Probabilistic Cooperative Caching in VANETs 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*—Social media traffic constitutes the highest percentage of Internet traffic, which is mostly facilitated by mobile devices. This leads to high cellular costs incurred by mobile users. To reduce these costs, we strive to enable social media users to rely more on vehicular rather than cellular networks for content access. However, this can be hindered by the high delay and low packet delivery ratio often associated with accessing data from distant content providers in vehicular networks. Thus, to bring the data closer to the requester, we propose the Probabilistic Cooperative Caching at Moving and Parked Vehicles (PCCMPV) scheme. In PCCMPV, we exploit the static and mobile nature of parked and moving vehicles, respectively, to dynamically populate valuable road segments with diverse cached data. To do so, we dynamically assign a probability of caching to nodes along the data delivery path to assess their importance as caching nodes. For parked vehicles, such a probability relies primarily on the traffic density of the corresponding road segment, as well as its closeness centrality, and remoteness from the nearest data holder. PCCMPV provides an implicit form of off-path caching by assessing the trajectory of moving vehicles encountered along the data delivery path to calculate their probability of caching. Performance evaluation of PCCMPV demonstrates significant improvements in terms of delay, packet delivery ratio, and cache hit ratio compared to other caching schemes in vehicular networks.**

***Index Terms*—VANETs, Caching, Social Media.**

I. INTRODUCTION

The proliferation of social networking has become a global phenomenon for Internet users recently. In 2017, social media users represented 71% of Internet users [1]. Such a high usage is expected to further increase in the future, with an estimated growth in social media users reaching 2.9 billion in 2020 [1]. More than 60% of this wide social media usage is largely facilitated by mobile devices [2]. This causes mobile users to incur high cellular costs, particularly in outdoor areas that are not equipped with Wi-Fi connection. One solution to alleviate such costs is to allow users to rely more on other types of free/less expensive networks for data access. With their ubiquitous availability, Vehicular Ad Hoc Networks (VANETs) are major candidates to consider.

VANETs have transpired as a communication paradigm that endeavors to enable communication among vehicles on the road. They act as a vital enabling technology for a wide range of applications in Intelligent Transportation Systems (ITS), including infotainment services, such as Internet access [3]. However, the quality of service provided in these applications can be affected by the challenging issues associated with data access in such networks.

Data access in VANETs can be achieved via communication between vehicles and roadside access points, typically referred to as Road Side Units (RSUs) [3]. However, wide deployment of RSUs requires large investments, and thus they may not be densely deployed [3]. Hence, the nearest available RSU might be located far-away from the requester. Accordingly, Internet users in VANETs often rely on Vehicle-to-Vehicle (V2V) communication to reach the nearest RSU [3]. V2V communications targeting distant content providers are often associated with high delay and low packet delivery ratio [3]. This is attributed to the highly dynamic topology and intermittent connectivity of VANETs [3]. In addition, Internet access in VANETs typically relies on a request-response data access model [3]. That is, a request is directed towards the data center, accessed through a RSU, and a reply is sent back to the requester. This could further reduce the quality of service.

In order to enable mobile users to regard VANETs as a convenient alternative to cellular networks for content access, it is important to improve the quality of VANET-based Internet service. Thus, in this paper, we strive to bring the data closer to the requester by proposing the Probabilistic Cooperative Caching at Moving and Parked Vehicles (PCCMPV) scheme. Cooperative caching has been widely used as a useful technique for improving the performance of data access in various network paradigms, including Mobile Ad Hoc Networks (MANETs) [4] and Information-Centric Networks (ICNs) [5]. In cooperative caching, caching decisions are made in a collaborative manner and/or the nodes form a cooperative cache by sharing cached data [4]. To do so, nodes exchange information pertaining to their cached data. Such a collaboration helps make more informed caching decisions, and thus reduces redundancy and wasted cache space, which improves cache hits. This is particularly important in case of huge amount of contents, such as in social media.

Cooperative caching has been proven to help bring the data closer to the requester, generate more content diversity, and make more efficient utilization of the nodes' storage resources [4] [5]. However, despite its demonstrated leverage in MANETs and ICNs, cooperative caching within VANETs has been mostly overlooked. This is due to the highly dynamic nature of vehicles. Such a dynamic nature can lead to unstable

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

caching decisions. For example, caching decisions made by neighbors based on their information exchange, can get rapidly revoked as vehicles move out of range.

In order to take advantage of the benefits of cooperative caching, we counteract the aforementioned problems by exploiting the static nature of parked vehicles. We do so to provide a more stable residence for both the cached replicas and the received cached content information about other vehicles. This enables the extension of the cooperation range between nodes, and thus making more informed caching decisions. Such an extension is facilitated by sending different nodes' cached content information from parked to moving vehicles, and vice versa, via beacon messages. As opposed to RSUs, it has been shown that roadside parked vehicles are natural infrastructures that exist in sheer numbers [7]. Thus, they can provide significant and cost-effective storage resources [7]. A study that observed on-street parking spaces in Ann Arbor city in the US has illustrated that their occupancy ratio can reach 93% and 80% during on and off-peaks, respectively [7].

