Cooperative Mobile Edge Computing System for VANET-Based Software Defined Content Delivery

Article *in* Computers & Electrical Engineering · July 2018

6 authors, including:

Jafar Al-Badarneh
Jordan University of Science and Technology

Yaser Jararweh
Jordan University of Science and Technology

Mahmoud Al-Ayyoub
Jordan University of Science and Technology

Ramon Fontes
University of Campinas

Cooperative Mobile Edge Computing System for VANET-Based Software-Defined Content Delivery

Jafar Al-Badarneh[a], Yaser Jararweh[a,*], Mahmoud Al-Ayyoub[a], Ramon Fontes[b], Mohammad Al-Smadi[a], Christian Rothenberg[b]

[a] *Jordan University of Science and Technology, Irbid, Jordan*
[b] *University of Campinas, Campinas, Brazil*

Abstract

Next-generation smart cities and Internet of Things (IoT) are getting more mature in terms of services and infrastructure requirements. Multiple smart vehicle applications are being conceived these days, including road traffic, road safety and infotainment, all of which are suffering from the WAN-latency problem. In this paper, we propose a Vehicular Adhoc Network (VANET)-based Software-Defined Edge Computing infrastructure supporting content delivery services among connected vehicles. The proposed approach leverages network base stations to embed mobile edge computing (MEC) services closer to the vehicles. Our approach can enable the delivery of more competitive services with reduced-latency by utilizing cooperative MEC search strategy for vehicle to infrastructure (V2I) communications as well as utilizing vehicle-level caching for vehicle to vehicle (V2V) communications between peers. The framework prototype has been implemented as a clean extension of the Mininet-WiFi emulator. Preliminary results serve as validation of the proposed framework and point out the potential benefits of the approach in mitigating WAN-latency in VANET.

Keywords: Vehicular Ad-Hoc Networks, Content Delivery Service, Mobile Edge Computing, Software Defined Systems, Internet of Things, Intelligent Transportation Systems

[*]Corresponding author

Email addresses: `jwalbadarneh14@cit.just.edu.jo` (Jafar Al-Badarneh), `yijararweh@just.edu.jo` (Yaser Jararweh), `maalshbool@just.edu.jo` (Mahmoud Al-Ayyoub), `ramonrf@dca.fee.unicamp.br` (Ramon Fontes), `maalsmadi9@just.edu.jo` (Mohammad Al-Smadi), `chesteve@dca.fee.unicamp.br` (Christian Rothenberg)

1. Introduction

Over the last decade, an emergent need for satisfying today's interactive, real-time and context-aware applications has raised. Today's applications tend to provide a wide range of services for users/subscribers through their handsets with just a few clicks and few pennies (e.g., social network services, banking services, and others). On the other hand, smart phones are being built with constrained resources in terms of battery capacity, storage capacity, and processing power; hence, those services should be delivered in a way that satisfies these constraints. Mobile Cloud Computing (MCC) has emerged to meet the imposed constraints by preserving the quality of service (QoS) being provided while preserving the consumption rates for such devices [1]. The adaptation of MCC makes applications accessible for broader segments of users, as well as serving computation/analysis intense requests without exploiting smart phones' computational power. Thus, such resources can be utilized for other localized tasks.

The MCC paradigm effectively addresses the obstacles imposed by the resource scarcity of mobile devices [1, 2], but, it still faces a number of issues that can be summarized as follows [3]: `Low Bandwidth`, where resources available in wireless interfaces are much lower than in wired interfaces. `Heterogeneity`, where the MCC's job of maintaining an always-on connectivity, energy efficiency, and on-demand scalability, gets much harder when it comes to handling request from heterogeneous devices and access technologies. `Availability`: the always-connected property of MCC can contradict with the availability goals under congestion events.

Recently, many applications have gained the industry's attention as part of the realization of Internet of Things (IoT) and smart cities [4], where more things are going to be connected to the Internet and contribute to the massive amounts of data generated. Things can range from a simple still-sensors to moving vehicles. Connected vehicles are considered as an integral part of the Intelligent Transportation Systems (ITS) [5]. For many IoT applications (especially the ones related to ITS), having a low latency is crucial to their success. Resolving latency issues with such applications requires innovative methods to deliver such services in the context of service locality, where users are served by nearby base stations.

One of the earliest attempts to address such needs was the notion of Cloudlets [6], which brought cloud services closer to the user mitigating the latency as well as improving the availability. Cloudlets deliver cloud services to mobile users within limited ranges and constrained mobility spaces bounded by the WiFi coverage. However, it is incapable of adapting to the current demand for high computing applications. Consequently, an emergent need for another scalable solution gave rise to Mobile Edge Computing (MEC). The basic idea of MEC is to offload the traffic from Cloud servers to a set of edge servers, whose responsibility is to deliver the service at the edge of the network [7, 8].

MEC needs to introduce the concept of MEC-enabled base stations, where base stations with different types (e.g., eNodeB for LTE) will be capable of delivering services other than telecommunication services such as storage, computing, content retrieval and caching at the edge of the networks. Such incorporation will satisfy the requirements of the context-aware real-time applications, hence, enhancing the quality of the user experience as well as stacking up a new revenue stream for mobile operators.

