# Tracking-based Cooperative Caching in VANETs for Social Networking

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

***Abstract*—Social media traffic is the predominant source of Internet traffic. Such traffic is primarily facilitated by mobile devices. As a result, mobile users tend to sustain high cellular costs. To alleviate such costs, we endeavor to allow social media users to depend more on vehicular rather than cellular networks for data access. However, this can be thwarted by the high delay and low packet delivery ratio often coupled with content access from distant data providers in vehicular networks. Hence, we strive to get the data closer to the requester by proposing the Cooperative Content Discovery and Placement at Vehicles (CCDPV) scheme. In CCDPV, we dynamically discover closer replicas to the requester using parked and moving vehicles. In particular, we utilize the last encounter information, as well as the static and mobile nature of parked and moving vehicles, respectively, to diffuse cached content information, track caching nodes, and dynamically locate closer caching nodes to the requester. CCDPV applies cooperative cache placement at both parked and moving vehicles. It caches data at parked vehicles located in congested road segments to cater for the higher possibility of requests occurring at them. CCDPV caches data at moving vehicles while taking data diversity relative to the vehicles direction of movement into consideration. Performance evaluation shows that CCDPV significantly improves delay, packet delivery ratio, and cache hit ratio compared to a caching scheme in vehicular networks that has some implicit cooperative features, as well as to a tracking-based cooperative cache discovery scheme in mobile ad hoc networks.**

***Index Terms*—VANETs, Cooperative Caching, Parked Vehicles.**

## I. INTRODUCTION

Social media usage has been widely proliferating throughout the globe. In 2017, social media users amounted to 71% of Internet users [1]. Such a proliferation is expected to further amplify in the future, with an estimated increase of up to 2.9 billion social media users in 2020 [1]. More than 60% of such excessive usage is primarily facilitated by mobile devices [2]. This causes mobile users to disburse high cellular costs, specially in outdoor areas where Wi-Fi connection is not available. One solution to reduce such costs is to provide social media users with the ability to depend more on other types of free/less costly networks for content access. Vehicular Ad Hoc Networks (VANETs) are prime candidates to consider for such an objective, due to their pervasive availability.

VANETs have manifested as a communication paradigm that promotes interconnection among vehicles on the road. They have been considered a key enabler of a broad spectrum of applications in Intelligent Transportation Systems (ITS), including infotainment services, such as location-based services and Internet access [3]. However, the strenuous issues associated with data access in VANETs can have a profound impact on the quality of service yielded in these applications.

Data access in VANETs can be attained through communication between vehicles and roadside access points, commonly known as Road Side Units (RSUs) [3]. However, massive investments are typically required to ensure vast deployment of RSUs, and thus they may not be intensely deployed [3]. Accordingly, the closest available RSU might be deployed at a significantly remote position from the requester. Hence, Vehicle-to-Vehicle (V2V) communication tends to be the predominant type of communication relied upon by Internet users in VANETs to reach the closest RSU [3]. V2V communications intended for remote content providers are often coupled with high delay and low packet delivery ratio [3]. This is due to the unique characteristics of VANETs, particularly their intermittent connectivity and highly dynamic topology [3]. In addition, VANET-based Internet access typically incorporates a request-response data access paradigm [3]. That is, a request is issued towards the data center, reached via a RSU, and a response is directed back to the requester. This could further exacerbate the quality of service.

In order to allow social media users to consider VANETs as an expedient alternative to cellular networks for data access, it is imperative to ameliorate the quality of Internet service in VANETs. Thus, in this paper, we propose the Cooperative Content Discovery and Placement at Vehicles (CCDPV) scheme. CCDPV is designed to bring the data closer to the requester through cooperative caching, with more emphasis on cooperative cache discovery (i.e., content discovery). Cooperative caching has been recognized as an advantageous technique for enhancing the performance of content access in different network paradigms, including Mobile Ad Hoc Networks (MANETs) [4] and Information-Centric Networks (ICNs) [5]. In cooperative caching, the nodes employ a form of collaboration in order to make informed caching decisions, and/or they establish a cooperative cache by sharing cached data [4]. Such a cooperative cache can be consulted in case of a local cache miss [4]. To do so, nodes tend to trade information related to the data they hold in their cache. Cooperative caching has been shown to bring the data closer to the requester, create increased data diversity, and achieve

978-1-5386-5097-4/18/$31.00 ©2018 IEEE

efficient utilization of the nodes' cache capacity [4][5], which in turn increases cache hits. This is crucial when dealing with substantial amount of contents, as the case in social media. However, despite its substantiated leverage in MANETs and ICNs [4][5], cooperative caching has been rarely lucubrated within the context of VANETs. This can be attributed to the extremely dynamic nature of vehicles, which shortens the lifetime of the cached content information, since they tend to rapidly become obsolete as vehicles quickly change their positions. This can trigger frequent instabilities in caching decisions, including cache discovery decisions.

One of the most important decisions in cooperative caching is the one made during the cache discovery process [4]. Discovering where the content is cached is the first process that needs to be performed when a requesting node sends a request. Many existing cooperative cache discovery schemes depend on some form of information exchange for tracking cached contents [4]. Such cached content information can be used to navigate requests towards nearby caching nodes rather than blindly directing them towards the far-away data center. This type of schemes is referred to as tracking-based schemes [4]. Such schemes can achieve reduced overhead and delay compared to broadcast-based schemes, where requests are flooded [4]. However, in dynamic networks, intensive number of messages might need to be exchanged to maintain up-to-date tracking information, which could lead to huge amount of overhead [4]. Thus, most existing tracking-based schemes in MANETs restrict information exchange to neighboring nodes only [4]. However, this limits the search space, and can cause failure to locate caching nodes [4].