To the best of our knowledge, PCCMPV is the first cooperative caching scheme in VANETs that caches data at both parked and moving vehicles. In PCCMPV, we populate valuable road segments with diverse cached data to increase cache hits. This is to allow data to be acquired from nearby caching nodes rather than the far-away data center. We do so by dynamically assigning a probability of caching to vehicles along the data delivery path. Such a probability evaluates the importance of vehicles as caching nodes. For parked vehicles, this probability is calculated based on the traffic density of the corresponding road segment, as well as its closeness centrality, and remoteness from the closest data holder. Most existing cooperative caching techniques tend to adopt on-path caching techniques only [4] [5]. In PCCMPV, we exploit the trajectory of moving vehicles to also apply an implicit form of off-path caching.

We evaluate the performance of PCCMPV using the NS-3 simulator. We compare it to two caching schemes in VANETs, the Caching-Assisted Data Delivery (CADD) scheme [8], and the Distributed Probabilistic Caching (DPC) scheme [6]. This is since CADD implicitly inherits some features of cooperative caching, and DPC is a non-cooperative caching scheme that has been shown to outperform other caching schemes in VANETs. Simulation results show that PCCMPV outperforms CADD and DPC, in terms of access delay, packet delivery ratio, and cache-hit ratio.

The rest of the paper is organized as follows. Section II presents an overview of some related work. Section III discusses the proposed scheme (PCCMPV). Section IV provides the scheme's performance evaluation and simulation results. Section V presents our conclusions and future work.

II. RELATED WORK

A. Cooperative Caching in MANETs and ICNs

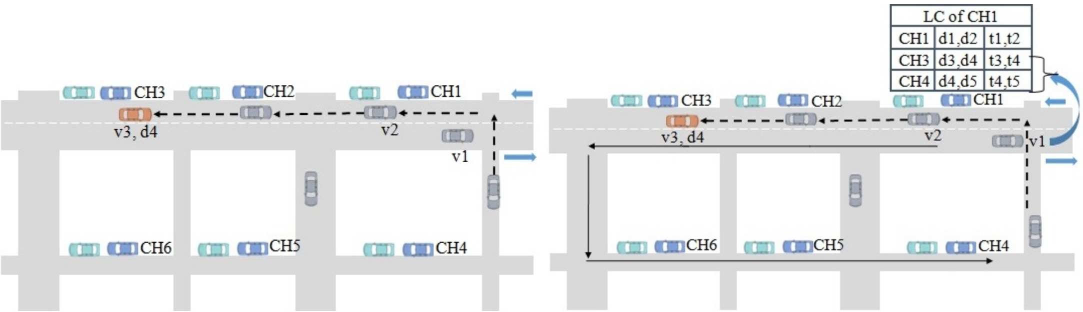
Several cooperative caching schemes have been proposed in MANETs and ICNs [4] [5]. In [9], a global cooperative caching scheme is applied, where each node sends its cached content information to all other nodes in the network, and an optimization technique is implemented for caching. Global cooperative caching schemes typically induce huge communication overhead, particularly in highly dynamic networks [5]. In [10], a path-based local cooperative caching scheme is employed, where cooperation occurs between nodes along the delivery path only. The authors in [11] propose a neighborhood-based local cooperative caching scheme, where cooperation is restricted to neighboring nodes. The main drawback of path-based and neighborhood-based local cooperative caching schemes is that the cooperation range is limited [4] [5]. Such a limited range can cause caching decisions to get quickly invalidated in highly dynamic networks. We expand such a range by exploiting the mobility of vehicles, the static nature of parked vehicles, and the fact that beacon messages are periodically exchanged among neighboring nodes. Such factors are used to acquire and distribute cache information among a wider range of nodes.

B. Caching in VANETs

Only few caching schemes have been proposed in VANETs [6]. In [12], the authors propose a non-cooperative caching scheme, where all moving vehicles along the data forwarding path cache the data. In CADD [8], caching occurs only at static nodes, called Road Caching Spots (RCSs), deployed at intersections. Using an implicit form of collaboration between on-path RCSs, the RCS that receives the highest number of requests is dynamically selected for caching. Thus, CADD inherits some features of cooperative caching. In DPC [6], a non-cooperative caching scheme is performed at moving vehicles. DPC considers three factors; data popularity, the degree and betweenness centrality of the vehicle in the ego network, as well as the relative direction of movement between the data provider and consumer. DPC has been shown to improve network performance compared to the scheme in [12]. Recently, some schemes have used roadside parked vehicles to cache large-sized contents [13]. These schemes divide the data into smaller chunks and use the sequential nature of parked vehicles to store them, so as to be later acquired by moving vehicles as they pass by. However, such schemes are more concerned with content downloading, rather than caching itself [13]. Most existing caching schemes in VANETs can be exposed to increased data redundancy, and thus wasted cache space and reduced cache hits [6]. This is due to the lack of explicit cooperation and cached content information exchange between nodes. In PCCMPV, we perform explicit cooperative caching at both parked and moving vehicles. As opposed to most existing techniques, along with on-path caching, we also apply an implicit form of off-path caching.