Such deliverables can pave the way towards MEC oriented service enhancements. One example is Content Delivery Networks (CDN), which are intended to facilitate content sharing such as videos - which is the most viral content type nowadays - over the Internet to end users in a way that mitigates traffic congestion and delivers a better QoS. Such outcomes are satisfied by implementing a set of techniques (e.g., content caching and replication) in order to attain content localization closer to users' locations [9].

Aside from the previously mentioned aspects, managing and provisioning the network resources require more flexible and reliable paradigms. The authors of [10, 8] proposed to use the Software-Defined System paradigm to manage MEC in an efficient way compared to the traditional platforms. Software-defined systems paradigm tends to mitigate the management overhead by isolating the data plane from the control plane, as well as providing programmability of control.

This paper aims to provide a Software-Defined Content Delivery support for MEC systems in environments where mobility is considered very important. The proposed framework has been built as an extension to Mininet-WiFi [11], a well-known emulator for software-defined wireless networking. The framework relies on mobility and Vehicular Ad-hoc Network (VANET) supported by Mininet-WiFi with the extension to operate software-defined systems features along-side with the already supported SDN functions. The framework implements a Software-Defined Content Delivery for VANET over MEC (VSDCD-MEC).

The rest of the paper is organized as follows. Section2 gives more details about VANET and the use of MEC and software-defined control with them. The proposed framework is articulated in Section 3, whereas the experiments and results are discussed in Section 4. In Section 5, we conclude this work and present our future plans.

2. Related Work

Smart phones and wireless devices have been widely adopted over the past decade [12]. Users are relying more on them to do almost everything from simple and mundane tasks (that can be performed on the smart phone itself) to complex tasks related to their careers (which might require high bandwidth connectivity to cloud services) [13]. These devices are large in numbers and need to be always connected via high bandwidth connection using different technologies such as cellular networks, WiFi, WiMAX, etc. This makes it harder to guarantee reliable communication infrastructure among this huge number of devices (nodes) especially with the emergence of peer-to-peer (P2P) applications, where nodes are connected to each others in a reliable environment. Mobile Ad-hoc Networks (MANET) have been introduced to provide a reliable way to handle such overhead by simply providing the support for dynamic creation and configuration of networks, more of that is expressed in [14].

Lately, Vehicular Ad-hoc Networks (VANET) have been getting more attention. VANETs are becoming more popular (and even more essential) to connect vehicles with each other and to the Internet. In fact, cars are getting "smarter". Technologies like AI and Edge Computing have emerged with the evolution of networking infrastructures and high-speed processors, opening the gates for intelligent applications and services to be part of our vehicles, such as autonomous driving, infotainment, road safety and traffic management. Vehicles are being equipped with more sensors and compute capabilities as per to the demanded intelligence toward having smarter and healthier cities. Vehicles will all be connected in various ways; with each other *Vehicle to Vehicle* (V2V), or with the surrounding infrastructure *Vehicle to Infrastructure* (V2I) via nearby base-stations (e.g., eNodeBs, roadside units (RSU), access points (AP), etc.) [15].

There exist various research projects discussing current VANET applications as well as future expectations of how these applications can be facilitated to support road safety and effectively manage traffic flows [16, 17, 18]. Al Ridhawi et al. [19] discussed how to harness today's technological advancements with the emergence of IoT on our cloud systems. They proposed a model for making the data as close as possible to the edge of the network with effective resource allocation and selection methods. The results of their simulation showed the potential of the approach for satisfying 5G service latency requirements.

With the emergence of 5G networks and smart cities, Edge Computing has became more of a need to satisfy the requirements as well as maintaining our economies. Aloqaily et al. [20] addressed the need for a compelling Smart Vehicle as a Service (SVaaS) model that makes use of the smart objects in our cities, such as vehicles, which nowadays have great computational capabilities (Data transfer, environment identification, etc.) [21]. The authors also considered really important service-related aspects, which are the service cost and the trade-off with the quality of experience (QoE) in terms of delay for vehicular cloud computing. They have proposed a QoE-based service provisioning scheme which fulfills the requirements needed for the service.

Due to the large number of applications, the continuously increasing demands, the imposed communication overheads, and the heterogeneity of wireless infrastructure, traditional networking solutions are unsuitable to meet these requirements. Therefore, Software-Defined Networking (SDN) emerged as an appealing solution to these concerns. SDN aims to mitigate the provisioning and management overhead of the network resources as well as boosting the performance of the inter/intra vehicle communications through its abstraction and network programmability properties.

Employing SDN in the context of VANET has also attracted researchers for proposing more reliable solutions for connecting vehicles. He et al. [22] proposed an SDN based architecture for managing VANET considering heterogeneous wireless devices and RSU. They validated the effectiveness of their proposed architecture via a traffic-trace-based simulator. Another sophisticated work was proposed by Salahuddin et al. [23]. The authors proposed a novel RSU equipped with SDN in order to dynamically mitigate, migrate, replicate the cloud services provided by the RSU.