In order to tackle the aforementioned problems, we utilize the static nature of parked vehicles to create a rather stable residence for cached content information. We do so to keep the information received from encountered vehicles alive at road segments for later use. In addition, we leverage such a static nature to provide a more stable tracking service of the movement of moving vehicles, including that of caching nodes. This further increases the lifetime of the cached content information. We rely on beacon messages that are typically exchanged periodically between neighboring nodes [3], as well as the mobile and static nature of moving and parked vehicles, respectively, to diffuse cached content information within the network. This diffusion occurs as parked and moving vehicles exchange certain information upon encounter, including their cached content information, via beacon messages. This information diffusion expands the search space without the need for sending extra messages. In contrast to RSUs, it has been substantiated that parked vehicles are available as natural infrastructures that are widely spread in huge numbers [6]. Hence, they can act as a source of substantial and cost-effective caching resources. A study scrutinizing on-street parking spaces in Ann Arbor city in the US, has shown that their occupancy ratio amounts to 93% and 80% in on and off-peaks, respectively [6].

To the best of our knowledge, CCDPV is the first cooperative caching scheme within VANETs that uses a tracking-based cache discovery procedure to dynamically locate closer caching nodes to the requester. In addition to cache discovery, CCDPV employs a cooperative cache placement scheme at both parked and moving vehicles. It caches data at parked vehicles located in congested road segments to cater for the higher possibility of requests occurring at or passing by them. CCDPV also caches data at moving vehicles while considering data diversity relative to the vehicles direction of movement.

We assess the performance of the proposed CCDPV scheme via the NS-3 simulator. We compare it to the Caching-Assisted Data Delivery (CADD) scheme in VANETs [7]. This is since CADD implicitly exhibits a somewhat cooperative caching behavior. In addition, in order to explicitly evaluate the performance of our proposed tracking-based cache discovery scheme, we compare CCDPV to a tracking-based cooperative cache discovery scheme in MANETs, namely the GroupCaching scheme [8]. Simulation results show that CCDPV achieves significant improvements in terms of access delay, packet delivery ratio, and cache-hit ratio, compared to CADD, and GroupCaching.

The rest of the paper is organized as follows. Section II highlights some related work. Section III presents the proposed scheme (CCDPV). Section IV illustrates the performance evaluation, as well as the simulation results, of the scheme. Section V discusses our conclusions and future work.

## II. RELATED WORK

In this section, we present some related work in cooperative caching in MANETs and ICNs, as well as caching in VANETs.

### *A. Cooperative Caching in MANETs and ICNs*

Various cooperative cache discovery schemes and cache placement schemes have been designed for MANETs and ICNs in the literature. In [9], a broadcast-based cache discovery scheme, where requests are flooded into the network, is applied. On the one hand, this approach expands the search area and thus expedites the discovery process. On the other hand, it can be significantly costly in terms of bandwidth, and can have a severe impact on the overall traffic load on the network [4]. In [10], a server-based approach is used for cache discovery. This approach limits the search space as it relies on finding the cached data at intermediate nodes encountered by the request packet en route to the server. However, it offers no guarantee that any of those nodes has the cached data [4]. In GroupCaching [8], a tracking-based cache discovery scheme is employed. In this scheme, nodes maintain some information about the cached contents of their 1-hop neighbors. As previously mentioned, tracking-based schemes can reduce the communication overhead and delay compared to broadcast-based schemes [4]. However, in dynamic networks, maintaining up-to-date lookup tables might require the exchange of large number of messages, and thus large overhead. Hence, most of these schemes often restrict the range of information exchange to neighboring nodes only, such as GroupCaching. Once two nodes are no longer connected, their corresponding cached content information are invalidated. This limits the search space and thus might result in many requests being eventually directed to the far-away data center.

In our proposed scheme, we expand the search space by exploiting the static and mobile nature of parked and moving vehicles, respectively, as well as the fact that beacon messages are typically exchanged between neighboring vehicles in a periodic manner. Such components are used to diffuse cached content information into the network, and to track the positions of the corresponding caching nodes, without the need for sending extra messages. Meanwhile, most existing cooperative cache placement schemes, such as GroupCaching [8], strive to increase the diversity of cached data through collaboration. Such a collaboration often takes place between the nodes along the delivery path or between a node and its neighbors [5]. However, the mobility of nodes is typically overlooked in such schemes [4]. In our proposed scheme, we consider the vehicles direction of movement, so as to ensure data diversity among vehicles heading in the same direction.

