Congestion Mitigation in Densely Crowded Environments for Augmenting QoS in Vehicular Clouds

Aloqaily, Moayad; Balasubramanian, Venkatraman; Zaman, Faisal; Ridhawi, Ismaeel Al; Jararweh, Yaser

DOI
10.1145/3272036.3272038

Publication date
2018

Document Version
Final published version

Published in
DIVANet'18

Citation (APA)
Aloqaily, M., Balasubramanian, V., Zaman, F., Ridhawi, I. A., & Jararweh, Y. (2018). Congestion Mitigation in Densely Crowded Environments for Augmenting QoS in Vehicular Clouds. In *DIVANet'18 : Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications* (pp. 49-56). Association for Computing Machinery (ACM). https://doi.org/10.1145/3272036.3272038

Congestion Mitigation in Densely Crowded Environments for Augmenting QoS in Vehicular Clouds

Moayad Aloqaily
Gnowit Inc.
Ottawa, ON, Canada
Moayad@gnowit.com

Venkatraman Balasubramanian
Delft University of Technology
Delft, Netherlands
V.balasubramanian@tudelft.nl

Faisal Zaman
Ciena Networks
Ottawa, ON, Canada
Fzaman@ciena.com

Ismaeel Al Ridhawi
American University of Middle-East
Eqaila, Kuwait
Ismaeel.AlRidhawi@aum.edu.kw

Yaser Jararweh
Duquesne University
Pittsburgh, PA, USA
Jararwehy@duq.edu

ABSTRACT

Parking lots in densely crowded environments such as stadiums, theaters and hospitals provide great opportunities for vehicular cloud services. A cloud environment formed by individual vehicles, where each vehicle offers its resources as a service has shown feasible practices in 5G network scenarios. Moreover, resource management in 5G must be achieved in accordance with user-centric QoS requirements. In alignment with this, a key enabler of the user-centric service scheme is Network Slicing. The formation of multiple slices in such a dense environment, the congestion between sender and receiver, and resource management and allocation are topics of current research. This paper has the following contribution: First, a framework of Vehicular Clouds being restricted to individual slices in 5G cellular networks is proposed. Second, a queuing strategy for congestion control in a densely crowded environment such as parking lots is designed. Finally, a resource allocation algorithm that enables maximum matching between the tasks to be executed and the candidate slices is developed. The novelty of this approach comes from the fact that congestion control is performed at the Access Points (AP). We do this by introducing a control module that makes queuing decisions at the time of request arrival. By incorporating control module in AP, our aim is to provide AP resources in terms of transmission period to different slices, thereby, allowing WiFi resources to be shared along with the 5G radio resources. The performance benefits of the proposed solution has been investigated through simulation tests.

KEYWORDS

Vehicular Clouds; Smart Vehicles; Congestion; QoS; 5G network; Crowd Management

ACM Reference Format:
Moayad Aloqaily, Venkatraman Balasubramanian, Faisal Zaman, Ismaeel Al Ridhawi, and Yaser Jararweh. 2018. Congestion Mitigation in Densely Crowded Environments for Augmenting QoS in Vehicular Clouds. In *8th ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet'18), October 28-November 2, 2018, Montreal, QC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3272036.3272038

1 INTRODUCTION

With today's widespread crowded environments such as live sporting events and theaters, Internet connectivity demands have led to an increase in access to mobile networks. For instance, in stadiums, due to the heavy utilization of the available WiFi points, there are always issues pertaining to congestion and networks breaking down. On an average, over 6.23 TB of data was transferred over WiFi network at the Super Bowl 2015 according to [1]. To elaborate further, it was quoted that "4G LTE data usage in and directly outside the stadium on game day was 754 GB". Furthermore, a report from T-mobile stated that of the 430 GB of data used in the stadium, 33% was for web browsing, 24% for social media and 17% for video or audio streaming. Due to this excessive network usage in such densely crowded environments, it has become imperative to find a way to exercise novel strategies to overcome congestion and network break-downs. To this end, the authors in [2, 3] investigated how devices carried by individual attendees themselves provide a network infrastructure that is utilized by the stadium attendees for sharing information. We explore a similar scenario in this paper, with the exception that crowded environment attendees can utilize the parking lot vehicular Cloud constructed from smart vehicles to gain access to the cloud. This will tremendously reduce both network congestion and delay.

Vehicular cloud applications have been investigated in the past [4, 5]. Nevertheless, a cataclysmic effect is observed with the increase in the number of vehicles observed at the parking lots. The proliferation of vehicles and the network interfaces as a response to the IoT-buzz, has resulted in a tremendous network traffic explosion as justified by the statistics. So, research in the past has provided solutions with respect to offloading of traffic. There are multiple ways of offloading network traffic as seen in the literature [6, 7]. Of particular interest to us in this paper is the multi-path TCP (MPTCP) enabled Vehicles i.e. vehicles equipped with multiple

---

DIVANet'18, October 28-November 2, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5964-1/18/10…$15.00
https://doi.org/10.1145/3272036.3272038

interfaces. Arguably, MPTCP has shown improved reliability by dynamically balancing traffic in the network. However, the impact of cellular and WiFi path coupling has resulted in questionable congestion control for certain scenarios [8]. The 5G network is promising quality of service guarantees for such scenarios. However, all of the QoS requirements have become user-centric. Therefore, we consider user-enabled vehicular clouds grouped as a network slice for providing service to users with low latency. Here, the key enabler of the user-centric service is the Network Slicing technique which is also a fundamental feature of 5G [9].

