

Caching and Forwarding Assistance for Vehicular Information Services with Mobile Requesters

Sherin Abdelhamid*, Sara A. Elsayed*, Tarun Kumar†, and Hossam S. Hassanein*

*School of Computing, Queen's University, Kingston, ON, Canada

{sherin, selsayed, hossam}@cs.queensu.ca

†Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada

13tkln@queensu.ca

Abstract—Smart vehicles have been capturing increasing attention as major providers of ubiquitous information services. In this paper, we propose a solution to enable expedited and cost-effective access of road information for vehicles as information requesters. The proposed caching-assisted data delivery with mobile requesters (CADD-MR) scheme employs caching on the data delivery path for handling later interests in similar data. CADD-MR depends on the use of light-weight road caching spots (RCSs) deployed at intersections for caching and heading-aware forwarding, and on vehicles as data carriers. To support vehicles as mobile information requesters, CADD-MR makes use of the RCSs for keeping track of the mobility registry of the requesting vehicles to ensure that data replies reach their corresponding mobile destinations, as long as they are still not expired. Performance evaluation of CADD-MR demonstrates significant improvements in the access cost, delay, and delivery ratio compared to a scheme that does not use RCSs.

Index Terms—Vehicular resources and data delivery, Caching, Mobile requesters.

I. INTRODUCTION

Smart vehicles have been considered major providers of ubiquitous information services that span across multiple domains and interests [1]. Thanks to the various in-vehicle sensors, smart vehicles have been major players in the public sensing domain [2]. Having a wide range of in-vehicle sensors along with abundant computing and communication resources, a vehicle can be considered a mobile sensor collecting data on the go, processing and sharing such data with interested parties for supporting a wide range of sensing-based information services [3]. Examples of such services include monitoring traffic and road conditions, sharing parking availability, and reporting crowd, pollution, and noise levels, to mention a few.

Two important factors should be taken into consideration while accessing vehicular resources to support sensing-based services. First, such an access paradigm involves an access cost challenge brought by the fact that whenever vehicular resources are accessed, resource owners must be rewarded. Second, a sensing interest needs to be forwarded towards a specific area of interest (AoI) every time data is required from that particular AoI, which leads to yet another challenge namely, the access delay.

In [4], we proposed the Caching-Assisted Data Delivery (CADD) scheme to resolve the access challenges highlighted above through caching replicas of the collected data on their

delivery path in light-weight entities called road caching spots (RCSs) deployed at each intersection. Such caching of the sensed data along the road facilitates responding to later interests in similar data without having to access vehicular resources again; thus, alleviating the access cost. Furthermore, access delay can be reduced through caching since it may happen to bring the data closer to the requesters through the stored replicas.

CADD is designed to support end-users' access to vehicular networks through static gateways acting as points of attachment of the interested users to the collecting vehicular network. Each user is registered and connected to a particular gateway through the Internet. All users' requests get injected into the vehicular network via the associated gateway. This gateway is also the entity through which the replies are delivered to the end-users. Thus, in CADD, vehicles act only as data carriers and generators and never as consumers. However, as discussed earlier, there is a vast range of applications that can benefit vehicles on roads as information consumers. Thus, a facility is needed to allow vehicles to work as interest initiators/requesters and enable them to receive road information on-demand. In this paper, we propose the Caching-Assisted Data Delivery with Mobile Requesters (CADD-MR) scheme targeting the objective of supporting vehicles as mobile requesters of road information.

In CADD-MR, in addition to their main functionalities as caching and forwarding assistants [4], the RCSs at intersections are used to assist in delivering reply packets to their mobile requesting vehicles. Since there is a high potential that a requesting vehicle leaves its interest-initiation position before it gets the corresponding reply, RCSs help in keeping track of the requesting vehicles' movement so that a reply packet can eventually reach the requesting vehicle even if it is moving.

