Proactive Caching at Parked Vehicles for Social Networking

Sara A. Elsayed*, Sherin Abdelhamid*†, Hossam S. Hassanein*
*School of Computing, Queen's University, Kingston, ON, Canada
{selsayed, sherin, hossam}@cs.queensu.ca
†Faculty of Computer Science and Information, Ain Shams University, Cairo, Egypt
shereen@cis.asu.edu.eg

***Abstract*—The majority of Internet users are active users of social networks and thus social media traffic represents the highest percentage of Internet traffic. The average daily usage of social media has been rigorously growing. Mobile devices are considered to be the most common facilitators of such usage. The aforementioned facts contribute to the excessive traffic load on the Internet, poor users experience in terms of quality of service, and high cellular costs spent by mobile users. In this paper, we strive to reduce these effects by proposing a scheme called Proactive Caching at Parked Vehicles (PCPV). PCPV aims to provide a better quality of service to vehicular users who tend to have a somewhat consistent social networking behavior. In particular, users who have a predictive behavior in terms of the type and time of access of social media platforms as a part of their daily routine during transit from one place to another. This is done by having the required data pre-cached and ready for users to proactively acquire at roadside parked vehicles as they pass by, rather than sending a request to the far-away data center and waiting for the reply. Performance evaluation of PCPV shows significant improvements in terms of delay, packet delivery ratio, and cache hit ratio compared to the reactive approach typically used in infotainment applications.**

***Index Terms*—Parked Vehicles, Caching, and Social Media.**

I. Introduction

Vehicular Ad Hoc Networks (VANETs) have emerged as a communication paradigm that strives to facilitate interconnection and communication between vehicles on the road. They have been capturing increasing attention in the research community due to their primary role as an enabling technology for Intelligent Transportation Systems (ITS) [1]. ITS aims to streamline a wide range of applications, including infotainment applications (e.g. Information services such as Internet access and location-based services).

In VANETs, vehicles can exchange information with each other (Vehicle-to-Vehicle communication-V2V), and with roadside network infrastructures (Vehicle-to-Roadside communication-V2R) [1]. Road Side Units (RSUs) are deployed along the road and are usually interconnected via the Internet [1]. However, the cost of wide RSUs deployment is rather expensive [1]. On the other hand, roadside parked vehicles are natural infrastructures that have been shown to exist in considerable numbers [2]. A study that monitored on-street parking in Ann Arbor city in the US shows that the occupancy ratio of parking spaces can reach 93% and 80% during on and off-peaks, respectively [2].

In order to support VANET-based applications, it is important to address the challenging issues associated with data access in such networks. Data access in VANETs can either be push-based (proactive) or pull-based (reactive) [1]. In existing push-based schemes, data is periodically broadcast to vehicles whether they need it or not. In these schemes, data delivery depends on opportunistic encounter with vehicles that happen to have the data. Thus, they are often considered unsuitable for user-specific applications [3]. In most infotainment applications, data access is pull-based [1]. That is, a request is sent to a particular area of interest (location-based services) or access point (Internet access) and a response is sent back to the requester. This process is often associated with high delay and low packet delivery ratio due to the highly dynamic topology and intermittent connectivity of VANETs [1], [3]. By the time the data is sent back to the requester, it might have already moved and the packet is either dropped or an attempt to track its new location is made [1]. One way to improve data access in VANETs is by bringing the data closer to the requester through caching.

Caching in VANETs can also play a vital role in reducing the effect of the ever increasing use of the Internet. Sustaining the scalability of the Internet to accommodate the excessive traffic growth has become a pressing research problem recently [4]. Due to the excessive traffic load on the Internet, some users may experience high delay and poor quality of service. Some research efforts have proposed developing a new Internet architecture to resolve this problem [4]. Note that mobile devices facilitate more than 60% of the time spent on the Internet [5]. This leads to higher cellular costs spent by users, particularly in outdoor areas that have no Wi-Fi connection.