### *B. Caching in VANETs*

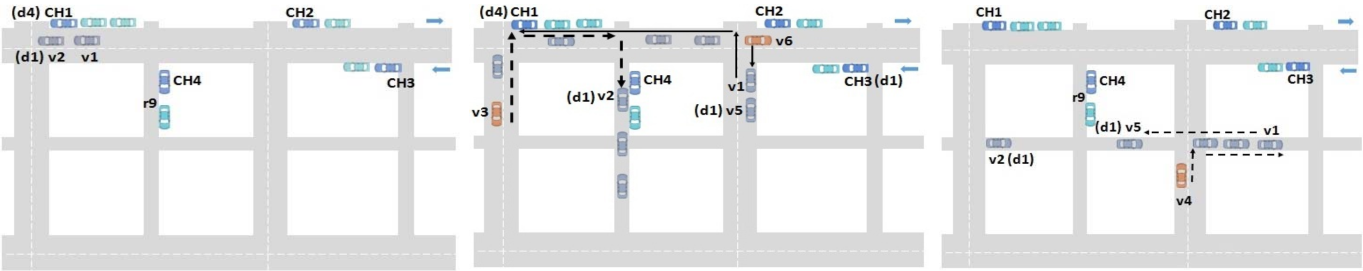
Caching has been used in very few schemes in VANETs [7][11][12]. In [11], a non-cooperative caching scheme is proposed, where caching occurs at every intermediate node along the data delivery path. In CADD [7], caching only takes place at static nodes, called Road Caching Spots (RCSs), located at intersections. An implicit form of collaboration is performed among on-path RCSs to dynamically choose the one that receives the largest number of requests for caching. Thus, some aspects of cooperative caching are implicitly inherited in CADD. In [12], a non-cooperative caching procedure is applied at moving vehicles. In this procedure, caching depends on content popularity, the betweenness and degree centrality of the vehicle relative to its neighbors, as well as the respective direction of movement between the content consumer and provider. Roadside parked vehicles have been recently used to cache large-sized contents in some schemes [13]. In these schemes, large contents are broken down into smaller chunks and the sequential nature of parked vehicles is exploited to store them, so as to be subsequently procured by moving vehicles as they pass by. However, the main focus of such schemes is directed towards content downloading rather than caching itself [13]. Most existing caching schemes in VANETs do not use an explicit form of cooperation or cached content information exchange among nodes. In addition, they typically rely on a server-based cache discovery approach [7][11][12], which significantly limits the search space, and can lead to reduced cache hits [4]. In our proposed scheme, we apply an explicit cooperative caching scheme at both parked and moving vehicles to increase data availability and diversity, and thus improve cache hits. We also exploit the static and mobile nature of parked and moving vehicles, respectively, to apply a tracking-based cache discovery procedure in order to dynamically locate closer caching nodes to the requester.

## III. COOPERATIVE CONTENT DISCOVERY AND PLACEMENT AT VEHICLES (CCDPV)

As previously mentioned, our goal is to ameliorate the quality of VANET-based Internet service, to allow social media users to exhibit more reliance on vehicular rather than cellular networks for content access. Thus, we aim at bringing the data closer to the requester by leveraging the use of cooperative caching within VANETs. We do so with more emphasis on cooperative cache discovery. The purpose is to make more informed caching decisions, including tracking-based cache discovery decisions that aim at dynamically locating closer caching nodes to the requester. We depend on the exchange of certain information between moving and parked vehicles via beacon messages. We refer to such information as the last encounter information. This includes the typically exchanged information in beacon messages (i.e., the vehicle's position, speed, and heading) [3], as well as the vehicle's cached content information, the next road segment to which it is heading, and a certain time threshold. As explained later, this additional information is used for tracking purposes.

We assume that requesting vehicles are targeting social media platforms that do not possess large-sized contents, such as Instagram. We are primarily concerned with the accounts of public figures, such as those of politicians, music stars, actors, etc. This is due to the usefulness of caching such data, since they capture the interest of many users. The original data provider (i.e., the data center) gauges the posting frequency of a given public figure. This reflects the Time to Live (TTL) of the data. TTL represents the anticipated time before a new post gets published by the public figure, thus indicating the staleness of the previously posted data. The data center appends the TTL of the data to the data packet that it creates. The data expiry time is deduced based on that value. Vehicles agree to allocate certain amount of their storage capacity to the caching service in exchange for some incentives. We assume the availability of parked vehicles at each road segment. Generally, traffic statistics, such as the traffic density and average speed of vehicles at each road segment are accessible to vehicles via some navigation service. Each moving vehicle, $v_i$, is aware of its own trajectory and is willing to share the next road segment to which it is heading, denoted $r^i_{next}$, with other vehicles. Unique names are assigned to different contents, as the case in named data networking [12].

Parked vehicles at each road segment form a cluster, managed by a scheme similar to the one in [13]. To ensure data diversity, a cluster head (CH) is chosen to make caching decisions at all parked vehicles in its cluster. Thus, the CH is responsible for maintaining the cached content information within its cluster. It is also responsible for interchanging such information with neighboring vehicles through the exchange of beacon messages. Hence, the nearest parked vehicle to the entrance of the road segment is appointed as the CH. This is to guarantee that moving and parked vehicles exchange the necessary information once the former arrive at the road. In case of a two-way road segment, if the length of the latter exceeds the transmission range, two parking clusters are created, with two CHs, one at each end of the intersection. In order to enhance data diversity, if there are two clusters in the same road segment, the cached content information at one cluster reaches the other through moving vehicles.



(a) At Time $t_1$. (b) At Time $t_2$. (c) At Time $t_3$.

Fig. 1. An Illustrative Scenario. (a) At Time $t_1$. The moving vehicle $v_2$, which has the data, $d_1$, in its cache, encounters the moving vehicle, $v_1$, and the cluster head, $CH_1$. The parking cluster of $CH_1$ has the data, $d_4$, in its cache. The next road segment to which vehicle $v_2$ is heading, is $r_9$. Based on the received beacon messages, each vehicle updates its TPP. (b) At Time $t_2$. Vehicles $v_1$ and $v_2$ have already moved away. $CH_1$ receives a request for $d_1$ issued by the requesting vehicle $v_3$ and destined to $CH_3$ (dotted line). Note that $CH_3$ is a caching node that was previously encountered by $v_3$. Upon receiving the request, $CH_1$ consults its TPP and determines that $v_2$ is closer to the requester than $CH_3$. Thus, it directs the interest packet to $v_2$ (dotted line). Similarly, when $v_1$ receives a request for $d_4$ from the requesting vehicle $v_6$, it consults its TPP and directs the packet to $CH_1$ (solid line). Meanwhile, $v_1$ encounters the caching vehicle, $v_5$; $v_1$ updates its TPP. (c) At Time $t_3$. The requesting vehicle, $v_4$, sends a request for $d_1$ to the distant data center. When $v_1$ receives the interest packet during the forwarding process, it checks its TPP and ranks the previously encountered data holders, $v_2$ and $v_5$, to select the closest node to the requester; $v_5$ is selected. Note that if $v_2$ was selected, it would have been reachable via $CH_4$, since the latter knows where $v_2$ headed after $r_9$.