There are several on-demand network services that can be enabled based on slicing techniques. One such service that we propose is a user-defined vehicular cloud service or more generally known as a vehicle-sourced compute environment. In this technique a vehicle makes a discovery and offers the residual resources within it as a service, each residual resource together forms a virtual composition providing a service to the consumer (e.g. a request coming from the attendees in a stadium). We group such a service in a network slice, thereby resulting in isolation of service and maintaining QoS. As seen in previous works, a slice formed [10, 11, 12] provides a user-centric service that allows programmatic control for the management entity which forms and maintains the slice. Furthermore, in an environment like a stadium it becomes imperative that the location of the Access Point (AP) fulfills the job of having a global resource view and enables allocation of resources. Hence, in this paper, deriving inspiration from software defined networks (SDN), we propose a framework for access point (AP) specific resource management that assists device clouds formed within the purview of APs in the region.

1.1 Objective

This paper considers a solution that is restricted to crowded environments with dense number of users and vehicles, where cloud service provisioning takes place on the spot at the location. For instance, as observed in [2], it is common that during a game, a person on one side of a stadium cannot get the same view of the game-play as that of the person on the other side of the stadium. In such cases, both of these persons can request a network slice within the stadium environment from the parking lot vehicular cloud that is at a one-hop distance instead of requesting a cloud service from a far-off data center. These slices are homogeneous compositions in nature and used based on the location of the vehicles in a stadium who have agreed to form a slice. As a cloud service request to a data-center would take up considerably more round trip time, a device cloud formed with collaborative support from the vehicles at the game site can provide a storage spot at the same place during the game. We consider only homogeneous resources but also extrapolate our evaluation for heterogeneous resources. In a time-bound application scenario, mobile vehicular clouds provide many benefits [13]. However, in pursuit of this, we end up having multiple issues.

1.2 Problems

1. As the requests are made to the APs, WiFi access points need to provide proper QoS as they act as local collection points for the composition service formations. Traditionally, slicing techniques have only been proposed for core networks, there are challenges that need to be investigated for a WiFi transmission-time resource shared slice coexisting with the nearest base-station resources.
2. In vehicular scenarios, there would be a constant degradation in QoS if the nodes fails to service a request within a specified time duration due to network traffic or vehicles moving out of a composition while the request is made (e.g. an attendee leaves early from the stadium taking his vehicle from the parking). There is no control to track the users movements. We restrict ourselves to situations where the information of users departing with their vehicles is first noted by the AP.

1.3 Contribution

In view of the above problem, the following are the main contributions of this paper:

1. We design a system for in-venue network where service compositions can change dynamically. An access point specific architecture is designed that assimilates composition information for slice specific service allocation. Here, we consider bandwidth and transmission period as resources shared between slices. A dedicated amount of transmission period is alloted by AP per slice.
2. We develop a congestion control mechanism at the AP that performs slice specific $data - flow$ selection with a delay aware queue. We refer to this mechanism as Jockey-Policy, such that, a packet entering the queue is moved to another queue based on delay requirements. As we are considering Multipath TCP scenarios, their needs to be proper sub-flow rescheduling in order to make the sub-flows aware of the $data - flow$.
3. We introduce an adaptive matching algorithm that enables maximum resource to job matching by examining a bi-partition graph. In doing so, our only goal is to ensure that every job is allocated to a candidate slice such that its execution is guaranteed.

The rest of the paper is organized as follows, Section 2 and Section 3 provide related works and basic concepts involved in this work respectively. Section 4 discusses the system design. Section 5 introduces the resource allocation algorithm. In Section 6, we elaborate on the experimental evaluations. Finally, in Section 7, we provide some future works and concluding remarks.

2 BACKGROUND AND RELATED WORK

The current trend in growth of smart vehicles and vehicular cloud usage has led to major developments [4][14]. In addition, vehicle classification is an important item in ITS which greatly help in resource allocation and management when employing applications such as traffic control, tracking, and other security applications [15]. Moreover, content delivery network is a new mechanism where vehicular nodes can act as providers and make the delivery process faster than usual [16]. Authors in [17] have elaborated on application offloading in vehicular clouds. In this, a service centric architecture is proposed that provides customized services to the user. Further, [18] shows performance characteristics of Vehicular clouds and its volatility that results in various challenges. With the well-known premise of long propagation delays besetting the

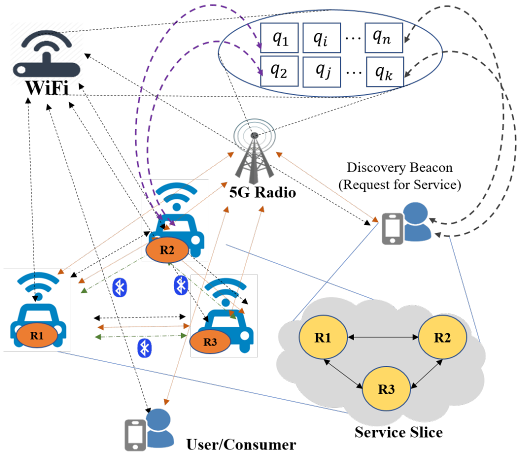


Figure 1: Coexistence of 5G and WiFi. Mainly bandwidth resources of 5G and transmission time of the WiFi AP are considered as factors for slice resource sharing. For example, a slice uses the channel available by CSMA, then a fraction of the time of medium access is shared between slices.