We strive to alleviate the aforementioned problems by shifting part of the social media traffic towards the vehicular network. In particular, we pre-cache such traffic at roadside parked vehicles to be proactively acquired by users as they pass by. We focus on social media as it represents the highest percentage of Internet traffic. In 2016, social media users constituted 68% of Internet users [6]. It has also been estimated that social media users will amount to 2.67 billion in 2018 [6]. In addition, there is a high correlation in the data accessed by most users on social media. This is attributed to the fact that the most followed accounts on social media platforms are those of public figures, such as politicians, actors, singers,

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

etc. For instance, the number of followers of the top 10 most followed public figures on Instagram ranges from 80 million to 120 million [7]. In this paper, we propose a proactive caching scheme, called Proactive Caching at Parked Vehicles (PCPV), to provide a better quality of service to social media users.

PCPV is based on the fact that some users tend to establish a daily routine, whether in the route they follow, the time of taking that route, or the type and time of requesting certain data. For instance, as a vehicle is slowly moving during a typical morning rush-hour on the user's daily route to work, he/she is used to dropping off their teenage children at school first, who in turn are used to accessing their Instagram accounts to check the latest posts of public figures. Such a daily routine provides a predictable behavior that PCPV exploits to have the required data pre-cached at roadside parked vehicles and ready for users to proactively acquire as they pass by.

In PCPV, the data center is the centralized entity that makes all vital decisions. It has three fundamental tasks. First, it predicts all the information required to make informative caching decisions. Second, it selects the appropriate road segments for caching. This decision is made by tackling the time-sensitive constraints using a greedy heuristic approach. Once a road segment is selected, the data center randomly chooses one of the parked vehicles available there to cache the replica. Third, the data center determines when to send the replicas to the selected parked vehicles to be cached at the appropriate time. In addition to satisfying proactive requesters, PCPV enables reactive requesters to benefit from the cached copies at parked vehicles as well. Moreover, moving vehicles engaged in V2V communication keeps a copy of the data in their cache to satisfy reactive requests.

To the best of our knowledge, PCPV is the first proactive caching scheme that pre-caches the data at parked vehicles for the users to acquire either before the time of their requests, or within the deadline specified. Hence, PCPV strives to achieve a certain quality of service demanded by users. This is done using a heuristic greedy approach to pre-cache the data at the proper time and place based on future requests, trajectories, and estimated period of encounter with road segments. PCPV is a cost-effective solution that relies on natural infrastructures.

The performance of the proposed PCPV scheme is evaluated using the NS-3 simulator and is compared to the reactive approach. Simulation results show that PCPV outperforms the reactive scheme in terms of access delay, packet delivery ratio, and cache-hit ratio, while maintaining proper satisfaction ratio.

The rest of the paper is organized as follows. Section II provides a brief overview of some related work. Section III presents a detailed discussion of the proposed scheme (PCPV). Section IV presents the performance evaluation and simulation results of the scheme. Section V concludes the paper and presents our future work.

II. Related Work

A. Data Access in VANETs

Many schemes have been proposed to resolve the problems associated with pull-based data access in VANETs [1]. Some schemes focus on tracking the location of requesting vehicles to successfully deliver the data back to them. Several approaches involve the use of a location service management protocol [1], [8], [9]. In [10], the signaling overhead associated with such protocols is alleviated using static nodes deployed at intersections to keep track of the heading of the requesting vehicles. On the other hand, few schemes in VANETs have been proposed to bring the data closer to the requester through caching. In [11], intermediate nodes along the data forwarding path cache the data. In [10], the static nodes deployed at intersections are used for caching.

Push-based schemes can also be used for data access in VANETs. An example of push-based schemes is proposed in [12], where the data provider periodically broadcasts the data to all the vehicles within its transmission range to be cached. Each vehicle can then exchange the cached data with other vehicles upon encounter. Thus, users can acquire the data they are interested in only if they encounter another vehicle that happens to have it. To increase data availability, in [3], the data provider periodically broadcasts the data to all the vehicles in highly dense roads and at intersections. These schemes eliminate the need for an infrastructure, which makes them cost-effective and suitable for highly dynamic vehicular environments. However, severe congestions can occur due to excessive transmissions, particularly in dense networks [3].

