 to RSU 2. The NDD of these RSUs from the MDX VANET test-bed has been used in this paper to demonstrate the effectiveness of our service migration model.

The authors in [12] detail a layered framework for migrating active service applications which are condensed in virtual machines (VMs) and containers. Containers are a developing technology and they consume less storage space compared to VMs, therefore, will be appropriate for service migration. Under the given framework, the migration performance in terms of ST for both VM and containers were examined in a controlled test-bed environment. They have considered five different applications for migration; No Application, Game Server, RAM Simulation, Video Streaming and Face Detection. The results on the performance of two i.e., 2 layer and 3 layer approaches for different applications with container technology as shown in Table 2 have been used in this work for the evaluation of our model.

Given the measurement results for service migration as listed in Table 2, we consider two use-cases for two different velocities i.e., 30Mph and 50Mph:

- Reactive Approach where the service migration starts once the MN reaches the next RSU's coverage region.
- Proactive Approach where the service migration starts before the MN reaches the next RSU's coverage region at point X called the proactive service migration time.

The graphs presented in Fig. 7 and 8 show the ratio for reactive and proactive service migration derived from Equation 4 and 6 respectively for different appli-

Table 2. Migration results for two-layer and three layer configurations [12]

Services	ST / Data Transferred 2 Layer	3 Layer
No Application	6.5 s /1.4 MB	11.0 s/ 1.9 MB
Game Server	7.3 s /2.2 MB	10.9 s/ 2.7 MB
RAM Simulation	20.2 s /97.1 MB	27.2 s/ 97.6 MB
Video Streaming	27.5 s /180.2 MB	37.3 s/ 184.6 MB
Face Detection	52.0 s /363.1 MB	70.1 s/ 365.0 MB

cation services presented in Table 2. For a service to be successfully migrated the ratio has to be above 1, which is the threshold.

In case 1, we can observe that the service migration in a mobile environment is successful for both No application, and Game server with 2 layer and 3 layer approach for both 30Mph and 50Mph as shown in Fig. 7. This is due to the fact that the size of the service is small compared to others. RAM Simulation service cannot be successfully migrated for 3 layer approach at 50Mph, this is due to the high speed and rest of the cases for RAM Simulation service can be successfully migrated. The other two services i.e, Video streaming and Face detection cannot be migrated in a reactive service migration.

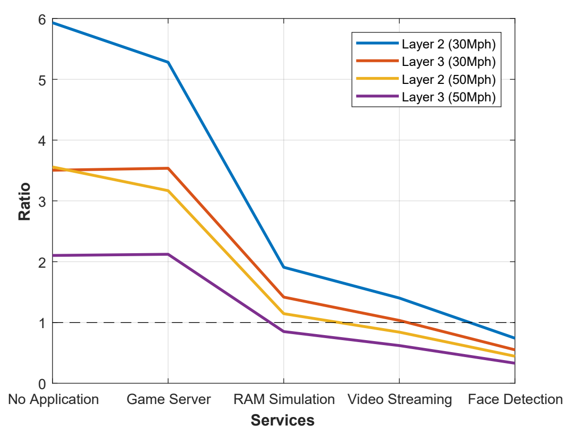

Fig. 7. Service Migration for Reactive Handover

In case 2, Fig. 8 shows the results of proactive approach i.e, service is migrated at the proactive service migration time, X. This is not a constant value and it changes according to the type of the service. We consider the values of X (in seconds) for No Application and Game server as 5s, RAM simulation and video streaming to be 15s and then finally face detection as 45s. The results show that the No Application, Game server, and RAM Simulation are successfully migrated for all cases. Video streaming, and Face detection are almost nearing the threshold 1 for 3 layer approach at 50Mph and all other cases can be successfully migrated. This shows the need for a proactive service migration approach in a highly mobile edge environment. In addition, proactive service migration time, X has to be explored in the future to develop efficient algorithms for such service migration. Here, the ST was measured in a controlled environment and therefore, estimating the ST in real-time is necessary.