Mobile Cloud Computing paradigm, current network architectures have begun integrating the cloud computing technology into mobile networks. This is referred to as the Mobile Edge Computing paradigm. Similar to edge computing, smart vehicles are capable of providing processing, storage, and networking at the edge of the mobile network with the ability to execute applications and services within close proximity of mobile users [19][20]. Quality of Experience (QoE) model has been employed with aforementioned model to guarantee user's satisfaction [21]. Similarly, the authors in [22] proposed a scheme for continuous availability of diversified cloud services targeting vehicular cloud users through a cluster-based and TTP model. Their solution incorporate the use of QoE as well. However, Our focus in this paper is primarily on low-latency applications for mitigation congestion in crowded environments. In [23], Rost et al. propose that the network slice represents "independent virtualized end-to-end network that is self contained with customized functions". We adapt this definition to our framework partially by creating isolation between individual composition slices. In doing so, we run multiple logical networks as independent storage entities inside a stadium which can be accessed from all sides of the stadium such that attendees get the video shots they are looking for from the storage environment at the parking lot.

With respect to scheduling in such environments, in [24] the authors propose a linear programming based model for task scheduling. This model performs efficient mapping of resources but is pre-assigned based on composition score. However, in this work, the allocation is primarily a maximum matching approach that allocates resources based on a bi-partitioned graph with an adaptive Berge's algorithm [25]. These resource units are nothing but logical computation slices formed. In [10] the authors propose a work that is closer to our approach, but the strategy of avoiding congestion is different. We follow a more selective request strategy that does not cater to an "always accept" methodology owing to limited resource availability. We do not make comparisons of this algorithm in this paper as it is out of the scope of this research work.

3 CONCEPTS

In this section we elaborate on fundamental concepts related to our system design.

3.1 Multipath TCP

Vehicle capabilities have grown over the years with built in services and sensors [26]. A protocol that enables maximum utilization of these interfaces is Multi-Path TCP. Multipath TCP [8] is an extension of TCP that allows multiple network sub-flows with one underlying TCP connection. For vehicular communication, an MPTCP approach is observed in [27]. Essentially, the benefit of this protocol comes to the forefront in application throughput, wherein network's resources are put to optimal usage. The major problem here is that multipath TCP requires a decision on which sub-flow the segments should be scheduled. Furthermore, in a Mobile Vehicular Cloud Scenario, each vehicle involved in a transport layer communication would be exchanging multiple control messages, along-with service continuity tokens during mobility resulting in heavy buffering in the on board vehicle. Our intention in this work is to establish acceptable QoS levels and maintain service isolation so that execution of one job does not affect another.

3.2 NSaaS: User-centric Network Slicing and Mobile Vehicular Clouds

A key paradigm towards providing independent isolated operating instances for specific network functions is called Network slicing. This paradigm allows programmatic control. Thus, it can be regarded as a "logical end-end user construct that is self-contained with customized functions including the usage of network related functions" that is used for providing services to the subscribed user vehicle. A major challenge that arises in crowded environments is how to make the Mobile vehicle and the user aware of the network slice. To this end, our proposed Wireless LAN control application acts as the middle-ware that keeps account of the slice resources and the vehicle client (at the user) chooses a reliable slice based on its network location.

3.2.1 Scenario. : Figure 1 shows a request beacon sent to an access point for discovering nearby vehicles (known vehicles of 2-3 friends) who volunteer their resources (R_1, R_2, etc) for NSaaS provisioning. This is a publish and subscribe mechanism followed in state of the art such as [3]. With the advent of multiple interfaces, an added benefit is observed where each of the interfaces in the vehicles collaborate and form an vehicular cloud. We propose such a scenario where MPTCP enabled vehicles together form a network slice. Here, once the request is received, the AP will assist in finding the resources. The ownership of the resources are with vehicles who are readily present in the vicinity and are ready to volunteer their services. Once received, the diverse collection of resources (natively stored) are composed into a usable device cloud infrastructure via service APIs [3]. Further, we need an effective algorithm to manage the slices and the requests that enable utility maximization. Typically, each attendee posts a video of his view in a "cloud-bucket"

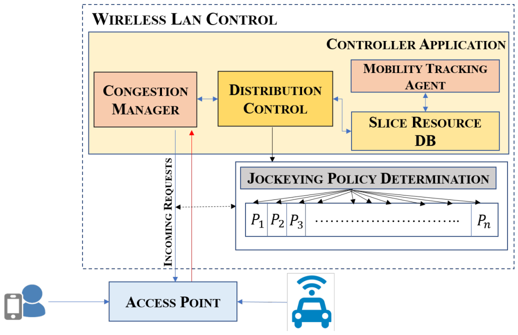


Figure 2: System design architecture

which can be accessed by anyone in the stadium then the problem can be solved. This *cloud − bucket* is a volunteer vehicular cloud that is formed purely based on the groups of vehicles who wish to participate in the cloud by offering their residual resources. In such a case, there are no overheads of having a security layer (as these groups of people are known to each other) or requesting cloud support that would entail a higher round trip time. This concept is referred to as NSaaS (Network Slice as a Service).