III. PROBABILISTIC COOPERATIVE CACHING AT MOVING AND PARKED VEHICLES (PCCMPV)

As previously mentioned, our objective is to improve the quality of Internet service in VANETs, to enable mobile users to depend more on vehicular rather than cellular networks for social media access. We do so by bringing the data closer to



LC of CH1		
CH1	d1,d2	t1,t2
CH3	d3,d4	t3,t4
CH4	d4,d5	t4,t5

(a) Non-cooperative caching.

(b) Cooperative caching using PCCMPV.

Fig. 1. An Illustrative Scenario. (a) In a non-cooperative caching scenario, assume that CH_1 receives a data packet d_4 on its delivery path to the requester v_3 (dotted line). Assume that both CH_3 and CH_4 already have d_4 cached in their own clusters. Unaware of that, CH_1 caches the data. This reduces data diversity due to the proximity between CH_1 and CH_3, as well as between CH_1 and CH_4. (b) Consider a similar scenario using PCCMPV. Assume that the vehicle, v_1, has previously passed by CH_3 and CH_4 and thus it already has information about the cached data at both clusters. Such information is maintained in its own LC. Afterwards, when v_1 enters the road where CH_1 resides, both CH_1 and v_1 exchange their LCs via beacon messages. Accordingly, CH_1 now knows the cached data in CH_3 and CH_4, including d_4. CH_1 then receives d_4, intended to the requester v_3. Thus, it makes an informed decision and does not cache the data. Meanwhile, CH_1 and v_2 have exchanged their LCs. Thus, v_2 now also has information about the cached data in CH_3 and CH_4. Hence, when v_2 receives d_4, it checks its own trajectory (solid line), and determines that it passes by both CH_3 and CH_4 (data holders). Also, it assesses the value of the other roads along its trajectory based on a number of metrics, including their proximity to CH_3 and CH_4, and does not cache d_4.

the requester through cooperative caching. In order to make informed caching decisions, we rely on information exchange between parked and moving vehicles via beacon messages.

We assume that requesting vehicles are interested in social media platforms that do not have large-sized contents, such as Instagram. We focus on the accounts of public figures, such as those of actors, singers, politicians, etc., since caching such data would be beneficial, as they attract many users. The posting frequency of a given public figure is estimated by the data center. Such information represents the Time to Live (TTL) of the data. TTL is the estimated time before a new post is posted by a public figure, rendering the earlier data as old. The data center associates the data packet with its TTL when it gets generated. Based on this value, the expiry time of the data is determined. Vehicles are willing to dedicate a certain amount of their storage capacity in exchange for some incentives. For simplicity, we assume that there are parked vehicles at each road segment. Vehicles are equipped with a navigation service that provides them with traffic statistics, such as the traffic density and average speed of vehicles per road segment. Changes in this traffic statistics are triggered by major traffic updates that occur at various traffic checkpoints throughout the day (example: morning, afternoon, rush hour, and evening). Moving vehicles know their own trajectories, and contents are uniquely named, such as in named data networking [6].

Parked vehicles at each road segment are organized into clusters. The way clustering occurs is similar to that in [13]. A cluster head (CH) is selected to be responsible for making caching decisions at all parked vehicles in its cluster, to ensure data diversity. The CH is responsible for exchanging cached content information with neighboring vehicles via beacon messages. Thus, the closest parked vehicle to the entry of the road segment is selected as the CH. This is to ensure that the information are exchanged between moving and parked vehicles as soon as the former enter the road. In case of a two-way road segment whose length is greater than the communication range, two CHs are selected, one at each end. Each CH maintains two data structures; the List of Neighbors (LN) and List of Clusters (LC). The LN of a CH contains metadata about the cached data in each parked vehicle in its own cluster. The LC of a CH contains metadata about the cached data in other clusters. Each entry in LN contains: the name of the data, the ID of the data holder, and the data expiry time. Each entry in LC contains: the ID of the CH, the names of the data cached in its cluster, as well as their expiry time. Moving vehicles also maintain their own LNs and LCs. The LNs of moving vehicles contain metadata about the cached data in their neighboring vehicles. Entries are removed from LCs and LNs once the data expires, or vehicles move out of range in case of LNs. Moving vehicles and CHs exchange their LCs via beacon messages. Note that before a CH sends its LC, it adds an entry of its own cluster by merging such information from its LN. The leverage of such information exchange is depicted in Figure 1. LNs are mainly used for cooperative cache discovery, which is the process of discovering where the data is cached [4]. It is the first step that needs to be done when a request is issued. We have adopted the discovery scheme in [11], where the cache of neighboring nodes is consulted in case of a local cache miss. In the next subsections, we discuss the operation of PCCMPV at both parked and moving vehicles.