As noted in [24], there is a lack of practical implementations for deploying SDN into challenging contexts like Edge Computing and VANET. The previous work lead by Fontes et al. [24] took a step toward deploying SDN in the VANET context by introducing a preliminary support for emulating SDN VANET topologies inside Mininet-WiFi [11]. Such support benefits from the extended OpenFlow wireless devices to bring support for the multi-interface boards hocked up with vehicles. The authors in [25] proposed the SD-CRN as a general framework for exploiting the software defined systems concepts for cognitive radio networks which might be useful for fulfilling the V2V communication requirements specifically in dense areas.

In this work, we exploit the VANET support delivered by Mininet-WiFi [11] to provide software-defined content delivery support for MEC. We introduce a new Mininet-WiFi extension that tends to benefit from the abstraction as well as the centralized control delivered by SDN to operate nonnetworking functions. The extension implements the support for deploying MEC support inside base stations as well as delivering a proper control of the network resources at MEC level. The proposed MEC support allows base stations to deliver services to moving vehicles in range with low latency and high bandwidth since most of the requests will be delivered by vehicles' local MEC nodes. There are many functions that can be offloaded from cloud to the edge of the network where the vehicles reside.

3. VANET-Based Software Defined Content Delivery Support for Mobile Edge Computing (VSDCD-MEC)

SDN aims to avoid bundling the data plane with the control plane inside network devices, Instead, it provides a more reliable form of centralized control by separating them in a way such that the control is provisioned by a centralized controller at the network level where network devices act as forwarding devices. When it comes to adding software-defined support to aspects other than networking, additional control overhead is incurred since controllers are needed to serve more functions. Figure 1 provides the architecture for the proposed framework, which is divided into three layers: (i) data plane, (ii) control plane and (iii) application layer.

The data plane consists of a set of moving vehicles that are connected to the network through (V2I) via wireless interfaces (WiFi, 3G, LTE, etc.), as well as forming ad-hoc networks with other vehicles in range through (V2V) communication. As mentioned before, delivering MEC support and localizing content inside MEC nodes for closely connected vehicles require additional control in order to deliver the service with the expected latencies and bandwidth as well as to avoid flooding the networking controller with other tasks, which will affect their performance and, hence, will affect the whole network.

The control plane consists of a set of extended controllers, each of which performs specific tasks as needed by the requesting node while still consulting other controllers for any additional data required via the so-called East/West APIs. As depicted in Figure 1, there exist three types of customized controllers that will be used to provide content delivery service for vehicles at the edge of the network.

1. `SDMEC controllers`: they are responsible for distributing contents over base stations with MEC support (e.g., determining which chunks of data should be located inside a set of MEC

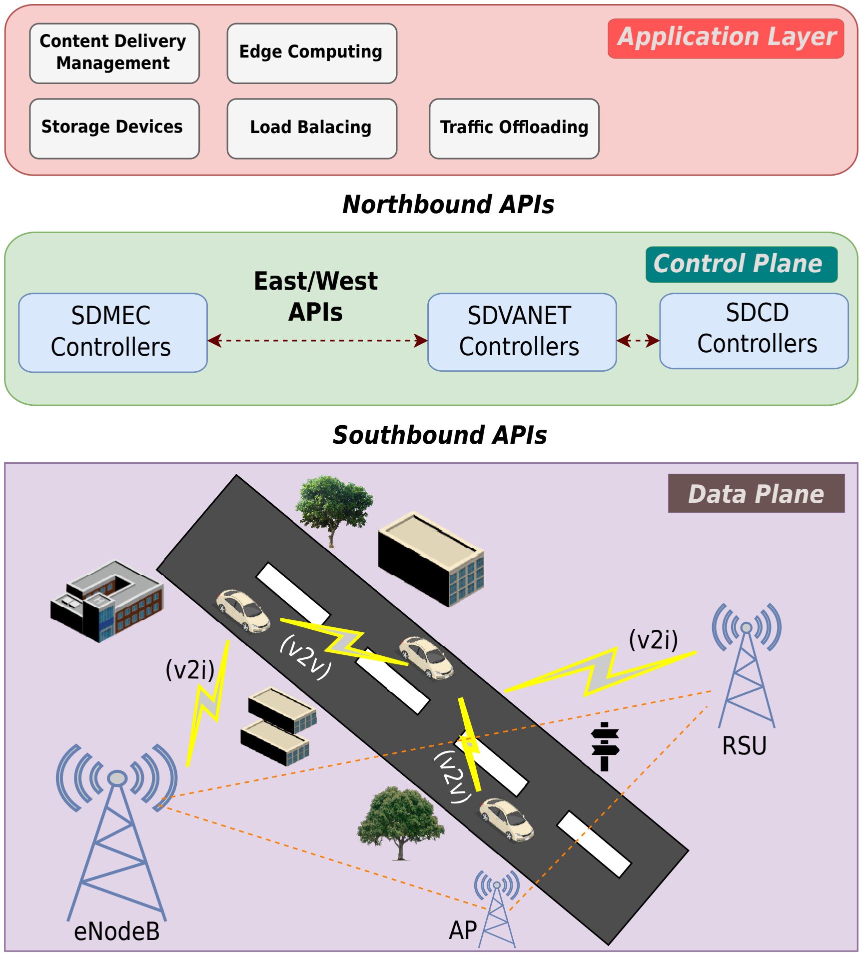


Figure 1: Proposed Architecture

nodes within a specified region in order to reduce the latency imposed by backhaul overhead -traveling with the request through the network to the cloud- and, of course, deliver services with high availability).