B. Content Downloading Using Parked Vehicles in VANETs

Recently, some schemes, such as [13]-[15], have proposed the use of roadside parked vehicles for downloading large contents. Such contents cannot always be obtained all at once because of the short connectivity period between vehicles and infrastructures [13]. In [13], a requesting vehicle sends a content download request to the nearest parking cluster. The parked vehicle located at the entry of the road segment is selected as the cluster head (CH). If the requested content is too large, the CH communicates with other parking clusters to download the remaining parts and have them ready for the requester. The authors assume that parked vehicles are connected to the content provider via the Internet. This incurs additional costs for parked vehicles. In [14], each moving vehicle sequentially uploads its content chunks as it passes by parked vehicles that form line clusters. This is done without considering the possible trajectories or needs of the requesters. When a request is received, the CH informs the requester of the IDs and locations of the parked vehicles that have the requested chunks to communicate with them. In this scheme, acquiring the data relies on opportunistic encounter of the requesting vehicles with line clusters that happen to have the data. In [15], upon receiving a request, the Internet connectivity between RSUs enables them to fetch the data from nearby parked vehicles rather than the far-away data provider. However, RSUs require very large investments [14].

III. Proactive Caching At Parked Vehicles (PCPV)

In this section, we present the assumptions, system model, and the functionality of the PCPV scheme at the data center, parked vehicles and moving vehicles.

A. Assumptions

PCPV is applied in urban environments. We assume that proactive requesters provide the data center with their trajectories, as well as access to their profiles on social platforms. In particular, platforms that have no large-sized contents, such as Instagram. This is in exchange for the service and based on a privacy agreement. When subscribing to the service, each user specifies his/her maximum acceptable delay. Accordingly, the data center knows the deadline of each user's request. The data center has a prediction module that enables it to predict three pieces of information based on historical data. First, it predicts the time of request of each user. To estimate such information of a subsequent request, predictors of requests' patterns such as [16] can be used. Second, it estimates the posting frequency of a given public figure. This information indicates the Time to Live (TTL) of the data. TTL is the expected time before a public figure posts a new post, marking the earlier data as old. Third, it estimates the period of encounter of the requesting vehicle with each road segment along its trajectory. Estimators such as [17] can be used to predict such information.

Parked vehicles that are willing to participate in the caching service, report their typical parking time and road segment, as well as the duration of their availability to the data center. They are then solicited by the latter in exchange for some incentives. The parked vehicles dedicate different sizes of their entire cache capacity for the caching service. This poses some restrictions on the available cache capacity. In order to reduce the traffic load imposed upon the data center, the number of replicas needs to be reduced while maximizing cache hits.

B. System Model

Let U be the set of proactive users. Every user $u \in U$ can be interested in a particular data $d \in D$, where D is the set of the latest data items posted by public figures that the users follow. Each data item $d \in D$ is associated with a Time-To-Live TTL_d. Every user $u \in U$ has a trajectory, T_u. Let $T_u = r_1$, r_2, $..r_k$ denotes a vehicle's trajectory sequence, where r_j depicts the vehicle's encountered road segment. Let R be the set of all road segments, with cardinality n.

Each road segment along the trajectory of a user u, $r_j \in T_u$, is associated with a time period, ϵ_j^u, specifying the start time, t_{jstart}^u, and the end time, t_{jend}^u during which user u is expected to encounter r_j. Each period is associated with a probability of encounter, δ_j^u, specifying the degree of accuracy of that period. The data center estimates a number, L, of possible periods of encounter of each user with each road segment along its trajectory such that $\sum_{y=1}^{L} \delta_{jy}^u = 1$. The request time of each user $u \in U$ for data d is τ_d^u and the deadline is η_d^u. The available cache capacity at a road segment r_j is C_j.

C. PCPV at the Data Center

A detailed discussion of the main tasks performed by the data center is provided in the following sections.

*1) **Selection of Road Segments for Caching**:* In order to reduce the number of replicas, users are grouped together based on the similarity of their requested data, and overlapping road segments. For instance, consider that both users u_1 and u_2 pass by r_4 along their trajectory. Thus, if they are interested in the same data, it is better to cache one replica at r_4 to serve both users. However, the time constraint is one of the most fundamental issues that needs to be taken into consideration before making such a decision. The example depicted in Figure 1 illustrates some of these time constraint restrictions. Another vital issue in road segments selection is the probability of encounter, which has a direct effect on the probability of cache hits since it determines the degree of accuracy of the estimated period. Thus, if r_4 passes all the time constraints, but the probability of encounter of u_1 by r_4 is relatively small, then a different road segment may be a better alternative for u_1. The selection procedure is composed of the following three stages:

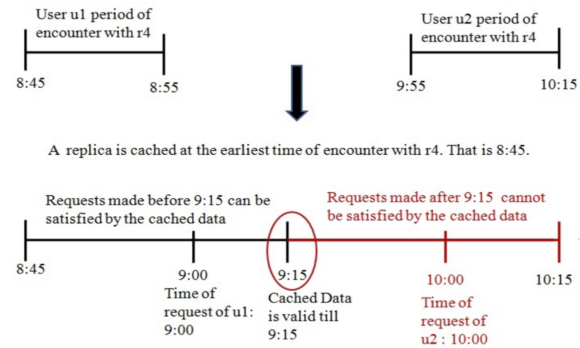


Fig. 1. Time Constraint Restrictions in the Caching Decision. If a replica is cached at 9:55, u_1 would pass by r_4 before the data is available. Thus, a replica should be cached at 8:45. A list of all users that should receive the replica is included in its header. The TTL of the data is 30 minutes, so it will be valid till 9:15 (8:45+TTL). As u_1 passes by r_4, the caching parked vehicle senses its presence and sends it the replica. User u_1 caches the data in its local cache. At 9:00 (u_1's time of request), it will find it available and still valid. However, for u_2, the data will no longer be valid at 10:00.

a) Grouping Users based on Data Similarity: The data center groups users that request the same data items together. Let G be the set of groups generated by the data center. To determine the sequential order for processing the groups, every group $g \in G$ is ranked based on the popularity of the data requested by its users. The popularity of the data is determined by the number of users within the group whose time of requests are within the TTL of the data. Once the groups are ranked, the following stages (b and c) are performed for each group.

b) Determining Feasible Road Segments for Caching: Some time-constraint checkpoints are tested to determine the road segments that are eligible for caching. For each user u in group g, and for each each road segment r_j along its trajectory, the period that has the maximum probability of encounter, P_j^u=$\max_{y=1}^{L} \delta_{jy}^u$, is selected. A feasibility matrix, denoted $F_{m\times n}^g$ is then created. A feasibility matrix indicates the road segments that are eligible for caching for each user within the group, where $m = |g|$. The feasibility matrix is filled based on Eqn. 1. The first condition in Eqn. 1 checks whether the road segment r_j is part of the user's trajectory. The second condition has to do with ensuring the validity of the data by the time the user requests it. The start time of the period of encounter, t_{jstart}^i, is the earliest time for caching to occur.

Thus, the expiry time of the data is obtained by adding the TTL to that value. The third condition in Eqn. 1 has to do with the deadline of the request. For instance, if the deadline of the request of u_1 is 9:20 and the time of encounter of u_1 with r_9 is 9:25, then by the time u_1 passes by r_9, the data will not be received at a satisfactory time. The fourth condition checks whether there is available cache space at the road segment.

$$f_{ij} = \begin{cases} 1 & r_j \in T_i, and \\ & t^i_{jstart} + TTL^g_d \geq \tau^i_d, and \\ & t^i_{jstart} \leq \eta^i_d, and \\ & C_j \geq 1 \\ 0 & Otherwise \end{cases} \quad (1)$$

c) Selecting Replicas Locations based on Overlapping Road Segments: The goal is to group as many users as possible such that the degree of cache hits is maximized and the number of replicas is reduced. For this purpose, an iterative procedure is executed as detailed in Algorithm 1. A matrix called the selection matrix, denoted $S^g_{m \times n}$ is created and iteratively updated to select the caching road segments for all users within group $g \in G$. The selection matrix is filled based on Eqn. 2. Note that for users to be grouped, it is imperative to ensure that the TTL condition checking the validity of the data is maintained, and that there is enough cache space. Let $U^{'}$ be the set of users in $g \in G$ who have not been assigned a replica location yet and $G^{'}$ be the set of users in $g \in G$ who have already been assigned a replica location. Initially, the selection matrix is set to the feasibility matrix, $U^{'}$ is set to $g \in G$, $G^{'}$ is empty, and the list of assigned replicas is empty (lines 10-14).

$$s_{ij} = \begin{cases} P^i_j & t^p_{jstart} + TTL^g_d \geq \tau^i_d, and \\ & C_j \geq 1 \\ 0 & Otherwise \end{cases} \quad (2)$$