A. *PCCMPV at Parked Vehicles*

This procedure is triggered when a CH receives a data packet to be forwarded. Note that the packet forwarding procedure works as follows: 1) If the packet source or a forwarding node has a CH in its neighborhood that is closer to the destination than itself, it forwards the packet to the latter. This is to enable more CHs to receive the data, so that caching decisions can be made. 2) Otherwise, the node anchors the packet towards the destination using greedy forwarding. In greedy forwarding, the packet is sent to the closest neighboring node to the destination. Aside from CHs, only moving vehicles are selected for forwarding. However, if the vehicle has no moving vehicles among its neighbors, it uses parked vehicles for forwarding. When a parked vehicle receives a data packet, it checks the caching and forwarding flags in the data packet. These flags are included to indicate whether the received data should be cached only, forwarded only, or both. Initially, only the forwarding flag is set. If both flags are set, then the parked vehicle caches a copy of the received data and forwards the original packet by applying the aforementioned forwarding procedure.

When a data packet is received by a CH, it determines whether or not to cache the data at its parking cluster. To do so, it calculates its own probability of caching. Such a probability represents the importance of the road segment at which the parking cluster resides. This importance is based on three metrics; the traffic density of the road segment, its closeness centrality, as well as its remoteness from the nearest data holder. The traffic density is used to reflect that the more congested the road segment is, the greater the possibility for the cached data to be hit. The closeness centrality is used to determine whether the cached content would be closely located, and thus rapidly accessed by requesters at other highly populated road segments. This is considering that the latter do not have the data cached in their corresponding parking clusters. We focus on such road segments due to the high chance of requests occurring at or passing by them. The remoteness from the nearest data holder, including the data center point of contact, is used to ensure data diversity. To calculate the probability of caching, the CH calculates three scores representing each of these metrics. Each score is a value in the range $[0, 1]$. Such a score represents the normalized value of the corresponding metric, calculated relative to that of all road segments in R, where R is the set of all road segments in the road network. The traffic density score of a road segment, r_i, at time, t, is denoted $\theta_i^{t,norm}$, its closeness centrality score is denoted $\psi_{i,d}^{t,norm}$, and its remoteness score is denoted $\chi_{i,d}^{t,norm}$. Note that in each data packet, we include a field in its header, referred to as the caching status field, which specifies the last parking cluster along the packet's delivery path that cached the data. Initially, such field is empty.

Algorithm 1 : PCCMPV at Parked Vehicles

```
1: Input:
2: Set of All Road Segments $R$
3: Reply Packet $d$
4: $\theta_i^{t_{prev},norm}$, $\psi_{i,d}^{t_{prev},norm}$, $\chi_{i,d}^{t_{prev},norm}$    //previous scores
5:
6: Cache_Placement(d, $CH_i$)
7: Begin
8:   if no traffic or cache status update since $t_{prev}$ then
9:       $\theta_i^{t,norm}$=$\theta_i^{t_{prev},norm}$
10:      $\psi_{i,d}^{t,norm}$=$\psi_{i,d}^{t_{prev},norm}$
11:      $\chi_{i,d}^{t,norm}$=$\chi_{i,d}^{t_{prev},norm}$
12:  else
13:      calculate $\theta_i^{t,norm}$    //Eq.1
14:      check LC and determine $M$    // $M$=set of roads caching $d$
15:      for all $r \in R$ do
16:          if $r \notin M$ and $\theta_r^{t,norm} >$ avg then
17:              add $r$ to $S$
18:      construct Graph $G(V, E)$    //$V$=$S\cup$ $r_i$
19:      for all $j \in V | j \neq i$ do
20:          $dist_{i,j}$+=$dist_{i,j}$
21:      $\psi_{i,d}^t$=$1/dist_{i,j}$    //Eq.2
22:      calculate $\psi_{i,d}^{t,norm}$    //Eq.3
23:      calculate $\chi_{i,d}^{t,norm}$    //Eq.4
24:  calculate $P_{r_i,d}^t$    //Eq.5
25:  if $P_{r_i,d}^t \geq th_c$ then
26:      cache $d$ at parked vehicle with max cache space in the cluster
27: End
```

Once a cluster head, CH_i, receives a data packet, d, it checks the caching status field in the packet and updates its LC accordingly. If d is not already cached in the cluster, the following procedure, illustrated in Algorithm 1, is then performed: (a) CH_i checks if a major traffic update has occurred or a change in the cache status of the road segments in R has become known to it since the last time the three scores of the road segment r_i, $\theta_i^{t,norm}$, $\psi_{i,d}^{t,norm}$, and $\chi_{i,d}^{t,norm}$, were calculated (line 8). If not, then the previously calculated values of the scores can be used (lines 9-11). Otherwise, CH_i recalculates them. It calculates $\theta_i^{t,norm}$ using Eq. 1, where θ_i^t is the traffic density of r_i (lines 12 & 13). It then calculates

$$\theta_i^{t,norm} = \frac{\theta_i^t - \min_{k\in R}\theta_k^t}{\max_{k\in R}\theta_k^t - \min_{k\in R}\theta_k^t} \tag{1}$$