In addition to its main objective of handling mobile requesters, CADD-MR targets solving access cost, delay, and delivery concerns. This solution is achieved through the use of a dynamic centrality-based caching mechanism (similar to CADD [4]), and by being heading-aware when forwarding data. A data-carrying vehicle keeps checking its heading and when it finds that it is going away from the packet destination, it seeks assistance from its neighboring RCS through offloading the packet to that RCS giving it a better delivery chance.

To evaluate the performance of the proposed CADD-MR

scheme, we modify the popular store-carry-and-forward (SCF) mechanism to support requesters' mobility. The modified SCF (M-SCF) is used for comparison purposes. We analyze the performance of CADD-MR compared to M-SCF using the NS-3 simulator. Simulation results show that CADD-MR significantly outperforms M-SCF in terms of the access delay, cost, and packet delivery ratio.

The remainder of this paper is organized as follows. Section II presents some related work in the areas of vehicular data gathering and data caching. In Section III, we present the proposed CADD-MR scheme detailing the operations of its involved entities. Section IV demonstrates the performance evaluation of CADD-MR and the simulation results. Finally, Section V concludes the paper.

II. RELATED WORK

A. Vehicle-based Data Gathering and Harvesting

Some schemes are proposed in the literature to provide access to vehicular sensing resources and utilize them for gathering on-road information on-demand through interest and reply dissemination. The Vehicular Information Transport Protocol (VITP) [9] is proposed to support location-aware services over vehicular networks. In VITP, vehicles send queries to retrieve information from areas of interest. Resolved replies are routed back from vehicles in these areas. Both the query and reply dissemination is handled by intermediate vehicles via multi-hop communication. VITP only takes care of specifying the syntax and semantics of the query and reply messages to be disseminated among vehicles. Its operation is independent of the underlying transmission and routing protocols.

In contrast to VITP, the Delay-Bounded Vehicular Data Gathering (DB-VDG) solution [10] manages routing the query and reply messages for supporting geographical vehicle-based data gathering services. In DB-VDG, vehicles act as mobile sensors and data relays. Queries are created, and replies are collected through a fixed base station. To minimize the communication overhead and provide a facility for multi-source data aggregation, vehicles in DB-VDG decide on either forwarding the received packets immediately or carrying them while moving. Such a forwarding/carrying decision is controlled by a delay bound.

The Extended Clustered Data Gathering Protocol (ECDGP) for vehicular networks [17] handles the complete data gathering process through involved techniques for process initiation and data collection, aggregation, and delivery. Also, it manages the medium access control through a reliable dynamic time division multiple access (RD-TDMA) technique. The protocol supports both delay-tolerant and real-time data gathering. ECDGP is a cluster-based protocol in which roads are divided into segments of two types: collection segment (CS) and silence segment (SS). To avoid collisions between adjacent segments, communication is allowed in CSs, while in SSs no communication is authorized. A vehicle is elected as a cluster-head (CH) in each CS and the RD-TDMA medium access technique is used to collect data. When CHs enter SSs, they run a spatial data aggregation technique to remove redundant

and undesired data. ECDGP builds on its predecessor, CDGP [18], that supports only delay-tolerant data gathering with no data aggregation involved.

Although the solutions mentioned above share with the proposed CADD-MR scheme its basic target of accessing the vehicular resources, they include data gathering mechanisms that do not support vehicles as mobile requesters. The VITP scheme does not involve forwarding intelligence, the DB-VDG scheme only supports forwarding packets from/to a fixed base station, and ECDGP assumes the requester is a fixed roadside unit (RSU).

B. Data Caching

Data caching has been an active topic of research in areas such as web caching [11] and information-centric network (ICN) caching [12]. Many caching mechanisms have been proposed with the most popular ones being the ubiquitous, probabilistic, and centrality-based caching [13].