4 SYSTEM DESIGN

We consider 5G cellular base stations (BS) to be the slice service anchor points. This means that while WiFi access points and cellular base stations follow widely different queuing strategies, the anchor points enable seamless application maintenance. The BS associates a buffer and a dedicated channel for each user i.e. each user can access a small cellular bandwidth. This accounts for contrasting delay and channel characteristics. The WiFi links on the other hand are the main service access agents and need to maintain a good QoS by servicing enough requests. Now, between the two buffers our interest is closely related to WiFi APs because of two reasons:

1. As our in-venue service is specific to the time frame in which the games are played and watched by the attending crowd in case of the stadium scenario, it becomes imperative that the slice formation service requests are met within the AP coverage of the stadium.
2. To continuously discover end-users (i.e. vehicles) willing to offer services at the game site, all users need to be connected to the APs in order to be part of the collaboration. Therefore, we assume the cellular buffer to be complying with requirements for coexistence [28].

Therefore, the losses incurred at each AP can demonstrate the quality of service granted by a composition. Furthermore, when investigating the congestion window, it was observed that when there are approximately two consecutive congestion window leaps in the cellular links, multiple congestion window leaps occur in WiFi. Essentially, based on this we employ a heuristic to simultaneously manage the two congestion windows as seen in the following sections.

4.1 Application Controller

Figure 2 depicts the proposed architecture which consists of the following modules.

- Access Point (Transmit/Receive Module) : This is the basic transmission and reception modules running inside the AP. Once a request for mobile vehicular cloud formation takes place, the AP sends multicast messages to a list of volunteer-vehicles who are registered with the AP. These end-points act as the primary points of service provisioning. This is also the first step in slice formation.
- Distribution Control: The distribution control is essentially a queuing model. The queue length is prepared based on a Jockeying threshold. A Jockeying threshold Policy [29] modifies the native queue, such that, a request is offloaded to a shorter queue which enables maintaining low latency and continuity of request arrivals in highly fluctuating channel conditions. The primary reason for such a maintenance scheme is to assure short service times which comes at the cost of discarding certain non-vehicular cloud packets. This holds the collection of queues from multiple interfaces and ensures the slice resource data-base is updated. The entire flow of requests from the on-board vehicular NIC to the Distribution Control and back is what we define as $data - flow$.
- Slice Resource DB: This is the data-base of all the volunteers who are willing to collaborate with one another. Forming a slice and the network resources dedicated to a slice in terms of transmission period and bandwidth is stored here. At the time of initiation of a resource composition or a **service slice**, the AP decides to keep reward points and penalization points for those who complete the requests within a specific period of time after the request is granted.

4.2 Congestion Manager

The Congestion Manager module performs the final threshold allocation to the queues before dispatching the services. It is important to do that due to variations in channel and aforementioned inconsistencies at the time of queuing of sub-flows. The AP must know the scaling of the congestion window for both WiFi and cellular links. We consider a ratio that is modeled based on prior work in [30] following the congestion window update with a multiplicative decrease (for packet loss in the path update $cwindow/2$) and additive increase (for each acknowledgment received in the update $cwindow + \Delta cwindow$).

The update of multi-path links are slower as has been stated in [8]. Intuitively, slower links means longer wait times. Thus, by following a *jockeying policy* that is installed at the controller application; the access point sniffs the local area traffic for congestion (e.g. link shows high fluctuations resulting in transmission period overlapping) and all the flows that have the tendency to remain longer than the jockeying threshold are either forcefully made to re-transmit with a new time out interval or move to a faster interface queue. Consider there are n waiting queues that needs to be serviced by c number service-slice. Now, as the general case, we assume the requests always tend to join the shortest queue and if the service is not obtained within a specific threshold called $Jockey - Threshold$ represented as j_t, the request will be moved to another shorter queue. Consider two queues i and j where length of queues $\kappa_i \geq \kappa_j$. At j_t, we have rate of leaving queue i as

$$rate = k(win_i - win_j) \tag{1}$$

where win_i and win_j are waiting times for requests in the queues. win_i is $\kappa_i - 1$ if $\kappa_i \geq 1$ composition, else 0.

That is, if the service rate at the AP with two servers is $\mu_1 \leq \mu_2$ with each arrival competing for being serviced, the total buffer is B with arrivals coming inside the same as $\lambda_1, \lambda_2..\lambda_n$. In such a case, the system reaches maximum throughput at a time t_{maxt} irrespective of the arrival. At any time, however, $\lambda \geq \mu$ would result in requests being dropped. Let's assume there are two flows coming in with arrivals λ_1 and λ_2 serviced at μ. Then, the probability of an arriving packet from flow 1 competing with flow 2 is

$$P = \frac{\lambda_1}{\lambda_1 + \lambda_2} \tag{2}$$

Our goal is always to maintain a consistent throughput for both flows, which means it could come as a trade-off when one stream that is equal to zero is maintained at a value greater than 0. For example, as the number of arrivals from flow 1 becomes significantly larger than flow 2, the system normally would serve only flow 1, however, we try to maintain the levels of both queues belonging to two different interfaces in order to prevent buffer-bloat at the queues. For further examination of the jockeying procedures, Reference [29] would be helpful. We maximize the successful request completion as follows

$$max\{x_i P_i\} \tag{3}$$

where each element that is queued and leading to a successfully serviced request is represented as P_i units and x_i is the channel fluctuation dependent decision variable. x_i reprecents a low fluction region and is equale to 1 if $m \geq 1$ else 0.