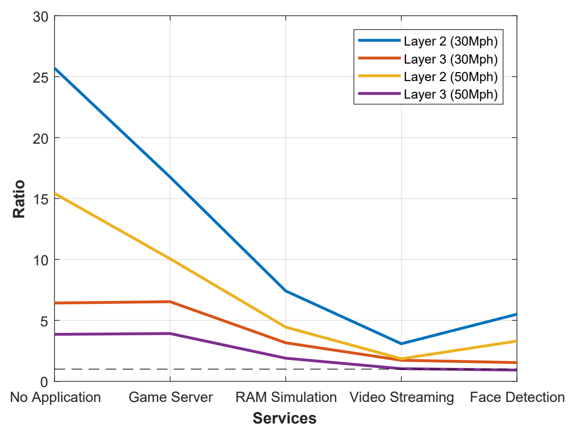

Fig. 8. Service Migration for Proactive Handover

6.3 Transport Protocol Performance Evaluation over VANET

To be able to estimate the ST in real time, the underlying transport protocol plays a significant role. In this context, this section compares the performance of two transport protocols: Simple Lightweight Transport Protocol (SLTP) and Transmission Control Protocol (TCP). The details on SLTP functionality and its internal structure can be found in [18].The aim is to demonstrate the effectiveness of SLTP as a low latency protocol which works by measuring the service time based on the available bandwidth. Two sets of experiments are conducted

- RSU to VANET server link performance
- RSU to RSU link performance .

Our first set of results will look at SLTP running at the VANET server and the RSU. Packets of different sizes were sent and the time taken to receive them back at the sender was measured. This gives us a direct measurement of protocol performance. We performed our benchmarks by using the following hardware specifications as shown in Table 3 for the VANET server and the RSU. It can be clearly seen that the resources on the RSU are quite limited compared with the VANET server or a modern PC.

Table 3. Hardware Specifications

Specifications	VANET Server	RSU
Processor	Intel (R) Xeon (R) CPU E5-2683 v4	MIPS 24Kc V7.4 (1 core)
RAM	32GB	64 MB SDRAM (512 Mbits)
Storage	500GB	16 MB Flash
Network	1 Gigabit Ethernet	1 Gigabit Ethernet
OS Type	Debian 3.16.43(64 Bit)	Debian 2.6.32.27 (32 Bit)

Since SLTP runs over User Datagram Protocol (UDP), the size of a single SLTP packet can be up to (64KBs - 8 bytes, the size of the UDP header). However, for testing we wanted to ensure that SLTP packets could fit into a whole number of Ethernet packets which can carry a payload of 1500 bytes. This means that each SLTP packet contained 1452 bytes of user data which is comparable to the packet utilization of standard TCP. Though SLTP supports a window size of 1 MB, it was decided to use a window size of 144 KBs to ensure that received buffers were not overrun. The result, shown in Fig. 9, indicates that in this arrangement, less CPU cycles as well as memory are available on the RSU and hence, the performance of SLTP is not much greater than the standard TCP.

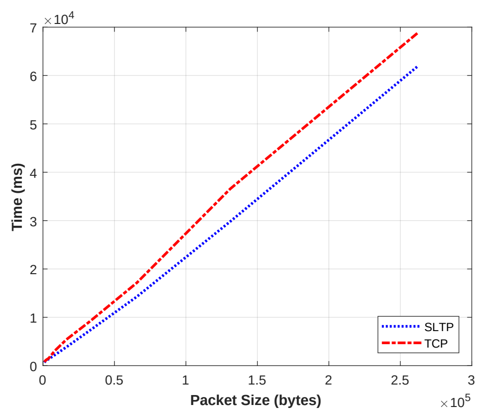

Fig. 9. RSU to VANET Server - Network Time: SLTP vs TCP

Finally, the results for RSU to RSU are given in Fig.10. These results show that with two RSUs there is very limited resources at user level and hence, TCP in the kernel will outperform SLTP. Overall, these graphs indicate that to get good performance in user space requires an abundance of CPU, memory and network resources which is an important factor for service migration at the edge.