In CCDPV, each vehicle maintains a List of Cached Data (LCD), containing the names of its own cached data items. Using beacon messages, each vehicle sends its last encounter information to all of its neighbors. If the sender of the beacon message is a CH, the last encounter information consists of its position, as well as the LCD and available cache space of its cluster. If the sender of the beacon message is a moving vehicle, $v_i$, the last encounter information is composed of its LCD, available cache space, position, speed, heading, the next road segment to which it is heading, $r^i_{next}$, and its estimated time threshold, $t_{i,next+1,e}$. This time threshold is defined as the vehicle's estimated time of departure from $r^i_{next+1}$, where $r^i_{next+1}$ is the road segment to which $v_i$ is heading after $r^i_{next}$. It is used to indicate the upper limit on the time during which $v_i$ is known to be located within close proximity to the last node that knows where it headed, as explained later in details. The vehicle, $v_i$, calculates $t_{i,next+1,e}$ based on the sum of its estimated travel time along the current road segment, $r^i_{cur}$, as well as $r^i_{next}$ and $r^i_{next+1}$. The travel time of $v_i$ at any given road segment, $r_j$, at time, $t_{i,j,s}$, where $t_{i,j,s}$ is the estimated time of arrival of $v_i$ at $r_j$, is denoted $\tau_j^{t_{i,j,s}}$. The value of $\tau_j^{t_{i,j,s}}$, given by Eq. 1, is calculated based on the length of $r_j$, $L_j$, and the estimated average velocity of vehicles on $r_j$ at time $t_{ij,s}$, $V_j^{t_{i,j,s}}$.

$$\tau_j^{t_{i,j,s}} = \frac{L_j}{V_j^{t_{i,j,s}}} \tag{1}$$

Each vehicle that receives a beacon message associates the received last encounter information with a timestamp and maintains it in a Table of Possible Providers (TPP). If the sender of the beacon message is a moving vehicle, $v_i$, the receiving vehicle determines the estimated time of arrival and departure of $v_i$ to and from $r^i_{next}$, denoted $t_{i,next,s}$ and $t_{i,next,e}$, respectively. The former and latter are calculated using the estimated travel time of $v_i$ on $r^i_{cur}$ and $r^i_{next}$, respectively. The receiving vehicle then adds this information to the TPP for later use. Note that the TPP provides information about the currently and previously encountered vehicles that hold the data in their cache and that can thus act as data providers. Even if the LCD in the received beacon message is empty, the last encounter information is still maintained in the TPP of the receiving vehicle. This is done for location tracking purposes. Parked vehicles subscribing to the caching service maintain a similar table. When a node receives a request packet, it ranks each caching node in the TPP based on its proximity to the requester, as well as the age of information. Based on the ranks, the request can be directed to a data holder that is closer to the requester than the current destined data provider. Initially, the latter is the data center. An entry in the TPP about a caching node is considered obsolete if the corresponding data reaches its expiry time. To demonstrate the leverage of the scheme, consider the illustrative scenario depicted in Figure 1. In the next subsections, we describe the cooperative cache discovery and cache placement procedures of CCDPV at both parked and moving vehicles.

### A. CCDPV Cache Discovery at Moving and Parked Vehicles

This procedure is either triggered by a requesting vehicle or a vehicle forwarding an interest packet, $v_f$. Initially, the current data provider to which the interest packet is directed, is the data center. Note that the interest packet is associated with the last time of encounter of the requesting vehicle. This indicates the last time at which the requesting vehicle has been located at the position included in the packet. Also, the last time of encounter of the current data provider, $v_c$, and its $t_{c,next+1,e}$ (if it is a moving vehicle), are also included in the interest packet. As illustrated in Algorithm 1, the cache discovery procedure is executed as follows:

(a) Upon receiving an unexpired interest packet, the vehicle, $v_f$, tracks the most recent location of the requester, $v_k$. The purpose of this tracking procedure is twofold. First, since we aim at finding a closer data provider to the requester, we strive to determine the most recently observed position of the latter. Second, we endeavor to alleviate the problem associated with the fact that by the time the data is issued back to the

**Algorithm 1** : CCDPV Cache Discovery Procedure

```
1: Input:
2: Forwarding Vehicle $v$
3: Interest Packet $I$
4: Reply Packet $r$
5: Requested Data $D$
6: Requesting Vehicle $Req$.        //source of $I$
7: Current Data Provider $C$        //destination of $I$
8: Neighborhood list $N$
9:
10: cache_discovery(I)
11: Begin
12:   if $I$ is not expired then
13:       if $Req$ is recorded in $v$'s TPP then
14:           if $t_{Req}^{TPP} \geq t_{Req}^{I}$ then        //$t$ is the last time of encounter
15:               $NewPos_{Req} =$ track_newPos(I, Req)        //Updated Pos.
16:               $t_{Req} = t_{Req}^{TPP}$        //Updated-$t$ recorded in $TPP$
17:               Update the $Req$ position and $t_{Req}$ in $I$
18:       if there is $D$ matching $I$ in the cache then
19:           generate a reply $r$
20:           forward $r$
21:       else if any node in $N$ has $D$ in its cache then
22:           update $C$ in $I$        //The neighbor with the cached data
23:           forward $I$
24:       else
25:           if $C$ is a moving vehicle then
26:               if $C$ is recorded in $v$'s TPP then
27:                   if $t_{C}^{TPP} \geq t_{C}^{I}$ then
28:                       $NewPos_{C} =$ track_newPos(I, C)
29:                       $t_{C} = t_{C}^{TPP}$        //Updated-$t$ recorded in $TPP$
30:           if $v$ is in the range of $C$'s Position then
31:               if $C$ is not in $N$ then
32:                   $C$=data center
33:           if ID of $D$ matches an entry in $v$'s TPP then
34:               determine $S$        //Set of possible providers of $D$ in the TPP
35:               for all $u \in U$ do        // $U = S \cup$ the data center $\cup C$
36:                   if $u$ is a moving vehicle then
37:                       $NewPos_{u} =$ track_newPos(I, u)
38:                       calculate $rank_{u}$ using Eq. 4
39:                   else
40:                       calculate $rank_{u}$ using Eq. 5
41:               Calculate $rank_{max} = \max_{u \in U} rank_{u}$
42:               $C = arg \max_{u \in U} rank_{u}$
43:       update $C$, its position $t_{C}$, $t_{c,next+1,e}$ in $I$        //if any changed
44:       forward $I$
45: End
```