As long as $U^{'}$ is not empty, the following steps are iteratively repeated. First, for each road segment $r_j \in R$, a primary user, p, is selected from the users in $U^{'}$ that consider r_j eligible for caching (lines 16 & 17). The primary user is the user that has the earliest time of encounter with r_j. Second, the selection matrix is updated by Eqn. 2 (lines 18-20). The selection matrix values reflect the maximum probability of encounter of each user u_i with each road segment r_j, denoted P^i_j. Third, the degree of cache hits over each road segment, denoted DCH_{r_j} is calculated as the sum of the selection matrix values for all users in $U^{'}$ over road segment r_j (lines 21 & 22). The segment selected for caching, denoted $r_{selected}$, is the one rendering the maximum degree of cache hits (lines 23 & 24). The cache capacity of $r_{selected}$ is reduced by one if the caching duration of the replica overlaps with that of a previously assigned replica for a user in a different group in G (line 25-27). Fourth, the selected road segment is the replica location for the users who consider it an eligible segment for caching. Thus, the set of users in $U^{'}$ who have non-zero entries in the selection matrix over the selected road segment are grouped together in $g^{'} \in G^{'}$ and removed from the set $U^{'}$(lines 28-32). A tuple is added to the list of assigned replicas associating $g^{'} \in G^{'}$ to $r_{selected}$ and the data item to be sent, d^g (line 33). If $U^{'}$is still not empty and the cache in all road segments is full, a Least Frequently Used replacement policy is used.

Algorithm 1 : Selecting Replicas Locations

```
1: Input:
2: TTL of the Data Requested by Users in g, TTL^g_d    ∀g ∈ G
3: Set of All Road Segments R
4: The Maximum Period of Encounter P^i_j    ∀r_j ∈ T_i ∀i ∈ U
5: Current Cache Capacity at Road Segment r_j, C_j    ∀r_j ∈ R
6: Feasibility Matrix F^g_{m×n} = f_ij    n = |R|, m = |g|, ∀g ∈ G
7:
8: SELECT_ROADSEGMENTS(g)
9: Begin
10:   Selection Matrix S_{m×n} ← {s_ij=f_ij}
11:   U' ← g    //Set of users in g with no assigned replica location
12:   g' ← ∅    //Set of users in g that have the same replica location
13:   G' ← ∅    //Set of g' with an already assigned replica location
14:   LR[] ← ∅    // List of Assigned Replicas
15:   while U' isNotEmpty do
16:     for all j ∈ R do    // Columns of Matrix
17:       Select Primary User from U'    //User with min t^u_jstart
18:       for all i ∈ U' do    // Rows of Matrix
19:         if s_ij ≠ 0 then
20:           Update s_ij based on Eqn.2
21:           DCH+ = s_ij    //Degree of Cache Hits
22:           DCH_List[] ← tuple (j, DCH)
23:     highest_DCH ← the maximum value of DCH
24:     r_selected ← the road segment with highest_DCH
25:     for all j ∈ R do
26:       if j=r_selected then
27:         C_j ← C_j−1    //If t^p_jstart + TTL^g_d overlaps
28:         for all i ∈ U' do
29:           if s_ij ≠ 0 then
30:             g' ← g' ∪ i
31:     G' ← G' ∪ g'
32:     U' = U' \g'
33:     LR[] ← tuple (g', r_selected, d^g)    // List of Assigned Replicas
34:   return LR[]
35: End
```

*2) **Determining when to Send the Replicas**:* The time of caching a replica destined to road segment r to satisfy the users in group $g^{'} \in G^{'}$, denoted $T_{caching}$, is the earliest time of encounter of these users with r. In dense vehicular environments, the fastest path from a source to destination is more likely to be the shortest path. Thus, the data center applies the Dijkstra's algorithm to find the shortest path to the destined parked vehicle. The time taken for a replica to be received by the latter, $T_{duration}$ is the length of the shortest path divided by the average propagation speed. Thus, the time to send the replica is $T_{caching}$-$T_{duration}$.

D. PCPV at Parked Vehicles

When the data center sends a data packet to a parked vehicle to be cached, it includes in the packet header a list of all proactive requesters, denoted LPR, that should be satisfied by the received replica. Upon receiving the data packet, the parked vehicle caches the data and extracts the LPR from the header. Beacon messages are periodically exchanged among neighboring vehicles. Thus, when a parked vehicle receives a beacon message, it checks if the source ID of the message matches one of the IDs in LPR. If so, the parked vehicle sends the replica to the moving vehicle. In addition, beacon messages include the list of cached items maintained by neighboring vehicles (as the case in the Group Caching scheme [18]). Thus, when a parked vehicle receives a request packet from a reactive requester, it first checks if the requested data is stored in its local cache. If not, it checks if one of its neighbors has a copy. If it does, the request is sent to that neighbor. Otherwise, the request is sent to the data center using greedy forwarding. In greedy forwarding, each node forwards the packet to the closest neighbor to the destination.