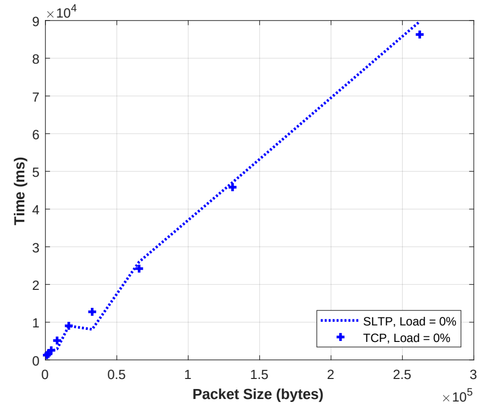

Fig. 10. RSU to RSU - Network Time on Load = 0%: SLTP vs TCP

6.4 RSU to RSU Link Performance under different load conditions

Since, SLTP runs in user space, it is important to understand how its performance is affected by different load characteristics of the system. In order to explore this, a flexible hog program was used to remove idle CPU cycles at the user level from the system. Hence, we were able to obtain readings with the system being under various loads, including 25%, 50%, 75% and 100%.

Fig.11 shows the time taken to transfer packets of different buffer sizes under different loads and it clearly shows that as the load increases SLTP underperforms TCP as less cycles are available in user space.

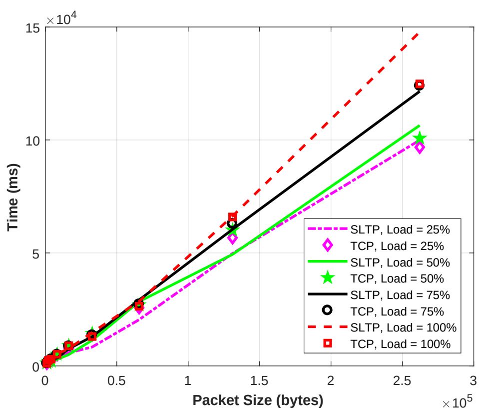

Fig. 11. RSU to RSU - Network Time on Load = 25% to 100%: SLTP vs TCP

The bandwidth results under different loads as shown in Fig.12 reveal that there are only significant differences for small packet sizes. However, after around 2KBs the bandwidth available falls to around 2.5MB/s. This is important for applications needing large packet transfers sizes such as multimedia applications.

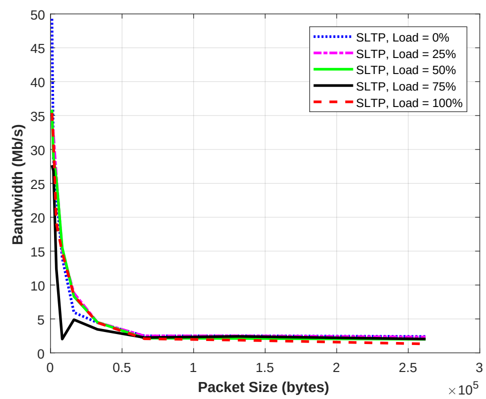

Fig. 12. RSU to RSU - Bandwidth on Load: SLTP

The latency results as measured by SLTP using different packet sizes under different loads are shown in Fig.13. It shows that the latency increases with increasing load especially after 50KBs.

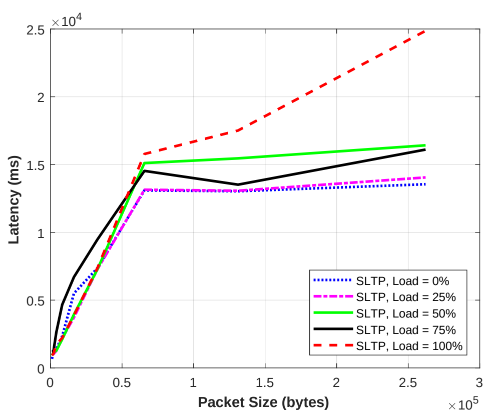

Fig. 13. RSU to RSU - Latency on Load: SLTP

Finally, the burst results as shown in Fig. 14 clearly show that the system is affected by high loads especially for small packets. After around 10KBs the burst size is severely reduced.