requester, its position might have significantly changed. Thus, the packet might be dropped if the requester cannot be tracked. In CCDPV, when $v_k$ passes by a neighboring node, including a CH, it sends information about the next road segment to which it is heading. Due to the static nature of parked vehicles, it is possible to reach $v_k$ by following its trails, via the CH in the road segment where it has last been seen, $r_{last}$. Note that even if the recent position of the requester cannot be closely estimated, it is possible to expedite the process of data access by finding a data holder that is closer to $r_{last}$ than the current data provider. This is since $v_k$ is reachable via $r_{last}$. The tracking procedure works as follows (lines 12-15): when a vehicle, $v_f$, receives an unexpired interest packet, it checks if the requesting vehicle is among the vehicles registered in its TPP (i.e., $v_k$ has been previously encountered by $v_f$). If it is, and the recorded time of encounter in the TPP, $t_k$, is more recent than the time of encounter associated with the interest packet, the requester's new position is estimated according to the information in the TPP (i.e., its position, speed, heading, $r^k_{next}$, and $t_{k,next+1,e}$). Otherwise, its position remains the same as that indicated in the interest packet. In order to estimate the new position of $v_k$, $v_f$ considers the following three cases:

1) The vehicle, $v_k$, has not reached $r^k_{next}$ yet. In other words, it is still moving on $r^k_{cur}$. This is the case if the current time, $t_{cur}$, is less than the estimated time of arrival of $v_k$ in $r^k_{next}$ ($t_{cur}<t_{k,next,s}$). Thus, the new position of $v_k$ can be estimated based on its old position, which is recorded in the TPP of $v_f$, and the estimated distance that it has traversed on $r^k_{cur}$ since the last time of encounter, $t_k$, denoted $D^{t_k}_{k,cur}$. The latter is calculated based on the vehicle's speed and heading (i.e. velocity vector, $\overrightarrow{V_k}$), as given by Eq. 2.

$$D^{t_k}_{k,cur} = \overrightarrow{V_k}(t_{cur} - t_k) \tag{2}$$

2) The vehicle, $v_k$, has reached $r^k_{next}$, and it is still there (i.e., $t_{k,next,s}<t_{cur}<t_{k,next,e}$). In this case, its new location can be estimated based on its last known position (i.e., the start point of $r^k_{next}$), and the total estimated distance that it has traversed on $r^k_{next}$ since its time of arrival there, $t_{k,next,s}$, denoted $D^{t_{k,next,s}}_{k,next}$. This estimated distance, is given by Eq. 3, where $\overrightarrow{V^{t_{k,next,s}}_{next}}$ is the velocity vector. This velocity vector is determined based on the heading of $v_k$ on $r^k_{next}$ and the estimated average speed of vehicles on $r^k_{next}$ at time $t_{k,next,s}$.

$$D^{t_{k,next,s}}_{k,next} = \overrightarrow{V^{t_{k,next,s}}_{next}}(t_{cur} - t_{k,next,s}) \tag{3}$$

3) The vehicle, $v_k$, has reached $r^k_{next+1}$ (i.e., $t_{cur}>t_{k,next,e}$). In this case, the new position of $v_k$ can be tracked through the CH at $r^k_{next}$, $CH^k_{next}$, since it would have information about $r^k_{next+1}$. Hence, in this case, we set the new position of $v_k$ to that of $CH^k_{next}$. Note that if $r^k_{next}$ has two CHs (i.e., it is a two-way road whose length exceeds the communication range), it is better to track $v_k$ via the CH that is near $v_k$'s exit point from $r^k_{next}$. This is since it is closer to $r^k_{next+1}$ than the CH near its entry point. If $v_k$ has already left $r^k_{next+1}$ (i.e., $t_{cur}>t_{k,next+1,e}$), the same logic is applied and the new position of $v_k$ is set to that of $CH^k_{next}$. Note that $CH^k_{next}$ has information about $r^k_{next+1}$ and the cluster head at $r^k_{next+1}$ would also have information about $r^k_{next+2}$, and so on.

Accordingly, the position of the requester and its associated last time of encounter are updated in the interest packet (lines 16 & 17). Note that when the data is found, the requester's position and its last time of encounter, are copied in the reply packet. The requester tracking process is applied by all vehicles along the data delivery path as well.

(b) The vehicle, $v_f$, checks if it has a match of the requested data in its own local cache (i.e., $v_f$ is an intermediate caching node or the destination of the interest packet). If so, the vehicle issues a reply packet and sends it back to the requester (lines 18-20).