In the area of caching in vehicular networks, only a few caching-based schemes have been proposed in the literature with the focus on using the mobile vehicles themselves as the caching entities. In [14], the VITP scheme [9] has been extended to support in-vehicle caching. While forwarding the reply packets to the requesting nodes, the intermediate nodes in the caching-enabled VITP scheme cache copies of such replies. When a VITP-enabled vehicle receives a query packet directed to an area of interest, the vehicle checks its local cache for the possibility of a cache hit. The vehicle forwards the query to a neighboring vehicle if it does not find a matching local replica. The Time-to-Live (TTL) value of the packets is used as the metric for cache replacement and management such that a packet is removed from the cache once its TTL reaches a pre-defined value.

Another example is the CRoWN framework [15] that introduces to the vehicular environment the concept of content-centric networking (CCN) [16], implemented on top of the IEEE 802.11p standard. In CRoWN, in-network caching and replication is exploited to achieve fast content retrieval in a communication environment driven by the contents instead of host addresses.

The above schemes have succeeded in achieving performance improvements compared to others that do not consider caching. However, with the very dynamic nature of vehicular networks, the opportunities of cache hits for replicas stored on mobile nodes that can be reached fortuitously are limited. Also, since vehicular networks are large-scale by nature, finding a vehicle with a replica of interest may require querying an extremely large number of vehicles. Therefore, a solution with static nodes for on-road caching is much more favored for the sake of increasing the opportunities of cache hits and minimizing the interest dissemination overhead.

III. CACHING-ASSISTED DATA DELIVERY WITH MOBILE REQUESTERS (CADD-MR)

The proposed caching-assisted data delivery with mobile requesters (CADD-MR) scheme targets three main performance-

improvement objectives in gathering and harvesting road information by mobile vehicles. It aims at minimizing the access cost and delay while improving the packet delivery ratio. To achieve such objectives, CADD-MR inherits the basic functionalities of its predecessor CADD scheme [4]. These include the utilization of light-weight RCSs deployed one at each intersection on traffic lights or electric poles. The RCSs work as on-road assistants to support caching and data forwarding. Replicas stored at RCSs can help in resolving later interests in similar data without having to request such data again from vehicles; hence, saving the access cost. Such replicas and cache hits also help in minimizing the access delay through introducing chances of having the data of interest stored as replicas closer to the requesters than the area of interest. CADD-MR inherits as well the heading-awareness feature of CADD. Instead of blindly forwarding the packets, the scheme takes into consideration the vehicles' headings for the purpose of improving the data delivery ratio and end-to-end delay.

As mentioned earlier, a main drawback of most of the vehicle-based data gathering and sharing schemes is that they do not support having the vehicles themselves as mobile requesters. To handle this issue, CADD-MR exploits the presence of the static RCSs at intersections to keep track of the route followed by a requesting vehicle since the interest has been initiated. As it moves, a requesting vehicle registers information about its identity, heading, and the sent request at each passing-by RCS. In such a way, RCSs are utilized for keeping track of and storing the mobility registry of the requesting vehicles. The registry is used to ensure that the replies follow their corresponding mobile destinations while moving and reach them eventually, as long as the replies are still not expired.

In the following subsections, we discuss the general operation of CADD-MR and the detailed functionalities of the requesting and forwarding vehicles in addition to the on-road RCSs.

A. General Operation of CADD-MR

In CADD-MR, an interest-initiator/requesting vehicle sends an interest packet to its closest neighboring vehicle. The interest packet is then disseminated in the network through vehicles and RCSs towards the area of interest (AoI) specified in the interest packet header. As in CADD, when an RCS receives an interest packet, it checks its cache for a replica matching the interest. In case of a cache hit, the interest packet is no longer forwarded, and the data packet is provided from the RCS's cache. Otherwise, the interest packet keeps getting forwarded towards the AoI using the same heading-aware geographical forwarding procedure adopted in CADD [4]. When receiving an interest packet, an RCS also stores information in the interest header about the RCS with the maximum centrality and the one with the second maximum centrality to be employed later for caching purposes.