E. PCPV at Moving Vehicles

Moving vehicles can receive three types of messages; either a request packet from a reactive requester, a data packet, or a beacon message. Beacon messages and request packets are handled in the same way as described in parked vehicles. If a data packet is received, the moving vehicle keeps a copy in its local cache (until the data expires) and forwards the packet to the destination using greedy forwarding, as in [11].

IV. Performance Evaluation

In this section, we evaluate the performance of PCPV compared to the reactive scheme. The performance metrics are: 1) the average access delay since a request is issued till a reply is received, 2) the packet delivery ratio, which is the ratio of successfully delivered reply packets to the total number of reply packets sent 3) the cache hit ratio, which is the ratio of the number of requests served by a caching node to the total number of requests, and 4) the satisfaction ratio, which is the ratio of proactive requests that have been served before the deadline to the total number of proactive requests.

A. Simulation Setup

Simulations are conducted using the NS-3 network simulator [19] over a 4×4 road grid topography, with 48 road segments. Realistic mobility traces are generated using the SUMO traffic simulator [20], with the maximum vehicular speed set to 40 km/h. The number of moving vehicles is set to 1000, 660 of which are requesters. We use varying penetration rates to compare PCPV to the reactive scheme. A penetration rate of 80% means that 80% of the requesting vehicles use PCPV while the remaining 20% use the reactive scheme. The percentage of proactive vehicles for which the data center estimates accurate predictions is varied to assess the performance of PCPV under erroneous predictions. Note that erroneous predictions indicate inaccurate estimations of the time of request, and the period of encounter of requesters with road segments along their trajectory. Simulations are run for a period of 2000 seconds each. The IEEE 802.11p WAVE standard is used, with the transmission range set to 150 m and the beacon interval set to 0.5 seconds.

The interest generation is uniformly distributed among 8 public figures, each of which generates 1-5 new posts (i.e. data items) every 15-30 minutes. In the simulations, 100 parked vehicles are uniformly distributed among the road segments, each with a maximum cache capacity of 25 packets. Parked vehicles dedicate 20% to 40% of their total cache capacity to PCPV. The number of road segments with parked vehicles willing to participate in the caching service is $\alpha \times k$, where k is the total number of road segments and α is a tuning factor that takes a value from (0-1). Unless otherwise specified, α is set to 1. The time of each request is set to be a random time within the requester's trip duration. The deadline specified by each proactive requester is set to take a random value ranging anywhere between the request time and the end of the trip.

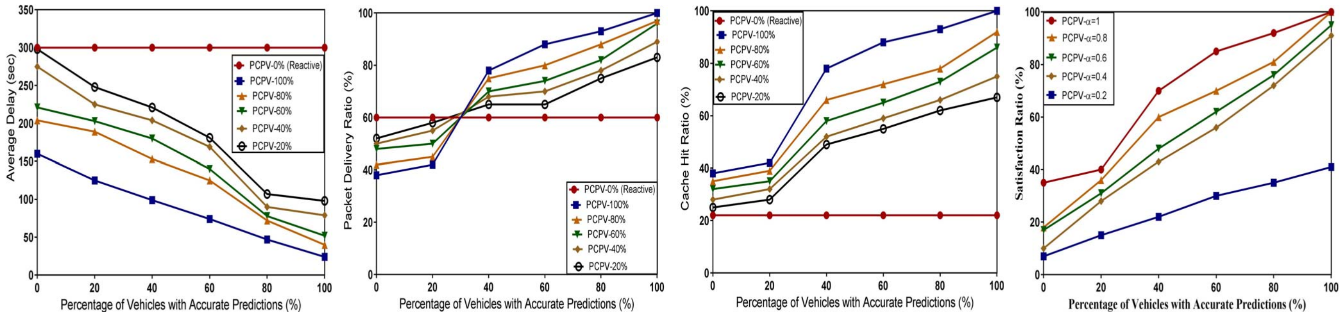
B. Simulation Results and Analysis

First, we compare PCPV and the reactive scheme in terms of the average delay over varying percentages of vehicles with accurate predictions. As shown in Figure 2(a), as the penetration rate increases, the average delay significantly decreases. This is because PCPV strives to have the replicas pre-cached at the proper time and place for the requesters to acquire either before the time of their request or within the deadline specified. However, the lower the percentage of vehicles with accurate predictions, the higher the delay. This is attributed to the risk of having replicas cached at road segments that proactive requesters encounter after the deadline.