2. `SDVANET controllers`: they tend to deliver the proper networking control by managing data traffic injected into the vehicles as well as adjusting the forwarding tables inside the vehicles based on the dynamic changes in the topology of the VANET.

3. `SDCD Controllers`: their role is to serve vehicles requests for contents either locally at the local MEC level, or by cooperatively consulting neighboring MEC nodes for the requested

content.

Finally, the application layer presents the subset of functions that are delivered along with communication, networking, and security-related functions.

Algorithm 1 explains the policy for providing content delivery service for vehicles at MEC level. It depicts the search strategy and the cooperation among MEC nodes to serve vehicles' content requests.

Table 1: Algorithm Variable Set

Variable	**Description**
$LMec$	Vehicle's local MEC node
$ContentID$	Requested content identifier
C	Content resulted from search operation
$NMECs$	Set of neighboring MEC nodes sorted by distance in a descending order from vehicle's local MEC node
T	Threshold for number of MEC nodes

The proposed method works as follows: as the network gets initialized, SDMEC controller distributes contents over available MEC nodes in a way that fits the context of the requests being served by each region. A vehicle (car) tends to request content as per to a cloud service it uses. SDVANET controller is consulted to identify the MEC node the car is connected to via the *AssociatedMec()* method in line 1 of Algorithm 1. The SDCD controller searches the requested content inside the vehicle's local MEC node via the *Search()* method in line 3 of Algorithm 1. If found, the content will be delivered to the vehicle instantly. Otherwise, the MEC nodes need to cooperate with each other. Both SDCD and SDVANET controllers consult each other in order to search neighboring MEC nodes for the requested content before requesting the content from the cloud server all the way up through the network backhaul via the *Search()* method in line 12 of Algorithm 1. As it is not feasible to search all neighboring MEC nodes, a threshold for the maximum number of MEC nodes that should be consulted before offloading the request to the cloud server has been specified. This threshold helps in delivering the service with the lowest latency available. If the content is not found in all of the neighboring MEC nodes, the cloud server will be consulted to serve the request via the *SearchCloud()* method in line 20 of Algorithm 1.

Algorithm 1 VSDCD-MEC cooperative search

```
LMec ← Vehcile.AssociatedMec()
Found ← false
C ← LMec.Search(ContentID)
if C ≠ null then
    Found ← true
    return C
end if
while not Found do
    Count ← 0
    for Each MEC m in NMecs[] do
        if Count ≤ T then
            if m.Search(ContentID) then
                Found ← true
                return C
            else
                Count + +
                continue
            end if
        else
            C ← SearchCloud(ContentID)
        end if
    end for
    if not Found then
        C ← SearchCloud(ContentID)
    end if
    return C
end while
```

The following equations formulate the thresholding role for mitigating service latency.

$$\sum CRTT \geq \sum MRTT \tag{1}$$

where $CRTT$ represents request round trip time from the vehicle to the cloud servers and back.

$$\sum CRTT = Req_{vc} + LT + Res_{cv} \tag{2}$$

and $MRTT$ represents the request round-trip time from a vehicle through MEC node(s) and back as depicted in Equation 1.

$$\sum MRTT = \sum_{mec} Req_{vm} + LT + H + Res_{mv} \tag{3}$$

In these equations, Req_{vc} represents the propagation time from vehicle to cloud, Req_{vm} represents the propagation time from vehicle to MEC node, LT represents the lookup time needed to find the requested content, H represents next hub delay within each neighboring MEC node, Res_{cv} represents the response time from cloud to vehicle, and Res_{mv} represents the response time from MEC to the vehicle.

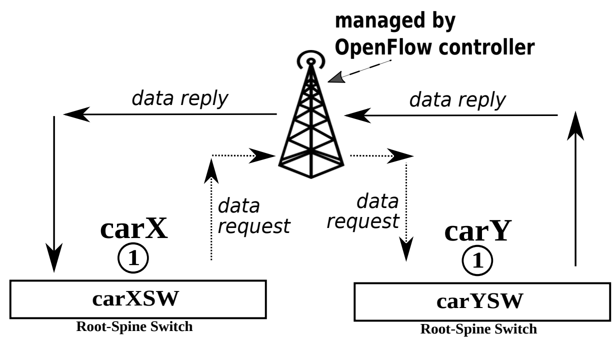
4. Experiments and Results

In order to demonstrate our approach for Software-Defined Content Delivery (SDCD), we present two use cases that consider both *Vehicle to Infrastructure* (V2I) and *Vehicle to Vehicle* (V2V) communications. Our SDCD approach is supported by Mininet-WiFi by means of a new extension and the vehicular network is supported by means of an existing integration between Mininet-WiFi and the *Simulation of Urban MObility* (SUMO) package [26].