When the interest packet reaches a vehicle in the AoI, it generates a data packet in response to the issued interest by

resolving the parameters available in the interest packet header. On its route to the destination vehicle, a copy of the reply packet gets cached at the RCS with the maximum centrality. This increases the likelihood of experiencing a cache hit by subsequent interests in the same data. Each reply-carrying vehicle checks the centrality of the passing-by RCS and sends a replica of the packet to it if its centrality matches the maximum centrality stored in the interest header. CADD-MR adopts the same caching mechanism proposed in CADD. Details about the adopted centrality-based caching mechanism and the underlying centrality computation are discussed in [4].

As the interest and reply packets are forwarded, each data-carrying vehicle checks its heading. The vehicle keeps carrying the packet if it is heading towards the packet's destination. If it is moving away from the intended destination, it forwards the packet to a passing-by RCS to give it a higher chance of being forwarded towards its destination.

While moving, a destination vehicle sends a registry packet to each passing-by RCS to register its ID, the ID of the interest packet, and its heading. Each RCS maintains a registry table that contains a tuple for each (vehicle ID, interest ID) pair, with the heading information stored for each associated pair. Upon receiving this registry packet, an RCS adds the necessary information to the registry table. A more detailed illustration of this process is provided later in this section.

Being unable to realize whether the destination vehicle has moved from its initial position or not, CADD-MR starts by directing the data packet towards the interest initiation position following the same forwarding mechanism as CADD. When the original position of the destination vehicle is reached, and the latter cannot be found, the reply-carrying vehicle anchors the packet towards the next RCS using greedy forwarding. This is done while checking for the destination in the neighborhood using the beacon packets exchanged among the neighboring vehicles. Similar to CADD, beacon packets are exchanged periodically between all vehicles and RCSs to be used for neighbor discovery. A beacon packet includes the ID, position, and velocity (for vehicles only) of the exchanging entities.

When an RCS receives a data packet intended for a missed destination, it checks its registry table to see whether or not it has an entry for that destination. If its registry table has a tuple with a matching (vehicle ID, interest ID) pair, the heading information is extracted, and the data packet is forwarded to the furthest vehicle in the corresponding road segment that is heading in the same direction. If no entry is found for that destination, the RCS drops the packet.

The reply packet follows the mobile destination via vehicular multi-hopping and using the guidance of the registry table at the passing-by RCSs. This is done until the packet is: delivered to the destination, the packet's expiry time (specified by the destination vehicle in the corresponding interest packet) is reached, or it is received by an RCS that does not have an entry for the destination in its registry table.

An instance of the operation of CADD-MR for handling mobile requesters is illustrated in the scenario shown in Figure 1. In this scenario, a requesting vehicle D is interested in data

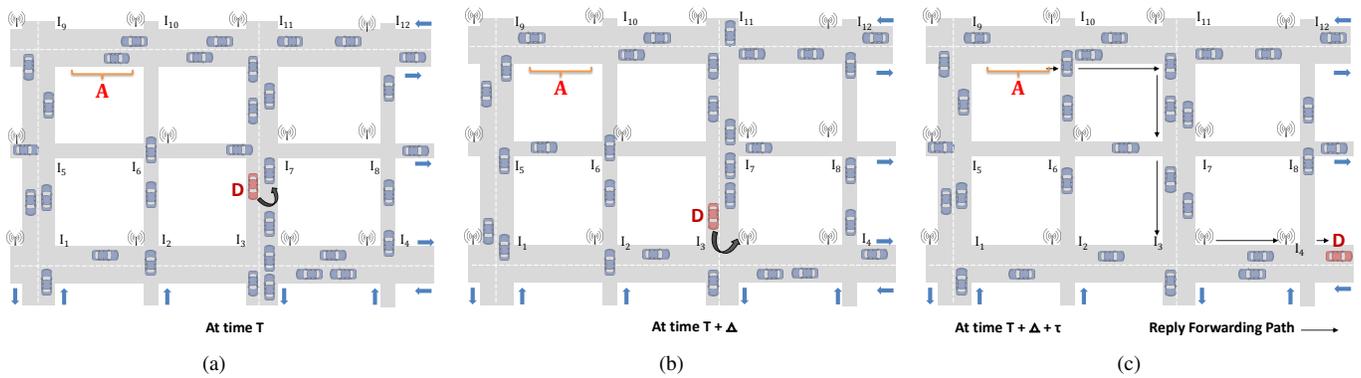


Fig. 1. An illustration of a scenario showing the use of RCSs to support mobile requesters.