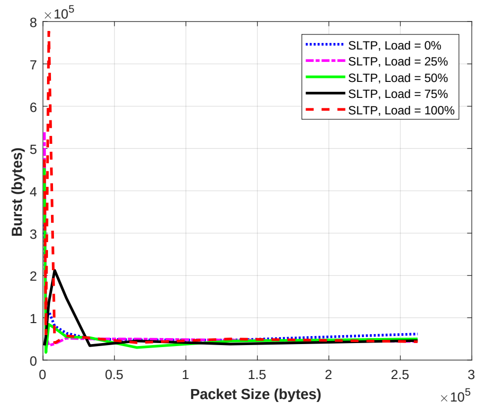

Fig. 14. RSU to RSU - Burst on Load: SLTP

7 Conclusion

The dynamics of the wireless environment makes the provisioning of guaranteed QoS a significant challenge especially within highly mobile environments like VANET. In this context, intelligent MEC is seen as part of the solution. This paper studies the impact of integrating MEC within VANET in terms of mobility, handover and service migration between edge access points for different applications. Additionally, a new service migration threshold model is proposed. The results collected from a real experimental vehicular network test-bed were used to better understand the impact of live service migration within a VANET environment in the context of reactive versus proactive service migration scenarios. The results show that a proactive service migration is more efficient within a MEC-VANET environment. Additionally, we investigated the performance of the underlying transport protocol and demonstrated its impact in the context of live service migration for highly mobile environments.

References

1. Barbarossa, S., Sardellitti, S., Lorenzo, P.D.: Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks. IEEE Signal Processing Magazine **31**(6), 45–55 (Nov 2014). https://doi.org/10.1109/MSP.2014.2334709

2. Femminella, M., Pergolesi, M., Reali, G.: Performance evaluation of edge cloud computing system for big data applications. In: 2016 5th IEEE International Conference on Cloud Networking (Cloudnet). pp. 170–175 (Oct 2016). https://doi.org/10.1109/CloudNet.2016.56
3. Hoang T. Dinh, Chonho Lee, D.N.P.W.: A survey of mobile cloud computing: architecture, applications, and approaches. Wireless comunication and mobile computing (2013)
4. Hussain, R., Son, J., Eun, H., Kim, S., Oh, H.: Rethinking vehicular communications: Merging vanet with cloud computing. In: 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings. pp. 606–609 (Dec 2012). https://doi.org/10.1109/CloudCom.2012.6427481
5. Imagane, K., Kanai, K., Katto, J., Tsuda, T.: Evaluation and analysis of system latency of edge computing for multimedia data processing. In: 2016 IEEE 5th Global Conference on Consumer Electronics. pp. 1–2 (Oct 2016). https://doi.org/10.1109/GCCE.2016.7800393
6. Kamiyama, N., Nakano, Y., Shiomoto, K., Hasegawa, G., Murata, M., Miyahara, H.: Analyzing effect of edge computing on reduction of web response time. In: 2016 IEEE Global Communications Conference (GLOBECOM). pp. 1–6 (Dec 2016). https://doi.org/10.1109/GLOCOM.2016.7841607
7. Kikuchi, J., Wu, C., Ji, Y., Murase, T.: Mobile edge computing based vm migration for qos improvement. In: 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). pp. 1–5 (Oct 2017). https://doi.org/10.1109/GCCE.2017.8229344
8. Kim, S.Y., de Foy, X., Reznik, A.: Practical service allocation in mobile edge computing systems. In: 2017 27th International Telecommunication Networks and Applications Conference (ITNAC). pp. 1–6 (Nov 2017). https://doi.org/10.1109/ATNAC.2017.8215372
9. Lertsinsrubtavee, A., Ali, A., Molina-Jimenez, C., Sathiaseelan, A., Crowcroft, J.: Picasso: A lightweight edge computing platform. In: 2017 IEEE 6th International Conference on Cloud Networking (CloudNet). pp. 1–7 (Sept 2017). https://doi.org/10.1109/CloudNet.2017.8071529
10. Li, H., Shou, G., Hu, Y., Guo, Z.: Mobile edge computing: Progress and challenges. In: 2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud). pp. 83–84 (March 2016). https://doi.org/10.1109/MobileCloud.2016.16
11. Mach, P., Becvar, Z.: Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys Tutorials **19**(3), 1628–1656 (thirdquarter 2017). https://doi.org/10.1109/COMST.2017.2682318
12. Machen, A., Wang, S., Leung, K.K., Ko, B.J., Salonidis, T.: Live service migration in mobile edge clouds. IEEE Wireless Communications **25**(1), 140–147 (February 2018). https://doi.org/10.1109/MWC.2017.1700011
13. Mapp, G., Katsriku, F., N., M.A., Chinnam, Lopes, R., Moreira, E., Vanni, R.M.P., Augusto, M.: Exploiting location and contextual information to develop a comprehensive framework for proactive handover in heterogeneous environments. Journal of Computer Networks and Communications **2012**(Article ID 748163), 1–17 (2012). https://doi.org/10.1155/2012/748163
14. Mapp, G., Gosh, A., Paranthaman, V.V., Iniovosa, V.O., Loo, J., Vinel, A.: Exploring seamless connectivity and proactive handover techniques in vanet systems. In: Intelligent Transportation Systems. pp. 195–220. Springer (2016)
15. Paranthaman, V.V., Ghosh, A., Mapp, G., Iniovosa, V., Shah, P., Nguyen, H.X., Gemikonakli, O., Rahman, S.: Building a prototype vanet testbed to explore com-