(c) If not, $v_f$ checks if any of its 1-hop neighbors, denoted $N$, has the data in its cache. If so, it sends the interest packet to it (lines 21-23).

(d) Otherwise, if the packet's destination (i.e. the current data provider) is a moving vehicle, $v_f$ tracks its most recent position. To do so, it applies the same tracking procedure applied for the requester (lines 24-29).

(e) If the estimated position of the current data provider is within the communication range of $v_f$, but the former cannot be found, then the current data provider is set to the data center. Otherwise, the current data provider remains the same (lines 30-32).

(f) The vehicle, $v_f$, checks its TPP. If an entry matching the name of the requested data is found in the TPP, it determines the associated set of vehicles in the table that can act as potential providers of the requested data, denoted $S$. The vehicle then assigns a rank to each node $v_i \in S$. This rank assesses the usefulness of the node as a potential data provider and the benefit of forwarding the interest packet towards it rather than the current data provider. The rank is based on two factors: 1) The age of information. That is, the older the information stored about the vehicle, the less the accuracy of its estimated position. In particular, the longer it has been since $v_i$ left its $r^i_{next+1}$, the further it is from where it can be reached (i.e., $CH_{next}$), and thus the less reliable its proximity information. Accordingly, if the current time exceeds the estimated time threshold of $v_i$, $t_{i,next+1,e}$, by more than a certain time step, the rank of $v_i$ is set to zero. As previously mentioned, $t_{i,next+1,e}$, indicates the estimated departure time of $v_i$ from $r^i_{next+1}$. 2) The second factor based on which the rank is calculated, is the estimated distance between the vehicle caching the data and the requester. The closer it is to the requester, the higher the rank. Thus, in order to calculate the ranks, if a vehicle $v_i \in S$ is a moving vehicle, its most recent location must be estimated before calculating its rank. In this case, the vehicle, $v_f$, tracks the most recent location of $v_i$ based on the last encounter information registered in the TPP, and using the same tracking procedure applied for the requester. Taking the new estimated positions into consideration, the vehicle calculates the rank of $v_i$, denoted $rank^m_i$, using Eq. 4, where $d_{iReq}$ is the distance between the requester and the potential provider (i.e., $v_i \in S$, or the current data provider, $v_c$, or the data center), $t_{cur}$ is the current time, $t_i$ is the last time of encounter with vehicle $v_i$ as recorded in the TPP, $t_{max}$ is the most recent time of encounter among that of $v_c$ and all moving vehicles in $S$, $d_{min}$ is the minimum distance between the requester and all possible providers, $\Delta$ is a certain time step, and $\alpha$ and $\beta$ are weighting factors set in the $(o, 1]$ range; $\alpha + \beta$=1. If $v_i$ is a parked vehicle, its rank, $rank^p_i$, is calculated using Eq. 5. In addition, $v_f$ ranks the current data provider, $v_c$, using Eq. 4 or Eq. 5. Note that if $v_c$ is a moving vehicle, its $t_{c,next+1,e}$ and last time of encounter, are associated with the interest packet. The vehicle also ranks the data center using Eq. 5, since it is the original data provider and it might be better to direct the packet to it (lines 33-40).

$$rank^m_i = \begin{cases} 0 & t_{cur} > t_{i,next+1,e} + \Delta \\ \alpha \frac{t_i}{t_{max}} + \beta \frac{d_{min}}{d_{iReq}} & \text{Otherwise} \end{cases} \tag{4}$$

$$rank^p_i = \frac{d_{min}}{d_{iReq}} \tag{5}$$

(g) The maximum rank among that of all $v_i \in S$, the current data provider, and the data center, is then determined. The node that has the maximum rank is selected as the current data provider and the packet's destination is updated. The last time of encounter, as well as the new estimated position of the destination and its $t_{c,next+1,e}$, are also updated in the packet (lines 41-43).

(h) The vehicle, $v_f$, anchors the packet towards the estimated position of the current data provider using the following forwarding procedure (line 44): 1) If there is a CH in the neighborhood of $v_f$ that is more adjacent to the destination than itself, $v_f$ forwards the packet to it. This is to enable the packet to encounter as many CHs as possible to benefit from the information maintained in their TPPs in the discovery process. 2) Otherwise, greedy forwarding is used to direct the packet towards the destination. In greedy forwarding, the nearest neighboring node to the destination is the one to which the packet is forwarded. Aside from CHs, forwarding occurs using moving vehicles only. However, if $v_f$ fails to find any moving vehicle within its neighborhood, it is possible to forward the packet to parked vehicles. Note that data packets are forwarded using the same forwarding procedure. The benefit of step 1 in data forwarding is to make cache placement decisions in as many parking clusters as possible.

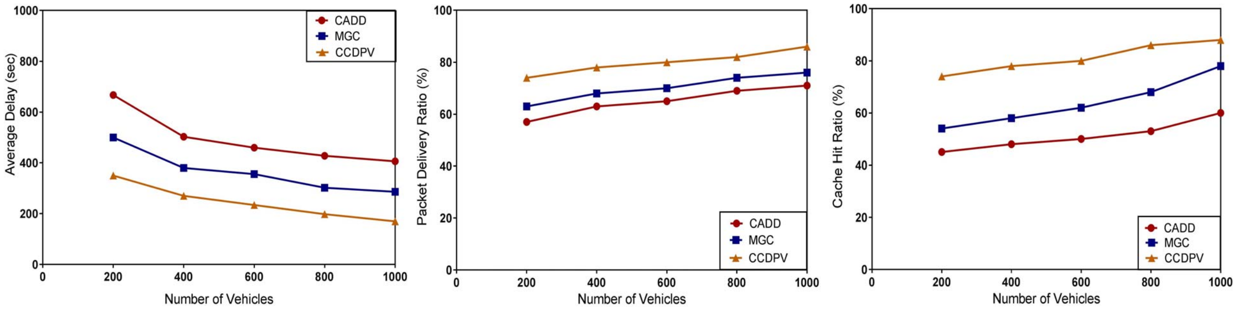
If $v_f$ is a requesting vehicle, it performs the aforementioned steps (b) and (c). In case of a cache miss, it sets the current data provider to the data center and applies steps (f)-(h).