$\psi_{i,d}^{t,norm}$. To do so, the value of the closeness centrality of r_i, before being normalized, denoted $\psi_{i,d}^t$, is first calculated as follows: CH_i checks its LC to determine, to the best of its knowledge, the set of road segments, denoted M, that already have the data cached (line 14). The remaining roads in R are then filtered based on their traffic density. Thus, CH_i creates a set, denoted S, of road segments that do not have the data cached, and whose normalized traffic density is above average (lines 15-17). CH_i then represents its accessibility to every other road segment (i.e. parking cluster) in S via a graph, $G(V, E)$, of $|V|$ vertices and $|E|$ edges (line 18). A weight is assigned to each edge e_{xy}, representing the shortest distance from r_x to r_y, measured in hop counts. The closeness centrality, $\psi_{i,d}^t$, of CH_i, given by Eq. 2, is thus defined as the inverse of the sum of the shortest-path distances between r_i and the remaining nodes in $G(V, E)$, where the shortest-path distance from r_i to r_j, is denoted $dist_{i,j}$ (lines 19-21). The cluster head, CH_i, calculates the normalized value of its closeness centrality, $\psi_{i,d}^{t,norm}$ using Eq. 3 (line 22). It then calculates $\chi_{i,d}^{t,norm}$ using Eq. 4 (line 23). Note that $\chi_{i,d}^t$=$min_{q\in M'} dist_{i,q}$, is the remoteness of r_i from the nearest

$$\psi_{i,d}^t = \frac{1}{\sum_{j\in V, i\neq j} dist_{i,j}} \tag{2}$$

$$\psi_{i,d}^{t,norm} = \frac{\psi_{i,d}^t - \min_{k\in R}\psi_{k,d}^t}{\max_{k\in R}\psi_{k,d}^t - \min_{k\in R}\psi_{k,d}^t} \tag{3}$$

$$\chi_{i,d}^{t,norm} = \frac{\chi_{i,d}^{t} - \min_{k \in R} \chi_{k,d}^{t}}{\max_{k \in R} \chi_{k,d}^{t} - \min_{k \in R} \chi_{k,d}^{t}} \tag{4}$$

data holder, including the data center point of contact (i.e. $M^{'}$=M $\cup$ data center). (c) The probability of caching d at CH_i, represented by the probability of caching at the road segment r_i where it resides, denoted $P_{r_i,d}^{t}$, is calculated using Eq. 5, where ω_1, ω_2, and ω_3 are weighting factors in the range $(0, 1]$, such that $\Sigma_{x=1}^{3}\omega_x$= 1 (line 24). (d) If $P_{r_i,d}^{t}$ is less than a predetermined caching threshold, th_c, the data will not be cached. Otherwise, CH_i selects the parked vehicle that has the largest available cache space in its cluster to cache the data (lines 25 & 26). CH_i sends a copy of the cached data to the selected parked vehicle after setting the caching flag. It also updates the caching status field in the original data packet to indicate its own cluster. A Least Frequently Used (LFU) replacement policy is used if needed.

$$P_{r_i,d}^{t} = \omega_1 \theta_{i}^{t,norm} + \omega_2 \psi_{i,d}^{t,norm} + \omega_3 \chi_{i,d}^{t,norm} \tag{5}$$

B. PCCMPV at Moving Vehicles

This procedure is triggered when a moving vehicle, v_m, receives a data packet, d, to be forwarded. In order to determine whether or not to cache the data, the vehicle, v_m, calculates its own probability of caching. Such a probability is determined based on the value of each road segment along the remaining part of its trajectory during which d remains valid. The valid remaining trajectory of v_m is denoted T_m^{rem}. The value of a road segment, r_j^m, along T_m^{rem}, depends on three metrics: 1) its stand-alone importance, 2) the estimated period of time that v_m will spend at the road segment, r_j^m, and 3) the possibility that the data packet will encounter r_j^m on its delivery path. The first metric is represented by the probability of caching the data, d, in r_j at time ts_{jm}, denoted $P_{r_j,d}^{ts_{jm}}$, which is calculated based on Eq. 5, where ts_{jm} is the time of arrival of vehicle v_m at r_j. The second metric is used to reflect that the longer the time v_m spends at r_j^m, the better. This is because the cached replica would then have a longer residency at the road segment, and can thus serve more users. Such a period represents the travel time on r_j^m starting from the time, ts_{jm}, denoted $tr_j^{ts_{jm}}$. This travel time, $tr_j^{ts_{jm}}$, is given by Eq. 6, where L_j is the length of r_j, and $a_j^{ts_{jm}}$ is the estimated average speed of vehicles in r_j at time ts_{jm}. The third metric is used to indicate that the less probable it is that d will pass by r_j^m, the better, since caching d at v_m would then give the road segment another chance of receiving and holding the data. This is considering that if a data packet does not reach r_j^m on its way to the destination, the latter will not have a chance to cache it. Such a possibility is calculated based on the distance between the road segment and the packet destination. The further the road segment, the less the chance of it receiving the packet. This is since the data packet is typically forwarded towards the destination. The aforementioned metrics are calculated in the form of three scores in the range $[0, 1]$. Such scores represent the normalized values of each metric, calculated for the road segment r_j^m, relative to all road segments in T_m^{rem}. The importance score of r_j^m, its travel time score, and distance score, are denoted $\delta_{j,d}^{ts_{jm}}$, $\tau_j^{ts_{jm}}$ and $\phi_{j,d}^{ts_{jm}}$, respectively. Based on these three scores, the probability of caching the data at v_m due to the individual value of r_j^m, denoted $P_{v_m,j,d}^{ts_{jm}}$, is determined. Note that the value of a road segment that already has the data cached in its parking cluster is set to zero. This is provided that the existing replica does not expire while v_m is on the road segment. This is since if that occurred, a cached replica in the vehicle would be useful for r_j^m, as it would make up for the expired one.