munication dynamics in highly mobile environments. In: International Conference on Testbeds and Research Infrastructures. pp. 81–90. Springer (2016)
16. u. R. Khan, A., Othman, M., Madani, S.A., Khan, S.U.: A survey of mobile cloud computing application models. IEEE Communications Surveys Tutorials **16**(1), 393–413 (First 2014). https://doi.org/10.1109/SURV.2013.062613.00160
17. Ren, J., Guo, H., Xu, C., Zhang, Y.: Serving at the edge: A scalable iot architecture based on transparent computing. IEEE Network **31**(5), 96–105 (2017). https://doi.org/10.1109/MNET.2017.1700030
18. Riley, L and Mapp, G: yRFC3: The Specification of SP-Lite, http://www.mdx.ac.uk/our-research/research-groups/y-comm-global-research-group/y-comm-research
19. Rimal, B.P., Van, D.P., Maier, M.: Mobile edge computing empowered fiber-wireless access networks in the 5g era. IEEE Communications Magazine **55**(2), 192–200 (February 2017). https://doi.org/10.1109/MCOM.2017.1600156CM
20. Rimal, B.P., Van, D.P., Maier, M.: Mobile edge computing empowered fiber-wireless access networks in the 5g era. IEEE Communications Magazine **55**(2), 192–200 (February 2017). https://doi.org/10.1109/MCOM.2017.1600156CM
21. Sardis, F.: Exploring Traffic and QoS Management Mechanisms to Support Mobile Cloud Computing using Service Localisation in Heterogeneous Environments. Ph.D. thesis (2014)
22. Sardis, F., Mapp, G., Loo, J., Aiash, M., Vinel, A.: On the investigation of cloud-based mobile media environments with service-populating and qos-aware mechanisms. IEEE Transactions on Multimedia **15**(4), 769–777 (June 2013). https://doi.org/10.1109/TMM.2013.2240286
23. Sun, X., Ansari, N.: Edgeiot: Mobile edge computing for the internet of things. IEEE Communications Magazine **54**(12), 22–29 (December 2016). https://doi.org/10.1109/MCOM.2016.1600492CM
24. Tran, T.X., Hajisami, A., Pandey, P., Pompili, D.: Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges. IEEE Communications Magazine **55**(4), 54–61 (April 2017). https://doi.org/10.1109/MCOM.2017.1600863
25. Yu, Y.: Mobile edge computing towards 5g: Vision, recent progress, and open challenges. China Communications **13**(Supplement2), 89–99 (N 2016). https://doi.org/10.1109/CC.2016.7833463
26. Zhang, K., Mao, Y., Leng, S., Vinel, A., Zhang, Y.: Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks. In: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). pp. 288–294 (Sept 2016). https://doi.org/10.1109/RNDM.2016.7608300