Second, we apply the same comparison in terms of the second metric. As shown in Figure 2(b), as the penetration rate increases, the packet delivery ratio is significantly improved. This is because in PCPV, the data is delivered to the requesting vehicle as it passes by the caching parked vehicle. This eliminates the risk of unsuccessful packet delivery caused by not knowing the requester's location. However, the higher the percentage of vehicles with erroneous predictions, the lower the packet delivery ratio. This is attributed to the risk of caching replicas at road segments that proactive requesters encounter either before the data has been cached or after it has become obsolete.

Third, we perform the same experiment to assess the cache hit ratio. As depicted in Figure 2(c), the increase in the penetration rate significantly improves the cache hit ratio. In PCPV, proactive requests are always served by a caching node and thus all packets delivered are attributed to a cache hit. However, as the percentage of vehicles with erroneous predictions increases, the number of packets successfully delivered to the proactive requesters decreases. Hence, the cache hit ratio decreases. This occurs for the same reasons discussed above. In contrast, in the reactive scheme, requests can either be served by a caching node or by the data center. PCPV enables reactive requests to be served by replicas intended for proactive requesters (i.e. the data cached at parked vehicles). In addition, all moving vehicles along the forwarding path from the data center to the caching parked vehicles keep a copy of the data in their local cache. Thus, the higher the number of proactive requesters, the more replicas sent by the data center, which increases the availability of cached items at moving and parked vehicles. This boosts the possibility of reactive requests being served by a caching node.

Fourth, we test the performance of PCPV in terms of the satisfaction ratio under different values of α, over varying percentages of vehicles with accurate predictions. In this experiment, the penetration rate is set to 100%. As shown in Figure 2(d), the satisfaction ratio significantly increases



(a) Average delay of PCPV and the reactive scheme.
(b) Packet delivery ratio of PCPV and the reactive scheme.
(c) Cache hit ratio of PCPV and the reactive scheme.
(d) Satisfaction ratio of PCPV under varying α.

Fig. 2. Performance Results over Varying Percentages of Vehicles with Accurate Predictions.

as α increases. This is because as α decreases, the risk of having proactive requesters with no available road segments for caching along their trajectory increases. In addition, decreasing α limits the number of road segments available for cache selection. This can force the data center to place replicas at road segments with which proactive requesters have low probability of encounter. This poses a risk of having the data cached at road segments that proactive requesters encounter after the deadline. Such risk further increases as the percentage of vehicles with erroneous predictions increases.

V. Conclusion and Future Work

In this paper, we have proposed the PCPV scheme that provides better quality of service for vehicular users who have a predictable social media access behavior. To do so, PCPV uses parked vehicles to pre-cache the data at the proper time and place for users to proactively receive as they pass by. We have introduced a greedy heuristic approach that takes into consideration all the time-constraint issues required to select the most suitable road segments for caching. To maximize cache hits, the probability of encounter of proactive users with road segments along their trajectory is also taken into consideration. In addition, PCPV enables reactive requesters to benefit from the replicas cached at parked and moving vehicles. Performance evaluation has shown that PCPV achieves significant improvements in terms of access delay, packet delivery ratio, and cache hit ratio compared to the reactive scheme, while maintaining proper satisfaction ratio. In our future work, we will consider developing a prediction module that limits the error, in the estimated time of encounter with road segments, to a maximum bound.

References

[1] S. Ilarri , T. Delot , R. Trillo, *"A Data Management Perspective on Vehicular Networks"*, IEEE Communications Surveys and Tutorials, vol. 17, no. 4, pp. 2420-2460, 2015.

[2] A. Adiv, and W. Wang, *"On-Street Parking Meter Behavior"*, Transportation Quarterly, vol. 41, pp. 281–307, 1987.