Algorithm 2 : PCCMPV at Moving Vehicles

```
 1: Input:
 2: TTL of the Data, $TTL_d$
 3: Reply Packet $d$
 4:
 5: Cache_Placement($d$, $v_m$)
 6: Begin
 7:   determine $T_m^{ro}$        //original remaining trajectory of $v_m$
 8:   calculate $tt_{m,ro}$        //total travel time along trajectory $T_m^{ro}$
 9:   $t_{m,valid}^{d}$=$min(TTL, tt_{m,ro})$ //time during which $d$ is valid
10:   for all $r_k^m \in T_m^{ro}$ do
11:       $tt_{m,rem}$+=$L_k$/$a_j^{ts_{jm}}$        // travel time of trajectory $T_m^{rem}$
12:       if $tt_{m,rem} \leq t_{m,valid}$ then
13:           add $r_k$ to $T_m^{rem}$
14:       else
15:           break
16:   for all $r_j^m \in T_m^{rem}$ do
17:       get $ts_{jm}$ and $te_{jm}$        // using the travel time at each road
18:       check $LC$
19:       if $d$ is cached at $r_j^m$ then
20:           if $t_{expiry}^{d} \geq te_{jm}$ then
21:               $P_{v_m,j,d}^{ts_{jm}} = 0$
22:           else if $t_{expiry}^{d} < te_{jm}$ then
23:               $ts_{jm}$=$t_{expiry}^{d}$
24:       if $P_{v_m,j,d}^{ts_{jm}}$ has not been assigned then
25:           calculate $\delta_{j,d}^{ts_{jm}}$, $\tau_j^{ts_{jm}}$, $\phi_{j,d}^{ts_{jm}}$        // Eq.7, Eq.8, Eq.9
26:           calculate $P_{v_m,j,d}^{ts_{jm}}$        // Eq.10
27:       $P_{v_m,j,d}^{ts_{jm}}$+=$P_{v_m,j,d}^{ts_{jm}}$        // calculate sum
28:   Calculate $P_{v_m,d}^{t}$        // Eq.11
29:   if $P_{v_m,d}^{t} \geq th_c$ then
30:       cache $d$ at $v_m$
31: End
```

$$tr_j^{ts_{jm}} = \frac{L_j}{a_j^{ts_{jm}}} \tag{6}$$

Once a moving vehicle, v_m, receives a data packet, d, it checks the caching status flag in the data packet and updates its LC accordingly. If the data is not already cached in v_m, the following procedure, illustrated in Algorithm 2, is triggered: (a) v_m determines its T_m^{rem} (lines 7-15). To do so, it determines its original remaining trajectory, T_m^{ro}, which is the total remaining part of its trajectory, including that during which d would not be valid. It also, calculates its total estimated travel time along T_m^{ro}, denoted $tt_{m,ro}$. It then determines the lifetime the data would have in its cache, denoted $t^{d}_{m,valid}$, by calculating the minimum between TTL_d and $tt_{m,ro}$. For example, if the validity time of d, TTL_d, is 20 minutes and the total travel time of v_m along T^{ro}_m is 30 minutes, then d would be valid at v_m for 20 minutes only. Road segments in T^{ro}_m, are sequentially added to T^{rem}_m as long as their total travel time does not exceed $t^{d}_{m,valid}$. (b) For each road segment, $r^m_j \in T^{rem}_m$ (line 16), the following steps are performed by v_m: (1) Determine the estimated time of arrival and departure to and from r^m_j, (ts_{jm} and te_{jm}), respectively (line 17). (2) Check LC to determine if d is already cached at r^m_j and if its expiry time, $t_{exp,d}$, is greater than te_{jm}. If so, set $P^{ts_{jm}}_{v_m,j,d}$, to zero (lines 18-21). (3) If the data is already cached at the road segment but its expiry time is less than te_{jm}, set ts_{jm} to the replica's expiry time (lines 22 & 23). (4) If the case in (2) does not apply, do the following: (lines 24-26) Calculate the three scores, $\delta^{ts_{jm}}_{j,d}$, $\tau^{ts_{jm}}_{j}$ and $\phi^{ts_{jm}}_{j,d}$, using Eq. 7, Eq. 8, and Eq. 9, respectively. Note that the value of $P^{ts_{jm}}_{r_j,d}$ in Eq. 7, is calculated using Eq. 5. Calculate the probability of caching the data at v_m due to the individual value of r^m_j, $P^{ts_{jm}}_{v_m,j,d}$, using Eq. 10, where ω_4, ω_5, and ω_6 are weighting factors in the range $(0, 1]$, such that $\Sigma^{6}_{x=4}\omega_x$= 1. (c) Calculate the probability of caching d at v_m at time t, denoted $P^{t}_{v_m,d}$, by determining the average caching probability at the road segments along T^{rem}_m, given by Eq. 11, where n=$|T^{rem}_m|$ (lines 27 & 28). If $P^{t}_{v_m,d}$ is less than the caching threshold, th_c, the data will not be cached. Otherwise, v_m caches the data (lines 29 & 30). A LRU replacement policy is used if needed. In addition, in order to further increase cache hits, upon receiving d, v_m notifies its neighboring vehicles, via beacon messages, of the name of the data it has received. This is to enable each moving vehicle, v_b, within v_m's neighbors, to calculate its own probability of caching, $P^{t}_{v_b,d}$. Each vehicle then sends its $P^{t}_{v_b,d}$ to v_m. The vehicle v_m then sends a copy of the data, with the caching flag set, to the neighbor with the maximum probability of caching. This is provided that such a probability is not less than th_c.