from an area of interest A . It generates an interest packet and forwards it towards A , as illustrated in Figure 1(a). A few moments later, vehicle D passes by intersection I_3 and it sends a registry packet to the RCS deployed at that intersection, as shown in Figure 1(b), for that RCS to keep track of D 's heading. While moving, D keeps registering its heading on all the passing-by RCSs. In such a way, the corresponding reply packet manages to follow the requesting vehicle D on its movement path until it reaches D eventually as shown in Figure 1(c).

B. CADD-MR at a Requesting Vehicle

As opposed to CADD where the destination is a static entity, named gateway, whose sole purpose is to act as mediator between the end-users and the vehicular network, requesters/destinations in CADD-MR are the mobile vehicles thus additional functionality is required. As depicted in Algorithm 1, a requesting vehicle generates an interest packet and sets the associated parameters, including the interest type, AoI, the centrality-related caching fields, and the expiry times of the interest and its corresponding reply. Once the interest packet is generated, the requesting vehicle sends it to the nearest neighbor. It then waits for the reply packet. If it does not receive the latter before its expiration time, it assumes the packet got dropped (lines 6-9).

In addition to the above functionality, the requesting vehicle in CADD-MR informs each passing-by RCS of its heading as it moves from its initial position at the time of the interest initiation. As long as it is moving, the requesting vehicle checks whether it is approaching an RCS. If it is, it generates a packet, called the registry packet by including its ID, the ID of the interest packet, and its current heading to the header. It then sends the packet to the current RCS as it passes by (lines 14-18).

C. CADD-MR at a Forwarding Vehicle

A packet-forwarding vehicle implements different forwarding logic for an interest and a reply packet. This forwarding logic is illustrated in Algorithm 2. In CADD-MR, the interest packet forwarding procedure is the same as CADD. We provide a brief summary of this procedure below while details

Algorithm 1 : CADD-MR at a Requesting Vehicle

```

1: Input:
2: Neighborhood list  $N$ 
3:
4:  $request\_data()$ 
5: Begin
6: generate interest  $i$  with the corresponding parameters
7: initialize the centrality-related caching fields of the header
8: send  $i$  to the nearest neighbor in  $N$ 
9: keep track of  $i$  till either getting a reply or it expires
10: End
11:
12:  $register\_heading()$ 
13: Begin
14: while moving do
15: if  $Intersection\_Mode = true$  then
16: generate registry packet  $r$ 
17: add vehicle ID, ID of  $i$ , and current heading to  $r$ 's header
18: send  $r$  to  $RCS_{cur}$ 
19: End

```

are presented in [4].

An interest packet that has not expired yet is checked by the receiving vehicle to determine whether or not it is in the targeted AoI. If it is, the vehicle generates a reply packet, extracts the caching values from the interest packet header, and sets them accordingly before initiating the reply forwarding procedure. If the vehicle is not in the targeted AoI, it forwards the interest packet after first checking its mode of operation. If it is in the segment mode (i.e., not approaching any intersection), it directs the interest packet towards the next RCS using greedy forwarding procedure by sending the packet to the closest neighboring vehicle to the next RCS, if that neighboring vehicle is closer to the latter than the vehicle itself. Otherwise, when it receives a beacon packet from a nearby RCS, it sets its forwarding mode to the intersection mode (the vehicle is close to an intersection and can thus seek help from the approaching RCS). In this case, the interest-carrying vehicle forwards the packet to the neighboring RCS (RCS_{cur}).

In CADD-MR, the reply forwarding procedure is triggered when the data packet is generated by the vehicle itself or it is received by the vehicle. Either way, the reply-carrying vehicle checks whether the destination is among its neighbors.