### B. *CCDPV Cache Placement at Moving Vehicles*

This procedure is triggered by forwarding vehicles along the data delivery path. In CCDPV, cache placement decisions are made via collaboration between the nodes along the data delivery path, as well as their neighbors. Our goal is to increase data diversity among vehicles heading in the same direction. This is to enable data diffusion in different road segments. A field, denoted $h_{prev}$, indicating $r_{next}$ of the last moving vehicle encountered along the delivery path, which has the data in its cache, is added to the data packet. As the packet propagates en route to the requester, a decision is made to either cache the data at the forwarding vehicle or at one of its neighbors. If the data is already cached at any of its neighbors, the vehicle caches the data if it is heading to a different road segment than the caching neighbor. If they are heading to the same road segment, the vehicle selects the neighboring vehicle that has the largest cache space among those heading to a different road. If the data is not already cached at any node, the same cache admission policy is applied but relative to $h_{prev}$. A Least Frequently Used (LFU) replacement policy is employed.

### C. *CCDPV Cache Placement at Parked Vehicles*

When a CH receives a data packet to be forwarded, it makes a cache placement decision to determine whether or not to cache the data at parked vehicles in its cluster or in the same road segment. Traffic density at the road segment is considered. The more populated the road segment is, the higher the chance for the cached data to be hit. Real-time estimation of the traffic density at a road segment can be



(a) Average delay.

(b) Packet delivery ratio.

(c) Cache hit ratio.

Fig. 2. Performance results of CCDPV, CADD, and Modified-GroupCaching, over varying vehicular densities.

calculated based on the number of beacon messages received by the CH from moving vehicles during a period of time. When a CH receives a data packet, it checks if any of the parked vehicles at its road segment has the data in its cache. If not, it calculates the traffic density at its road segment, $r_k$, relative to the maximum density at all roads. This calculated value is denoted $\theta'_k$. The CH relies on traffic statistics, accessible via the navigation service, to estimate the traffic density at all other road segments. If $\theta'_k$ is less than a certain threshold, $\sigma$, no caching occurs. Otherwise, the packet is cached at the parked vehicle that has the highest cache space at the road segment. A LFU replacement policy is employed.

## IV. PERFORMANCE EVALUATION

In this section, the performance of CCDPV is evaluated compared to CADD [7]. This is since CADD is a VANET-based caching scheme that implicitly exhibits a somewhat cooperative caching behavior. Also, to evaluate the performance of our proposed cache discovery scheme, we explicitly compare it to the cooperative cache discovery scheme executed in GroupCaching [8]. To do so, we modify GroupCaching by implementing its own underlying cache discovery procedure, along with our cache placement scheme. Note that GroupCaching employs a commonly used tracking-based cache discovery scheme in MANETs, which does not yield excessive amount of overhead for highly dynamic networks [4]. Thus, it is applicable in VANETs [4]. The comparison is in terms of the following performance metrics: 1) the average delay starting from the time an interest packet is issued till a response is received, 2) the packet delivery ratio, which is the ratio of data packets successfully received by requesters to the total number of data packets generated, and 3) the cache hit ratio, which is the ratio of the number of data packets received from a caching node to the total number of data packets delivered.

### A. Simulation Setup

We use the NS-3 network simulator [14] to implement CCDPV, CADD, and Modified-GroupCaching (MGC). A $6\times6$ road grid topography that is composed of 120 road segments, is created. We use the SUMO traffic simulator [15] to promote the generation of realistic mobility traces, with the maximum speed of vehicles set to 40 km/h. We test the performance of CCDPV under varying vehicular densities, including low, medium, and high, in the range of 200-1000 vehicles. Simulations are conducted throughout a total simulation period of 2000 seconds each. The IEEE 802.11p WAVE standard is employed and the communication range is set to 150 meters. The beacon interval is assigned a value of 0.5 seconds. The interest generation is uniformly distributed among 20 requesters, with a 75-second generation rate. The interest of the requesters is directed towards 1-4 public figures on social media, each of which generates 4 new posts every 15 minutes. The number of parked vehicles is 250 and they are uniformly distributed among the road segments. These vehicles reside in their parking spaces throughout the entire simulation period. The amount of cache capacity that vehicles are willing to allocate for caching, represents 30% of the available content that can be requested. The time step, $\Delta$, as well as the $\alpha$ and $\beta$ weights, used for calculating the ranks of vehicles in the TPP, are set to 4 minutes, 0.2, and 0.8, respectively. The traffic density threshold, $\sigma$, is set to 0.5.

### B. Simulation Results and Analysis

First, we evaluate CCDPV compared to CADD and MGC in terms of average delay over varying vehicular densities. As demonstrated in Figure 2(a), CCDPV significantly improves the delay compared to both schemes. This can be attributed to two reasons. First, the increased data availability induced by the cooperative caching decisions at both parked and moving vehicles in CCDPV. Such decisions lead to increased data diversity and more efficient utilization of the vehicles' storage resources. This improves cache hits, and thus allows requesters to avoid procuring the data from the far-away data center. This is in contrast to the lack of such data diversity and informed caching decisions in CADD, which does not employ an explicit cooperative caching scheme. This explains the fact that CADD renders the highest amount of delay among the three schemes. Second, the tracking-based cache discovery procedure employed in CCDPV expands the search space and dynamically locates a closer replica to the requester. This increases the possibility of acquiring the data from nearby caching nodes, which further improves the delay. This is as opposed to the restricted search space in CADD and MGC. Note that as the number of moving vehicles increases, road segments get more congested and thus vehicles tend to slow down. This increases the validity time of the last encounter information registered in the vehicles TPP in CCDPV. It also improves the accuracy of the estimated positions of caching nodes. Hence, it makes it easier to locate closer caching nodes to the requester, which further improves the delay.