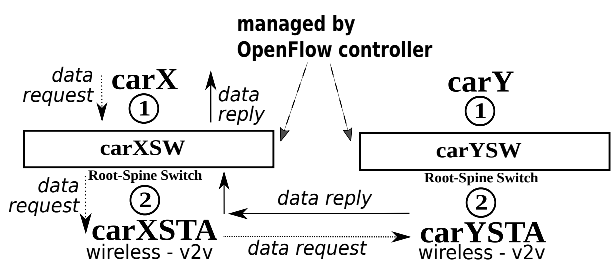
The whole idea behind incorporating content delivery services with MEC is to reduce the service time by cutting off the time needed for the request to travel all the way up to fetch the requested content from its origin (cloud server). Therefore, we have designed some experiments to evaluate the latency among multiple MEC setups through V2I and V2V interfaces inside vehicles.

4.1. Node Car Architecture

Figure 2 illustrates the flow pipeline processing extended from the node car architecture previously presented in [24], which brings the programmability of the SDN paradigm to the multi-interface devices on board of vehicles. The advantage of the proposed architecture is to allow researchers to define the vehicular data traffic behavior at the Root-Spine Switch and unlock any approach of OpenFlow control. From the user perspective, applications will be running inside vehicles (`carX`,`carY`). Vehicles will be able to communicate traffic between each other through the wireless V2V interface crossing the *Root-Spine Switch* allowing programmable flow management by an SDN controller. It is worth to mention that such flow management performed by SDN controller was already supported by Mininet-WiFi for V2I scenarios since this type of scenario requires a central device (e.g., *eNodeB*, *RSU* or *AP*) and the SDN controller can natively manage such type of devices.



(a) Flow pipeline processing for Vehicle to Infrastructure communication



(b) Flow pipeline processing for Vehicle to Vehicle communication

Figure 2: Flow pipeline processing

Two sets of experiments commenced in this study presenting use cases for V2I and V2V communication among cars and how SDCD works along with VANET to deliver requested services at MEC level. The simulation environment for both sets of experiments is configured to work with

SUMO [26]. OpenStreetMap[1] is used to generate our simulation map. The map is chosen randomly covering a specific area around New York City, NY, USA. NetConvert[2] is used to generate road network flow for the selected map in order to configure vehicles' routes.

4.1.1. Vehicle to Infrastructure

The 'I' in the term V2I represents the infrastructure that is open for communication with vehicles. It represents the network edge (Access Points (AP), Road Side Units (RSU), or Base Stations (BS)) that are close to the vehicles. Our experiment stresses out embedding content delivery capabilities within the edge. The experiment discusses V2I and the role of SDCD in mitigating request serving latency. The proposed node car architecture for V2I communication is depicted on Figure 2(a).

The use case mainly showcases the ability of our proposed system to deliver content to a vehicle with the minimum latency possible, by offloading the vehicles' requests from content sources (cloud) to the closer edge (base stations) that hosts the requested content. For the sake of delivering a proof of concept, the cloud server is populated with a set of ten contents that vary in size (*Min:850, Max:9000, Mean: 4750* storage units). The server is connected to a network of base stations with MEC capabilities via a switch. The base stations are distributed in the simulation environment. A set of ten moving vehicles are instantiated inside the environment and their connectivity provisioned through VANET. Contents are distributed evenly among MEC nodes where each MEC node is set to have a local copy of one content with an ID that matches MEC node ID in order to simplify the experiment. E.g., eNodeB1 has Content1, eNodeB2 has Content2, and so on. We have evaluated our proposed experiment for various numbers of MEC nodes in terms of service latency. `CarX` is set to be connected to `MEC1`. It issues ten separate requests for all contents. Each request is processed as follows: (i) the MEC node tries to search its local storage to see whether the requested content exists or not; (ii) If not, the controller perform cooperative search inside the neighboring MEC nodes. The idea of search thresholding is also applied here, where four MEC nodes are set to be searched at most for the requested contents; (iii) if not found, the request will be handed over to the cloud storage to be served.

[1] `https://www.openstreetmap.org/`
[2] `http://sumo.dlr.de/wiki/NETCONVERT`

4.1.2. Vehicle to Vehicle

The second set of experiments deliver a proof of concept for utilizing SDCD for V2V communications, presenting Mininet-WiFi capabilities for the node car architecture (Figure 2(b)). Leveraging SDCD for V2V communication serves the case where there exist vehicles with poor infrastructure connection which prohibits them from receiving/delivering services effectively. Vehicles in such cases can still get the intended services by leveraging the V2V communication scheme and connect to other vehicles in range which have a better connection to the base station. This would potentially enhance both QoS and QoE. In terms of the service cost, it would be cheaper for vehicles to be served right through their neighbors instead of consulting base stations or the cloud server for the service.

The proposed SDCD for V2V also delivers support for vehicle-level caching, where each vehicle tends to cache contents for serving near-future requests from neighboring vehicles. Vehicle-level caching can be leveraged by V2V communication among the vehicles to deliver better service time, cutting down the trip a request might go through to be served (Local MEC, Neighboring MECs, Cloud server) acting as the closer serving edge node.

Figure 3: V2V Experiment.