The term m is the fading parameter of the Nakagami-m model. Owing to it's empirical measurements and it ability of encompassing a large variety of channels, we have, the probability density function given by [31] as,

$$p_\gamma(\gamma) = \frac{m^m \gamma^{m-1} \exp(-m\gamma/\gamma')}{\gamma'^m \Gamma(m)} \tag{4}$$

Here, the signal to noise ratio is the γ value used for statistical distribution. The m value is the fading parameter. As the value of x becomes close to zero, it is clear that the requests are in a deep fade region that would result in heavy packet drop. However, our intention is to manage the queues such that the transmission time dedicated is not wasted. In essence, minimizing congestion and maximizing request acceptance is our primary goal.

To this end, we model a ratio called the Selection Index, defined as, total requests serviced (P_i) over the requests accepted (P_t)

$$Selection - Index = P_i / P_t \tag{5}$$

Proof. Using the exchange principle, we can take S_t serviced units s.t t paths selected with capacity B over channel parameter y_t, resulting in a new cost as follows:

$$\sum_{t \in n} y_t P_t \tag{6}$$

The exchange principle works such that we replace some portions of congested links i with known back-up links j, such that the channel fluctuation parameters are $y_1, y_2..y_i..y_j.., y_n$. Essentially, we can consider a residual δ and δ‘ that could replace a portion of i with j. That is, consider i to be where congestion is present due to collision in transmission times of two slices. In such a case, a reserved portion from j is is added. This eventually results as $y_i - \delta$ and $y_j + \delta'$. With respect to the total capacity B, equation 9 becomes

$$S_1 y_1 + S_2 y_2 + ..S_i(y_i - \delta) + .. + S_j(y_j + \delta') + S_n y_n = B \tag{7}$$

$$\sum_{t \in n} s_t y_t + (s_j \delta' - s_i \delta) = B \tag{8}$$

From the above equation we know that if $(s_j \delta' = s_i \delta)$ then it would result in maximum allocated capacity of B. This is after congested link exchanges.

Since $P_j \delta' \geq P_i \delta$ and $P_j / P_i \geq \delta / \delta'$, and since $\delta / \delta' = S_j / S_i$, thus,

$$P_j / S_j \geq P_i / S_i \tag{9}$$

This shows the increase in successful request completion based on the ratio of Requests Completed Successfully to link fluctuations. We refer to this as the AP Selection Index. □

4.3 Mobility Tracker

Mobility Tracker is the agent that handles transient disconnections and maintains on-going sessions. The module examines the current location of the resource which has become part of the composition and updates the location in the database. This module works cohesively with the Slice Resource DB because once a service slice is formed, the tracking agent knows the current location of the provider. Hence, this provider's support of overall serving capability and the overall QoS of the entire serving slice depends on how good the individual unit of the vehicular cloud behaves. For a given session ID and a sequence number, our pre-defined preferences chose certain links based on the updates of the Mobility Tracker. Issues other than those pertaining to the composition, such as, different kinds of network triggered handovers and user triggered handovers are of importance but are out of scope of this work.

4.4 On-board NIC

This module runs at the provider/consumer side. Multipath TCP connections are established based on the four-way handshake i.e. after the traditional three-way handshake and addition of tokens, the on-board NIC makes an MP_Join call to add sub-flows. Owing to the heterogeneity of links, transport layer channels built over these links needs to be monitored very closely. In order to distribute data among multiple transport connections, achieving a balanced workload among these connections is not a trivial task. Additionally, when moving from one access network to the other, there has to be seamless movements, which means the session is maintained despite the movement and change of link. All of these movements need to be monitored by the application. The Mobility Agent performs this flow check periodically. We consider a stadium environment where the movements are very much limited to the stadium space. Furthermore, it does not happen often that a game

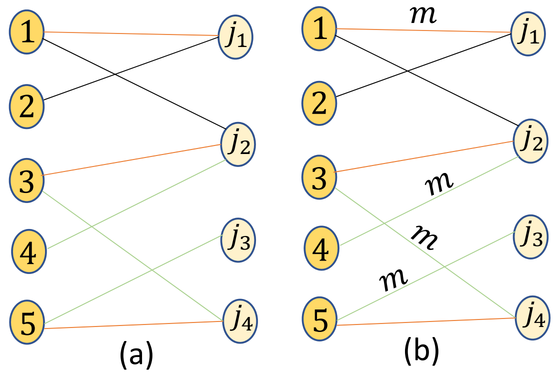


Figure 3: (a) Bipartite maximum matching. (b) Maximizing matching between jobs and resource slices

has started and an attendee is missing the game for something else. Hence, for this work we assume that the movement from one place to another is very minimal, such that the serving slice is always under the purview of an AP inside the stadium.

5 RESOURCE ALLOCATION

In this section, we show how tasks are embedded onto resources for execution. Figures 3 elaborate the process diagrammatically. Consider a set of parallely executable tasks that need service. Our goal here is to find at least one slice that can provide satisfactory request embedding. To this end, we adapt Berge's Algorithm for Bipartite Matching [25].