Second, we perform the same experiment in terms of the packet delivery ratio. As depicted in Figure 2(b), CCDPV significantly improves the packet delivery ratio compared to CADD and MGC. This is attributed to the tracking procedure applied in CCDPV, as a part of the cache discovery process, to track the requester's position. This is achieved through a rather stable tracking service leveraged by the static nature of parked vehicles. As a result, the risk of dropping the data packet, due to the inability to locate the requester, is significantly reduced. Such a risk typically increases as the requester moves far-away from its request initiation position. On the other hand, neither CADD nor MGC employ any tracking procedure to track the requester. As previously mentioned, the higher the vehicular density, the slower the rate of vehicles movement. This makes tracking the requester much easier and more successful. Thus, the packet delivery ratio increases as the number of vehicles increases. In addition, the significantly improved delay in CCDPV reduces the risk of the requester moving too far away before the arrival of the data. This further increases the number of successfully delivered data packets.

Third, we conduct the same comparison to assess the cache hit ratio. As depicted in Figure 2(c), CCDPV yields significantly increased cache hit ratio compared to both schemes. This can be attributed to the same reasons illustrated above. In addition, CCDPV caches diverse data at moving vehicles while taking their direction of movement into consideration. This helps diffuse the cached data into the network. Meanwhile, it also exploits the static nature of parked vehicles to provide the cached data with a stable residence that can be easily reached. This further improves cache hits. In addition, the stable tracking service provided by parked vehicles to track the location of caching nodes extends the lifetime of the cached content information. This helps sustain the expanded search space for a longer time, which makes it easier to locate caching nodes. In contrast, in MGC, once two neighboring nodes move out of range, their cached content information gets immediately nullified. On the other hand, CADD yields the least cache hit ratio among the three schemes. This is attributed to the aforementioned reasons, as well as the fact that the latter uses a server-based cache discovery scheme, which does not involve any cached content information exchange. Rather, it relies on opportunistic encounter with caching nodes en-route to the server. This further limits the search space compared to CCDPV and MGC, which reduces cache hits.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed the CCDPV scheme, which aims at improving the quality of VANET-based Internet service. This is to enable the use of VANETs as an expedient alternative to cellular networks, hence alleviating cellular costs for social media users. To do so, CCDPV employs a tracking-based cache discovery scheme to dynamically navigate requests towards closer caching nodes to the requester. CCDPV relies on beacon messages, as well as the mobility and stability of moving and parked vehicles, respectively, to diffuse cached content information into the network. It also exploits parked vehicles to keep the cached content information alive at road segments for subsequent use, as well as to provide a rather stable tracking service. This helps expand the search space without having to send extra messages. In addition, CCDPV caches diverse data at parked vehicles while taking traffic density into consideration. It also caches data at moving vehicles based on their direction of movement, to further increase data diversity and diffusion. Simulation results have shown that CCDPV achieves significant improvements in terms of delay, packet delivery ratio, and cache hit ratio, compared to a caching scheme in VANETs that exhibits an implicit collaborative behavior, as well as to a tracking-based cache discovery scheme in MANETs. In our future work, we will incorporate a mobility prediction module to generate real-time estimations of the vehicles travel time on road segments.

## REFERENCES

[1] *"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 2018, April 20).

[2] *"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 2018, April 20).

[3] 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.

[4] S. Glass, I. Mahgoub, and M. Rathod, *"Leveraging MANET-Based Cooperative Cache Discovery Techniques in VANETs: A Survey and Analysis"*, IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp.2640–2661, 2017.

[5] A. Ioannou, and S. Weber, *"A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking"*, IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp.2847–2886, 2016.

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

[7] S. Abdelhamid, H. S. Hassanein, G. Takahara, and H. Farahat, *"Caching-Assisted Access for Vehicular Resources"*, in the IEEE Conference on Local Computer Networks (LCN '14), pp. 28–36, Sept. 2014.

[8] S. N. Trambadiya, P. A. Ghosh, and J. N. Rathod, *"Group Caching: A Novel Cooperative Caching Scheme for Mobile Ad Hoc Networks"*, Int. J. Eng. Res. Develop., vol. 6, no. 11, pp. 23–30, 2013.

[9] N. E. Majd, S. Misra, and R. Tourani, *"Split-Cache: A Holistic Caching Framework for Improved Network Performance in Wireless Ad Hoc Networks"*, in Proc. IEEE Glob. Commun. Conf. (GLOBECOM), Austin, TX, USA, pp. 137–142, Dec. 2014.

[10] M. Varvello et al., *"Scale: A Content-Centric MANET"*, in Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), pp. 29–30, 2013.

[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] G. Deng, L. Wang, F. Li, and R. Li, *"Distributed Probabilistic Caching Strategy in VANETs through Named Data Networking"*, in Proc. of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 314–319, 2016.

[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] The NS-3 Network Simulator. [Online]. Available: https://www.nsnam.org/ (Accessed 2018, April 20).

[15] SUMO (Simulation of Urban Mobility). [Online]. Available:http://sumo-sim.org/ (Accessed 2018, April 20).