In this experiment, we shed light on the aforementioned case, where our simulation holds tens of running cars distributed over the map of New York City. For better understanding of the use cases presented in this section, we pause the simulation running in SUMO after 28 seconds of its start. At T=28, we chose *Car2* and *Car6* to show how V2V can be utilized to deliver the requested contents to *Car6*. The V2I and V2V communications are provisioned by the `bgscan` module. This module is responsible for issuing background scans using (wpa_supplicant)[3] for the purpose of roaming within an ESS (i.e., within a single network block with all the AP using the same SSID). As shown in Figure 3, both cars (*Car2* and *Car6*) are in range of *eNodeB5*. `bgscan` scans show that *Car6* connection to eNodeB5 via `Car6-wlan` interface is very poor with signal strength -90.00 dBm, while it has a better connection to the neighboring *Car2* via V2V interface `Car6STA` with signal strength -67.00 dBm. *Car2* has a stronger connection to *eNodeB5* in comparison with *Car6* since *Car2* is closer to the base station than *Car6* with signal strength -77.00 dBm.

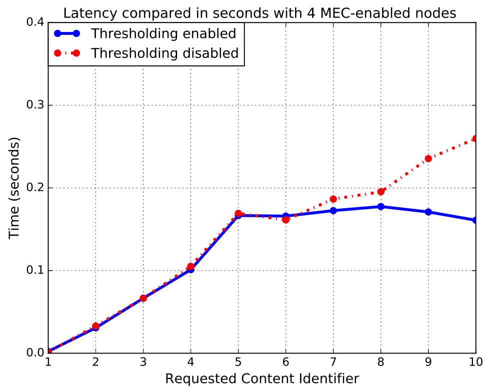
Car6 is intended to request the whole set of contents. Given the network stats resulting from bgscan, requests will be served through *Car2*. Since vehicles can leverage the ability to serve requests from their cache. In-vehicle cache will be consulted first for the inbound requests before escalating to the closer MEC node (*eNodeB5*), resulting in a better service time for *Car6*'s requests with cached responses.

4.2. Results

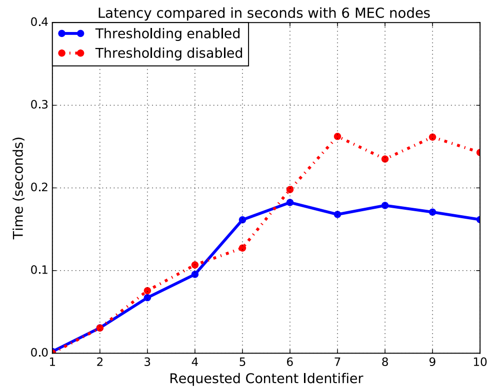
In this section, we evaluate service latency for both V2I and V2V showing the significant role of mobile edge computing in mitigating WAN-Latency for vehicle-centric applications.

For V2I, incorporating the proposed edge approach should facilitate faster content delivery for the requesting vehicle compared to delivering contents from the core of the cloud. Figure 4 compares the response time needed to deliver the requested contents when both the thresholding technique enabled and disabled with different threshold values. As shown in the subfigures of Figure 4, contents [1-4] are expected to be served through the neighboring MEC nodes regardless of the thresholding behavior due to the facts that (i) they are located closer to the requesting vehicle and (ii) the minimum threshold used is within the minimum threshold specified in the experiment. The effect of the thresholding technique starts to pay off after attempting to consult more neighboring

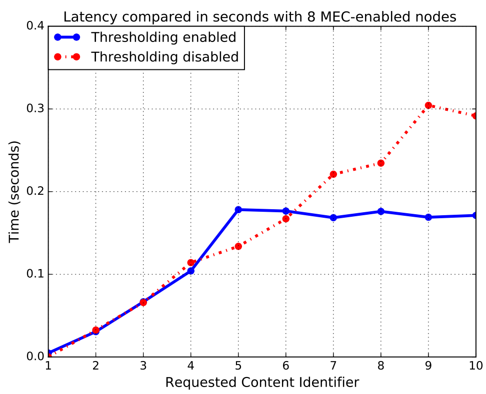
[3] `https://w1.fi/wpa_supplicant/`



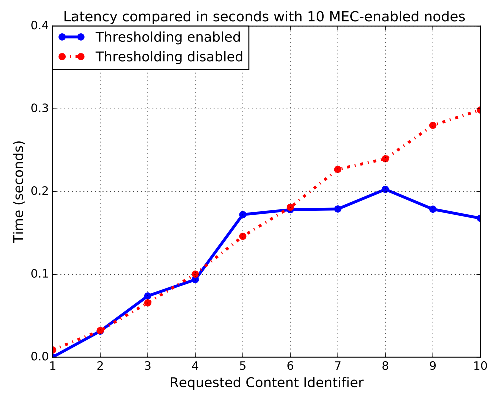
(a) Latency compared in seconds with 4 MEC nodes



(b) Latency compared in seconds with 6 MEC nodes



(c) Latency compared in seconds with 8 MEC nodes



(d) Latency compared in seconds with 10 MEC nodes

Figure 4: The effect of cooperative search thresholding for SDCD

MEC nodes exceeding the threshold for contents [5-10]. The main role of adapting such technique is to cap the service latency under the baseline of requesting contents from the core of the cloud. It becomes infeasible to consult all neighboring MEC nodes looking for a content that is not located in a nearby location.