$$\delta^{ts_{jm}}_{j,d} = \frac{P^{ts_{jm}}_{r_j,d} - \min_{g \in T^{rem}_m} P^{ts_{gm}}_{r_g,d}}{\max_{g \in T^{rem}_m} P^{ts_{gm}}_{r_g,d} - \min_{g \in T^{rem}_m} P^{ts_{gm}}_{r_g,d}} \tag{7}$$

$$\tau^{ts_{jm}}_{j} = \frac{tr^{ts_{jm}}_{j} - \min_{g \in T^{rem}_m} tr^{ts_{gm}}_{g}}{\max_{g \in T^{rem}_m} tr^{ts_{gm}}_{g} - \min_{g \in T^{rem}_m} tr^{ts_{gm}}_{g}} \tag{8}$$

$$\phi^{ts_{jm}}_{j,d} = \frac{dist^{m}_{j,dest} - \min_{g \in T^{rem}_m} dist^{m}_{g,dest}}{\max_{g \in T^{rem}_m} dist^{m}_{g,dest} - \min_{g \in T^{rem}_m} dist^{m}_{g,dest}} \tag{9}$$

$$P^{ts_{jm}}_{v_m,j,d} = \begin{cases} 0 & d \text{ is cached}, \\ & t_{exp,d} \geq te_{jm} \\ \omega_4 \delta^{ts_{jm}}_{j,d} + \omega_5 \tau^{ts_{jm}}_{j} + \omega_6 \phi^{ts_{jm}}_{j,d} & \text{Otherwise} \end{cases} \tag{10}$$

$$P^{t}_{v_m,d} = \frac{\sum_{k \in T^{rem}_m} P^{ts_{km}}_{v_m,k,d}}{n} \tag{11}$$

IV. PERFORMANCE EVALUATION

In this section, the performance of PCCMPV is evaluated compared to DPC [6] and CADD [8]. The performance metrics used are: 1) the average delay from the time an interest packet is sent till a data reply is received, 2) the packet delivery ratio, which is the ratio of reply packets that are successfully delivered to the total number of reply packets issued, and 3) the cache hit ratio, which is the ratio of the number of replies received from a caching node to the total number of received replies.