Algorithm 1: Adaption of Bin Packing Berge's Algorithm for Bipartite Matching.

```
Input  :Bipartite graph $G = (U, V, E)$, tasks $T_n$
/* Cardinality of U and V is given by $n_1, n_2$; E defines Edge Set
   */
MappingOfTask()
begin
    foreach $t$ in $T_n$ do
        Search an Edge in $G$ which Maximises bipartite
          matching
        map $t$ to the respective edge.
    end
    Return $\{n_1, n_2, E$, Satisfied $t\}$
end
Output:$T$ mapped to G
```

6 EVALUATION AND DISCUSSION

The evaluation is divided in to two sub-sections:

1. Describing the mechanics of Jokeying policy and examine the overheads.
2. the allocation algorithm that enables maximum matching between the jobs to be executed and the candidate resources.

Cases considered for the evaluation are:

Table 1: Scenarios for evaluation within Mininet

Scenario No.	Criteria
1	Mobility with MPTCP
2	Event-driven with MPTCP
3	Mobility without MPTCP
4	Event-without MPTCP

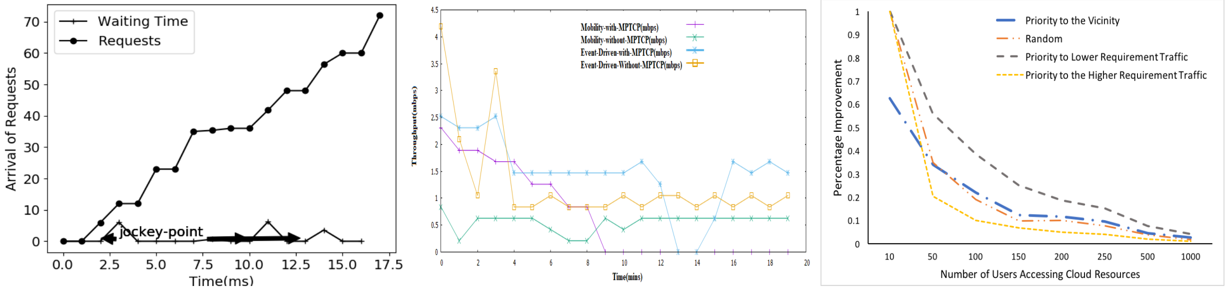
1. Vehicles close to the Access Points are considered as the priority users for slice assignments. For example, slices are formed by the vehicles at one-hop from the user/consumer.
2. A random assignment is achieved, based on a first in first out request selection by the AP.
3. The resource assignment is achieved based on ascending order of requests. For instance, a 3GB request is serviced before a 4GB request by the AP .
4. The resource requests and their assignments are achieved based on the descending order of requests (The exact opposite case of scenario 3).

Both the scenarios are used to perform a sensitivity analysis of the framework and the algorithm separately hence are divided into two sets.

6.1 Experiment and Setup

We utilize the Mininet simulator only for the network analysis. We follow an experimental design approach for collecting information about a system. As we do not have any industry grade access points at our disposal, we provide simulation tests of different scenarios as observed in Table 1. Our endeavor is to have a preliminary proof of concept model, so, we consider a mobility scenario, where the network environment is affected by factors like latency between client-server and fluctuations in the link that affects the size of the buffer. In order to simulate, we consider only mobile devices such as laptops operating over the MPTCP kernel as vehicular-MPTCP kernel implementations are not available. For all experimental design techniques, a domain selection is necessary. To this end, we evaluate constantly varying mobility environments. We synthetically model an event by randomly switching a controller on or off based on the movements. Further, to emulate a changing channel we follow the $Nakagami - m$ model [31] that encompasses all classes of fading. Afterwards, we let the controller run as much as the coverage region for individual access points. The event-driven approach is the manual intervention of a designer, that is we manually change the location parameters to inflict a random change in the environment which in turn affects the bandwidth.

The considered topology consists of 5 mobile vehicles communicating over the MPTCP kernel v0.94. These sub-flows undergo different path features. As Mininet provides an emulated environment and offers flexibility in setting-up of the test-bed. Then, we create queuing modules on Python to evaluate the jockey-policy. Different scenarios (See Table 1) have been considered for networking analysis. For all conducted simulations, a 4GB RAM on a Linux operating system has been used. For resource allocation procedures, we use a MacOS and create slices locally for our evaluation.



(a) Jockeying policy evaluation **(b) Bandwidth evaluation** **(c) Improvement based on task acceptance**

Figure 4: Performance Evaluation

Table 2: Defining Value of X used for the Simulation shown in Fig. 4c and 5

Serial	Resource	Magnitude
1	RAM	4.5 GB
2	Storage	41.5 GB
3	CPU	4.5 Cores

As shown in Figure 4a, the points of jockeying are visible where the waiting times are not affected, however, overheads due to packets moving between queues is observed. Because the time-stamped *jockeying* requests produce an extra over-head of moving from one queue to another, it is worthwhile to investigate in the future if this affects the entire Vehicular cloud operation.

6.2 Discussion

We did not find an implementation close to our work for making a comparison. Therefore, we decided to have scenarios where in the proposed architecture is used and compare it with an environment where the architecture is absent. We generate bulk transfers using iperf with each session lasting for 100 seconds to examine bandwidth and shared transmission periods. The mobility scenarios behaved as expected with changing node movements starting from the AP all the way to locations away from the AP. Our intention is to share the AP transmission period and bandwidth between slices. As shown in Figure 4b, an improvement has been observed with our algorithm running in the controller.