With this scenario, referring to equation 1, giving up the lookup process among neighbors becomes more beneficial in terms of service latency and better QoS. With all MEC setups, using thresholding gives better results compared with not using thresholding in terms of delivering the requested content with the least latency possible.

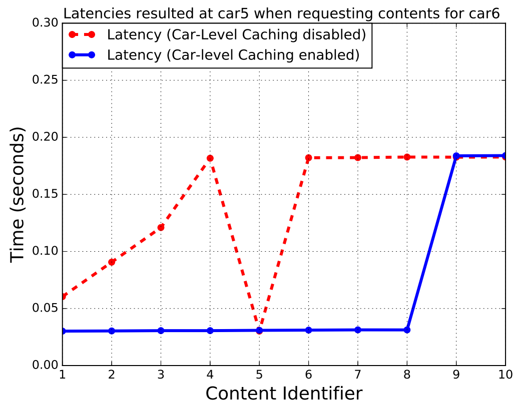


Figure 5: Resulted latencies from v2v experiment

Along with the proposed thresholded cooperative search method, an interesting aspect with regards to the threshold value leaves us with the question of what is the best threshold to be used for a given number of MEC nodes. This process actually involves many factors to be considered in order to decide what is best for a given topology. This can be thought of as an optimization problem: for a given topology setup having `X` number of MEC-enabled base stations, what is the best threshold value for consulting neighboring MEC nodes, taking into consideration many factors, such as density of the environment, level of interference, bandwidth, etc.? We plan to cover more about such aspects in future work.

For V2V, we have measured the latencies for the same sorts of requests experimented in V2I, but, this time, comparing the effects of caching support inside vehicles. As shown in Figure 5, enabling vehicle-level caching have significantly resulted in a better service time. Contents [1-8] are served from `Car2`'s cache, whereas the remaining contents are served from their origin `Cloud Server` after several failed attempts to retrieve the content through neighboring MEC nodes.

With vehicle-level caching disabled, the same technique applied for V2I is performed. As observed in the figure, there is a dropoff in the latency for content [5]. The reason behind such low latency is related to the fact that the content is localized at the MEC node to which the vehicle is connected (Local MEC node). Since `car2` is connected to `eNodeB5`, content located inside `eNodeB5` will be retrieved quickly. We also notice that the first three requests are served via neighboring MEC nodes, which are within the search threshold specified of four MEC nodes. The remaining requests are served by the source (*cloud server*), as they are not found inside the local MEC node nor within the neighboring ones with the specified threshold.

This work can be reproduced by using the code available at our public repository[4], and the explanatory video[5] showing how the experiments are conducted.

5. Conclusions and Future Work

In this paper, we have discussed our efforts to introduce Software-Defined Edge Computing approach for delivering content requested by vehicles at the edge of the network. The proposed framework introduces the ability for incorporating mobile edge computing capabilities inside network base stations allowing vehicles to communicate various services via their V2I interfaces more instantly and with better qualities and minimal WAN-Latency. We also proposed a vehicle-level caching technique to enable vehicles to communicate with neighboring ones on the roads seeking even more instantaneous services through V2V interfaces. This technique enables high service availability as well as better QoS. The proposed framework is built as an extension of Mininet-WiFi and evaluated using a set of illustrative experiments. The experiments compare content delivery service latency at both the core cloud and the edge of the network. The results of the experiments show the potential benefits of our proposed approach in building applications for next-generation vehicles, which are considered an integral part of the realization of smarter cities. These applications include road safety and quality-focused transportation applications. The proposed capabilities can be extended to include more reliable edge systems for vehicle in the field, such as security and storage. We will cover more of that in our future work in addition to broader open research questions around network slicing for the V2X ecosystem [27].

References

[1] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013, including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X12001318

[4] `https://github.com/jafar-albadarneh/SDS-Mininet-WiFi/tree/27f098d9edf29f82bc2afb384f9e7c55bb855b2f`

[5] `https://www.youtube.com/watch?v=skyHDlHdwcs`

[2] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 4–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863107

[3] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013. [Online]. Available: http://dx.doi.org/10.1002/wcm.1203

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[5] H. Hartenstein and L. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications magazine*, vol. 46, no. 6, 2008.

[6] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," in *6th International Conference on Mobile Computing, Applications and Services*, Nov 2014, pp. 1–9.

[7] U. Drolia, R. Martins, J. Tan, A. Chheda, M. Sanghavi, R. Gandhi, and P. Narasimhan, "The case for mobile edge-clouds," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, Dec 2013, pp. 209–215.

[8] J. Al-Badarneh, Y. Jararweh, M. Al-Ayyoub, M. Al-Smadi, and R. Fontes, "Software defined storage for cooperative mobile edge computing systems," in *2017 Fourth International Conference on Software Defined Systems (SDS)*, May 2017, pp. 174–179.

[9] R. G. Garroppo, G. Nencioni, L. Tavanti, and B. Gendron, "The greening potential of content delivery in residential community networks," *Computer Networks*, vol. 73, pp. 256 – 267, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614003041

[10] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa, "Sdmec: Software defined system for mobile edge computing," in *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, April 2016, pp. 88–93.