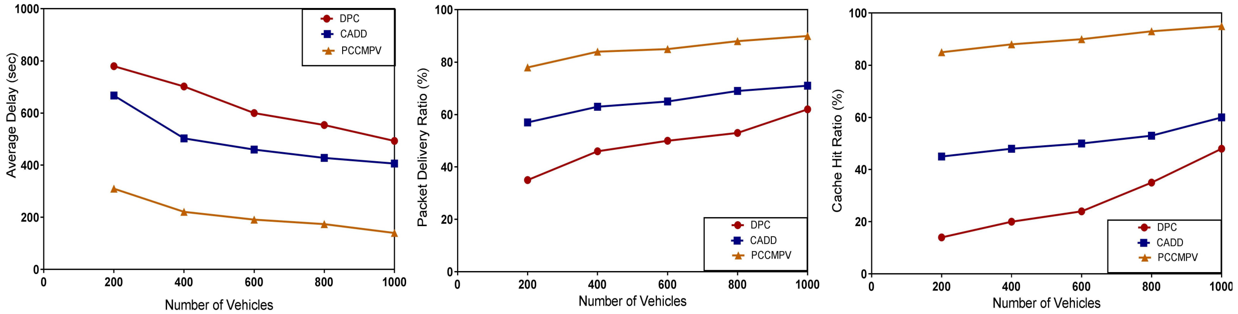
A. Simulation Setup

The NS-3 network simulator [14] is used to implement PCCMPV, CADD, and DPC. Simulations are conducted over a 6×6 road grid topography, consisting of 120 road segments. The SUMO traffic simulator [15] is used to foster the creation of realistic mobility traces, with the maximum vehicular speed assigned a value of 40 km/h. We vary the number of moving vehicles to test the performance of PCCMPV under different vehicular densities. Simulations are run for a period of 2000 seconds each. We use the IEEE 802.11p WAVE standard. The transmission range and beacon interval are set to 150 m and 0.5 seconds, respectively. The interest generation is uniformly distributed among 20 requesting vehicles with an injection rate of 75 seconds. Requesters are interested in 1-4 public figures on social media, each of which has 4 new posts every 15 minutes. In the simulations, 250 parked vehicles are uniformly distributed among road segments and they remain in their parking spaces for the entire simulation period. The amount of cache space that vehicles dedicate for caching, constitutes 30% of the content available for request. The weighting factors, ω_1-ω_3, used in Eq. 5 are set to 0.3, 0.4, and 0.3, respectively, while those used in Eq. 10, ω_4-ω_6, are set to 0.4, 0.3, and 0.3, respectively. The caching threshold th_c, is set to 0.4.

B. Simulation Results and Analysis

First, we compare PCCMPV, CADD, and DPC in terms of average delay over varying vehicular densities. As depicted in Figure 2(a), PCCMPV significantly outperforms both schemes. This can be attributed to the increased data availability, caused by the more informed caching decisions made in PCCMPV. Such decisions enable caching more diverse data at valuable nodes, which leads to more cache hits. This facilitates acquiring the data from nearby caching nodes rather than the far-away data center. As the number of vehicles increases, the amount of exchanged information increases, which improves caching decisions, and thus improves the delay. On the other hand, DPC incurs higher delay than CADD and PCCMPV, due to the lack of any form of collaboration between nodes. This is in contrast to the explicit and implicit collaboration between caching nodes in PCCMPV and CADD, respectively.

Second, we conduct the same comparison relative to the second metric. As shown in Figure 2(b), PCCMPV significantly improves the packet delivery ratio over CADD and DPC. This can be attributed to the highly improved delay in PCCMPV, which in turn reduces the chance of dropping



(a) Average delay. (b) Packet delivery ratio. (c) Cache hit ratio.

Fig. 2. Performance results of PCCMPV, CADD, and DPC, over varying vehicular densities.

packets due to the inability to find the requester. This typically occurs when the requester moves too far away from its request initiation position. The earlier the data arrives, the less the chance for this to occur.

Third, we apply the same experiment to evaluate the cache hit ratio. As depicted in Figure 2(c), PCCMPV yields much higher cache hits than CADD and DPC. This arises from the same reasons discussed above. In addition, PCCMPV makes caching decisions based on the importance of road segments, whether the ones at which parked vehicles reside or those that moving vehicles pass by along their trajectory. This provides more stable caching decisions, leveraged by parked vehicles. Such a leverage is further amplified by exploiting the mobility of vehicles to enable a form of off-path caching at valuable road segments. In particular, those that are less likely to receive the data. The higher the number of vehicles, the higher the number of road segments visited by caching nodes, and thus the higher the cache hits.

V. Conclusion and Future Work

In this paper, we have proposed the PCCMPV scheme to improve the quality of Internet service in VANETs. This is to make it a convenient alternative to cellular networks. Such an objective is driven by the need to reduce cellular costs for social media users. To do so, PCCMPV exploits the static and mobile nature of parked and moving vehicles, respectively, to acquire and distribute cached content information among nodes. Based on the exchanged information, PCCMPV pours diverse data into highly valuable road segments. This is done by dynamically assigning a probability of caching to vehicles along the data delivery path to evaluate their importance as caching nodes. PCCMPV further exploits the trajectory of moving vehicles to induce a form of off-path caching at road segments. Simulations have shown PCCMPV to yield significant improvements in terms of delay, packet delivery ratio, and cache hit ratio, compared to a caching scheme that has an implicit collaboration feature, as well as to a non-cooperative caching scheme, in VANETs. In our future work, we plan on applying mobility prediction techniques to estimate real-time information about the travel time of vehicles on road segments.

Acknowledgment

This research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number: STPGP 479248.

References

[1] *"Number of social media users worldwide from 2010 to 2020 (in billions)"*. [Online]. 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"*. [Online]. 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] 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.

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

[8] 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.

[9] V. Sourlas, P. Flegkas, L. Gkatzikis, and L. Tassiulas, *"Autonomic Cache Management in Information-Centric Networks"*, in Proc. 13th IEEE Netw. Oper. Manag. Symp. (NOMS), Maui, HI, USA, pp. 121–129, Apr. 2012.

[10] Z. Li and G. Simon, Time-shifted TV in Content Centric Networks: The Case for Cooperative In-Network Caching, in Proc. 53rd IEEE Int. Conf. Commun. (ICC), Kyoto, Japan, pp. 1–6, Jun. 2011.

[11] 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.

[12] 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.

[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)