On running the application on Mininet, several ambiguities were observed. Firstly, due to lack of queue discipline customization, the accuracy of the bandwidth utilization needs to be further examined. Secondly, the multi-path kernel needs to always be called for mobility events which would be sub-optimal in some cases. Event-driven scenarios are more controlled. That is, the controller running above the topology knows of the event already which prepares a pipe with the next available link.

It is very clear also from the jockey-policy plots that benefits in terms of request service time is obtained, the waiting time for other requests are not affected. That is, we assume our system to already have n requests, therefore, the $(n+1)^{th}$ request will have the option of jockeying to another queue with the policy enabled $jockey - time - stamp$. This time-stamp is a backward counter that goes to the next available buffer and waits till the counter clocks to -1. This is also the jockey-threshold. As stated previously, jockeying creates system overheads while hopping between queues thereby stalling the newer requests, which needs to be further investigated.

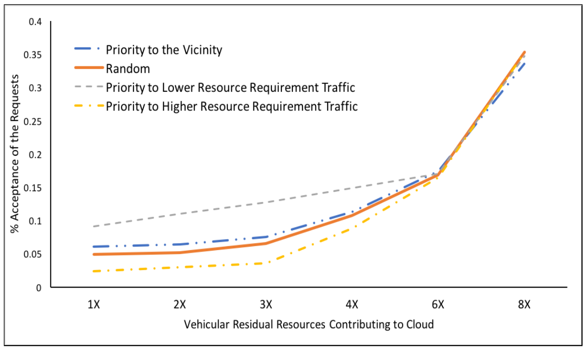


Figure 5: Task to slice Mapping

6.2.1 Resource Allocation Evaluation. For this evaluation, we simulate an environment with twenty users and generate uniformly distributed AP IDs. Gradually, we increase the number of users, following the same procedure for the AP ids. To this end, our approach relies more on setting aggregated resources per slice that enables mapping incoming jobs to resource slices. As shown in Figure 4c, slices are created based on the resource requests consisting of CPU, RAM and Storage requirements as a meta-file. We model the RAM, CPU and Storage requirements based on current trends in the mobile phone growth internal resource distribution. On increasing the density of users, a sensitivity analysis has been conducted. As expected, a saturation is seen with the number of users accessing a specific set of cloud resources, increases up to 1000. It is also observed that the results approximately merge. The priority for high resource requirement is called $Higher\ Priority\ traffic\ requirement$ likewise for low resource requirement. We observe a classical resource management problem when some of the resources are not able to satisfy any requests. In case of random requests, such a problem occurs often.

Figure 5 shows the behavior of the environment when vehicles contribute to form mobile clouds. Cloud resource is kept constant to prepare a static-allocation based resource assignment pool. We observe in all scenarios an average up-to 60% improvements with our architecture owing to the knowledge of the slice resource DB with constant resources. The value X represents the vehicular computing resources used. Assuming we have one vehicle which is capable of sharing one storage unit of resource to form a cloud, then X=1. Likewise, we follow a homogeneous vehicular cloud distribution technique to maintain simplicity for this evaluation. The average resource available for sharing by the vehicles forming cloud is shown in table 2. The proposed solution will be evaluated in a scenario which reproduces a crowded scenario, including regular traffic besides the one created by the volunteer cloud itself in the future.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed and implemented an experimental framework of the Vehicular Cloud with vehicles together forming a cloud resource unit. This framework was used to examine a network slice that is managed at the WiFi access points in densely crowded environments where attendees volunteer their vehicle resources for the formation of a vehicular cloud. The experimental design approach allowed quantifying of the factors that affect the performance of the network in times of coupling between multiple paths. We have also proposed a methodology to share bandwidth and transmission period of an AP dedicated to different slices. We developed a jockeying framework at the AP that allows time-sensitive flows to be moved to a faster service queue. Further, we proposed a maximum matching scheme for job to candidate slices that enables resource to job matching. In doing so, we see benefits achieved in terms of bandwidth. Over 60% improvement is observed with the architecture. In the future, an implementation model will be investigated and results will be shown.

REFERENCES

[1] Super-bowl stats. fiercewireless//com/wireless/super-bowl-traffic-stats-verizon-leads-at-4-1-tb-t-mobile-trails-430-gb.

[2] Utsav Drolia, Nathan Mickulicz, Rajeev Gandhi, and Priya Narasimhan. Krowd: A key-value store for crowded venues. In *Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture*, MobiArch '15, pages 20–25, New York, NY, USA, 2015. ACM.

[3] V. Balasubramanian and A. Karmouch. An infrastructure as a service for mobile ad-hoc cloud. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–7, Jan 2017.

[4] Azzedine Boukerche and Robson E. De Grande. Vehicular cloud computing: Architectures, applications, and mobility. *Computer Networks*, 135:171 – 189, 2018.

[5] Zhengguo Sheng, Xiping Hu, Peyman TalebiFard, Victor C.M. Leung, Ruifeng Chen, and Yingjie Zhou. Sensor cloud computing for vehicular applications: From analysis to practical implementation. In *Proceedings of the Fourth ACM International Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '14, pages 53–59, New York, NY, USA, 2014. ACM.