[11] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 384–389.

[12] M. Kakihara, "Grasping a global view of smartphone diffusion: An analysis from a global smartphone study." in *ICMB*, 2014, p. 11.

[13] Y. Wen, X. Zhu, J. J. Rodrigues, and C. W. Chen, "Cloud mobile media: Reflections and outlook," *IEEE transactions on multimedia*, vol. 16, no. 4, pp. 885–902, 2014.

[14] A. Bakshi, A. Sharma, and A. Mishra, "Significance of mobile ad-hoc networks (manets)," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 4, pp. 1–5, 2013.

[15] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[16] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer, "Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing.* ACM, 2001, pp. 259–262.

[17] M. Guo, M. H. Ammar, and E. W. Zegura, "V3: A vehicle-to-vehicle live video streaming architecture," *Pervasive and Mobile Computing*, vol. 1, no. 4, pp. 404–424, 2005.

[18] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.

[19] I. A. Ridhawi, N. Mostafa, Y. Kotb, M. Aloqaily, and I. Abualhaol, "Data caching and selection in 5g networks using f2f communication," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–6.

[20] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "A generalized framework for quality of experience (qoe)-based provisioning in a vehicular cloud," in *2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, Oct 2015, pp. 1–5.

[21] M. Aloqaily, I. A. Ridhawi, B. Kantraci, and H. T. Mouftah, "Vehicle as a resource for continuous service availability in smart cities," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–6.

[22] Z. He, J. Cao, and X. Liu, "Sdvn: enabling rapid network innovation for heterogeneous vehicular communication," *IEEE network*, vol. 30, no. 4, pp. 10–15, 2016.

[23] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for rsu clouds in support of the internet of vehicles," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 133–144, 2015.

[24] R. D. R. Fontes, C. Campolo, C. E. Rothenberg, and A. Molinaro, "From theory to experimental evaluation: Resource management in software-defined vehicular networks," *IEEE Access*, vol. 5, pp. 3069–3076, 2017. [Online]. Available: https://doi.org/10.1109/access.2017.2671030

[25] Y. Jararweh, M. Al Ayyoub, A. Doulat, A. A. A. Al Aziz, H. A. B. Salameh, and A. A. Khreishah, "Sd-crn: Software defined cognitive radio network framework," in *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. IEEE, 2014, pp. 592–597.

[26] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[27] C. Campolo, R. Fontes, A. Molinaro, C. E. Rothenberg, and A. Iera, "Towards 5G network slicing for the V2X ecosystem," in *Workshop on advances in Slicing for Softwarized Infrastructures (S4SI 2018) IEEE conference on network softwarization (NETSOFT)*, Jun 2018.

Authors' Biographies

Jafar Albadarneh is a Research Assistant in the Faculty of Computer Science at Jordan University of Science & Technology, Amman Jordan, where he received his masters degree in Computer Science. His research interests include software-defined systems, edge computing, machine learning and IoT.

Yaser Jararweh received his Ph.D. in Computer Engineering from University of Arizona in 2010. He is currently an associate professor of Computer Science at Jordan University of Science and Technology, Jordan. He has co-authored many technical papers in established journals and conferences in fields related to cloud computing, HPC, SDN and Big Data. He is chairing many IEEE events such as AICCSA, SDS, FMEC, ICICS, SNAMS, BDSN, IoTSMS and many others.Yaser served as a guest editor for many special issues in different established journals. Also, he is the steering committee chair of the IBM Cloud Academy Conference. He is associate editor in the Cluster Computing Journal (Springer), Information Processing and Management and others.

Mahmoud Alayoub Received his Ph.D. in computer science from Stony Brook University in 2010. He is currently an associate professor of computer science at Jordan University of Science and Technology (JUST). His research interests include cloud computing, high performance computing, machine learning and AI. He is the co-director of the High Performance and Cloud Computing research lab at JUST.

Ramon Dos Reis Fontes s currently pursuing the Ph.D. degree in electrical and computer engineering with the State University of Campinas, Campinas, Brazil. He was a visiting Ph.D. student with the DIANA Team, Sophia Antipolis, France, in 2016. He is currently a Professor of Computer Science with the Federal Institute of Education, Science and Technology of Bahia, Brazil. His research interests include software-defined wireless networking, OpenFlow, and future Internet architectures.

Mohammad Alsmadi Received his Ph.D. in Computer Science from Graz University of Technology in 2012. He is currently an assistant Professor at Jordan University of Science and Technology, Computer Science department. He has co-authored several technical papers in established journals and conferences in fields related to Social and Semantic Computing, Knowledge Engineering, Natural Language Processing, Technology Enhanced Learning.

Christian Esteve Rothenberg is an Assistant Professor in the Faculty of Electrical & Computer Engineering (FEEC) at University of Campinas (UNICAMP), Brazil, where he received his Ph.D. and currently leads the Information a& Networking Technologies Research & Innovation Group (INTRIG). His research activities span all layers of distributed systems and network architectures and are often carried in collaboration with industry, resulting in multiple open source projects in SDN and NFV among other scientific results. Christian has two international patents and over 100 publications, including scientific journals and top-tier networking conferences such as

SIGCOMM and INFOCOM