[3] J. Zhao, Y. Zhang, and G. Cao, *"Data Pouring and Buffering on the Road: A New Data Dissemination Paradigm for Vehicular Ad Hoc Networks,"* in IEEE Transaction on Vehicular Technology, vol. 56, no. 6, pp. 3266–3276, 2007.

[4] G. M. Brito, P. B. Velloso, and I. M. Moraes, *"Information Centric Networks: A New Paradigm for the Internet"*, Wiley, 2013.

[5] *"Mobiles Hierarchy of Needs Revealing Key Insights into Global Mobile Trends"*, Available: https://www.comscore.com/Insights/Press-Releases/2017/3/comScore-Releases-New-Report-Mobiles-Hierarchy-of-Needs, (Accessed 2017, May 3rd), 2017.

[6] *"Number of social media users worldwide from 2010 to 2020 (in billions)"*, Available: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/, (Accessed 2017, May 3rd), 2017.

[7] E. Esano *"Top 100 Instagram Users by Followers"*, Available: https://socialblade.com/instagram/top/100/followers, (Accessed 2017, May 3rd), 2017.

[8] M. Gerla, B. Zhou, Y.-Z. Lee, F. Soldo, U. Lee, and G. Marfia, *"Vehicular Grid Communications: The Role of the Internet Infrastructure"*, in Second Annual International Workshop on Wireless Internet (WICON06), pp. 1-10, ACM, 2006.

[9] H. Saleet and O. A. Basir, *"Location based Message Aggregation in Vehicular Ad Hoc Networks"*, in IEEE Global Communications Workshops, , pp. 1-7, IEEE Computer Society, 2007.

[10] S. Abdelhamid, S. A. Elsayed, T. Kumar, H. S. Hassanein, *"Caching and Forwarding Assistance for Vehicular Information Services with Mobile Requesters"*, in the IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops), pp. 832–839 , 2015.

[11] N. Loulloudes, G. Pallis, and M. D. Dikaiakos, *"Caching Dynamic Information in Vehicular Ad Hoc Networks"*, ser. Euro-Par 2010-Parallel Processing. Springer, pp. 516–527, 2010.

[12] B. Xu, A. Ouksel, and O. Wolfson, *"Opportunistic Resource Exchange in Inter-vehicle Ad Hoc Networks,"* in Proceedings IEEE International Conference MDM, pp. 4-12, 2004.

[13] H. Gong and L. Yu, *"Content Downloading with the Assistance of Roadside Cars for Vehicular Ad Hoc Networks"*, Mobile Information Systems, vol. 2017, Article ID 4863167, 9 pages, 2017.

[14] N. Liu, M. Liu, G. Chen, and J. Cao, *"The Sharing at Roadside: Vehicular Content Distribution Using Parked Vehicles"*, in proceedings of the 31st IEEE Conference on Computer Communications (INFOCOM 2012), Mini-Conference, Orlando, FL, March 2012.

[15] Z. Su, Q. Xu, Y. Hui, M. Wen, and S. Guo, *"A Game Theoretic Approach to Parked Vehicle Assisted Content Delivery in Vehicular Ad Hoc Networks"*, IEEE Transactions on Vehicular Technology, vol. 66, no. 7, July 2017.

[16] R. Atawia, H. Abou-zeid, H. Hassanein, and A. Noureldin, *"Joint Chance Constrained Predictive Resource Allocation for Energy-Efficient Video Streaming"*, IEEE Journal on Selected Areas in Communications, vol.34, no. 5, pp. 1389–1404, 2016.

[17] H. Abu-Ghazaleh and A. Alfa, *"Application of Mobility Prediction in Wireless Networks Using Markov Renewal Theory"*, IEEE Trans. Veh.Technol., vol. 59, no. 2, pp. 788–802, Feb. 2010.

[18] S. N. Trambadiya, P. A. Ghosh, and J. Rathod, *"Group Caching: A Novel Cooperative Caching Scheme for Mobile Ad Hoc Networks"*, International Journal of Engineering Research and Development, vol. 6, no. 11, pp. 23–30, 2013.

[19] The NS-3 Network Simulator. [Online]. Available: https://www.nsnam.org/ (Accessed 2017, September 25).

[20] SUMO (Simulation of Urban Mobility). [Online]. Available:http://sumo-sim.org/ (Accessed 2017, September 25)