[6] Ashwin Ashok, Peter Steenkiste, and Fan Bai. Vehicular cloud computing through dynamic computation offloading. *Computer Communications*, 120:125 – 137, 2018.

[7] Felipe Leite Lobo and et. al. Solve: A localization system framework for vanets using the cloud and fog computing. In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '17, pages 17–22, New York, NY, USA, 2017. ACM.

[8] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. Exploring mobile/wifi handover with multipath tcp. In *CellNet '12*, 2012.

[9] Network slicing for 5g: Challenges and opportunities. techblog.comsoc.org/2018/05/18/ieee-comsoc-papers-on-network-slicing-and-5g/.

[10] M. Richart, J. Baliosian, J. Serrati, J. L. Gorricho, R. AgÃijero, and N. Agoulmine. Resource allocation for network slicing in wifi access points. In *2017 13th International Conference on Network and Service Management (CNSM)*, pages 1–4, Nov 2017.

[11] C. L. I, S. Han, Z. Xu, S. Wang, Q. Sun, and Y. Chen. New paradigm of 5g wireless internet. *IEEE Journal on Selected Areas in Communications*, 34(3):474–482, March 2016.

[12] R. Wen, J. Tang, T. Q. S. Quek, G. Feng, G. Wang, and W. Tan. Robust network slicing in software-defined 5g networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, Dec 2017.

[13] Moayad Aloqaily, Burak Kantarci, and Hussein T Mouftah. Vehicular clouds: State of the art, challenges and future directions. In *Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on*, pages 1–6. IEEE, 2015.

[14] Moayad Aloqaily, Burak Kantarci, and Hussein T Mouftah. Fairness-aware game theoretic approach for service management in vehicular clouds. In *Vehicular Technology Conference (VTC-Fall), 2017 IEEE 86th*, pages 1–5. IEEE, 2017.

[15] Abdul Jabbar Siddiqui, Abdelhamid Mammeri, and Azzedine Boukerche. Towards efficient vehicle classification in intelligent transportation systems. In *Proceedings of the 5th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, DIVANet 2015, Cancun, Mexico, November 2-6, 2015*, pages 19–25, 2015.

[16] Fabrício A. Silva and et. al. Content replication and delivery in vehicular networks. In *Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications, DIVANet@MSWiM 2014, Montreal, QC, Canada, September 21-26, 2014*, pages 127–132, 2014.

[17] Xiping Hu and et. al. Towards a service centric contextualized vehicular cloud. In *Proceedings of the Fourth ACM International Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '14, pages 73–80, New York, NY, USA, 2014. ACM.

[18] Stephan Olariu and Ryan Florin. Vehicular clouds research: What is missing? In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '17, pages 77–84, New York, NY, USA, 2017. ACM.

[19] Ismaeel Al Ridhawi, Moayad Aloqaily, Yehia Kotb, Yousif Al Ridhawi, and Yaser Jararweh. A collaborative mobile edge computing and user solution for service composition in 5g systems. *Transactions on Emerging Telecommunications Technologies*, 2018.

[20] Moayad Aloqaily, Burak Kantarci, and Hussein T Mouftah. Multiagent/multiobjective interaction game system for service provisioning in vehicular cloud. *IEEE Access*, 4:3153–3168, 2016.

[21] Moayad Aloqaily, Burak Kantarci, and Hussein T Mouftah. On the impact of quality of experience (qoe) in a vehicular cloud with various providers. In *High-capacity Optical Networks and Emerging/Enabling Technologies (HONET), 2014 11th Annual*, pages 94–98. IEEE, 2014.

[22] Ismaeel Al Ridhawi, Moayad Aloqaily, Burak Kantarci, Yaser Jararweh, and Hussein T. Mouftah. A continuous diversified vehicular cloud service availability framework for smart cities. *Computer Networks. https://doi.org/10.1016/j.comnet.2018.08.023*, 2018.

[23] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker. Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks. apr 2017.

[24] B. Venkatraman, F. A. Zaman, and A. Karmouch. Optimization of device selection in a mobile ad-hoc cloud based on composition score. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 257–262, April 2017.

[25] Claude berge. pnas.org/content/pnas/43/9/842.full.pdf.

[26] Moayad Aloqaily, Ismaeel Al Ridhawi, Burak Kantraci, and Hussein T Mouftah. Vehicle as a resource for continuous service availability in smart cities. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*, pages 1–6. IEEE, 2017.

[27] Jorge Mena, Peter Bankole, and Mario Gerla. Multipath tcp on a vanet: A performance study. *SIGMETRICS Perform. Eval. Rev.*, 45(1):39–40, June 2017.

[28] Cengis Hasan and et. al. On lte-wifi coexistence and inter-operator spectrum sharing in unlicensed bands: Altruism, cooperation and fairness. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '16, pages 111–120, New York, NY, USA, 2016. ACM.

[29] Ernest Koenigsberg. On jockeying in queues. *Management Science*, 12(5):412–436, 1966.

[30] Matei Popovici and Costin Raiciu. Exploiting multipath congestion control for fun and profit. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, HotNets '16, pages 141–147, New York, NY, USA, 2016. ACM.

[31] Gordon L. Stüber. *Principles of Mobile Communication (2Nd Ed.)*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.