Algorithm 2 : CADD-MR at a Forwarding Vehicle

```
1: Input:
2: Forwarding vehicle  $V$ 
3: Interest packet  $i$ 
4: Reply packet  $p$ 
5: Destination Vehicle  $D$ 
6: Set of neighboring road segments  $\mathcal{S}$  sent from a neighboring  $RCS$  along
   with their densities
7: Neighborhood list  $N$ 
8: Initial position of destination obtained from the interest packet header  $L$ 
9: Communication Range  $C$ 
10:
11:  $forward\_interest(i)$  //Same as in CADD [4]
12:
13:  $forward\_reply(p)$ 
14: Begin
15: if  $V$  is a neighbor to  $D$  then
16:   send  $p$  to  $D$ 
17: else
18:   calculate  $D$ 's initial search region  $R$  according to  $L$  and  $C$ 
19:   if  $V$  lies in  $R$ , and  $D$  is not in  $N$  then
20:     if  $Segment\_Mode = true$  then
21:       send  $p$  to the neighbor closest to  $RCS_{next}$ , if it is closer
         than  $V$ 
22:       if  $p$  is not relayed then
23:         keep holding  $p$  and check for  $D$  in  $N$ 
24:       else if  $Intersection\_Mode = true$  then
25:         send  $p$  to  $RCS_{cur}$ 
26:     else if  $V$  is not in  $R$  yet then // Initial position not reached
27:       if  $Intersection\_Mode = true$  then
28:         prioritize  $\mathcal{S}$  according to the density and direction priority
29:         for all  $s_i \in$  the prioritized segment set do
30:           if list of neighboring vehicles on  $s_i$  is not empty then
31:             send  $p$  to the farthest vehicle on  $s_i$ 
32:             break
33:           if  $p$  is not relayed then
34:             if  $V$  is heading towards  $D$  then
35:               keep holding  $p$ 
36:             else
37:               send  $p$  to  $RCS_{cur}$  with the forwarding flag  $ON$ 
38:           if  $RCS_{cur} = RCS_{max}$  then
39:             send a replica of  $p$  to  $RCS_{cur}$  with the caching flag
                $ON$ 
40:           else if  $V$  is heading away from  $RCS_{max}$  then
41:             forward a replica of  $p$  to  $RCS_{max}$  using the same
               procedure
42:         else //In the segment mode
43:           if  $N$  is empty then
44:             keep holding  $p$ 
45:           else
46:             send  $p$  to the neighbor closest to  $RCS_{next}$ , if it is closer
               than  $V$ 
47: End
```

If it is, the data packet is directly sent to the destination (lines 15 & 16). If it is not, the initial search region of the destination is calculated based on its initial location at the time of the interest initiation and the maximum coverage range. The carrying vehicle uses its own current position to determine whether or not it lies within the destination initial search region. If it is and yet the destination cannot be found among its neighbors, then it infers that the destination has moved from its initial position (line 19). In that case, if the vehicle is in the segment mode, it anchors the packet towards the next RCS using the same greedy forwarding procedure discussed earlier while continuously checking for the destination within its neighbors (lines 20-23). If it is in the intersection mode, it

forwards the packet to the neighboring RCS (lines 24 & 25).

As long as the destination initial position has not been reached, any data-carrying vehicle follows the same forwarding procedure used in CADD (lines 26-46). If the vehicle is in the segment mode, it anchors the packet towards the next RCS the same way discussed before (lines 43-46). In the intersection mode (lines 28-41), the vehicle uses the segments' density assessment, received from the neighboring RCS, along with segments' direction priority assessed in terms of the segment's heading towards the destination's initial location. Based on these two factors, the vehicle prioritizes and sorts the road segments. It then loops over the list of prioritized segments in a descending order to check for the availability of vehicles in a greedy manner, where the furthest vehicle on the best road segment is chosen as the next forwarder. Once the next forwarder has been found, the vehicle stops looping over the list of prioritized segments. Details about the density and direction assessment mechanisms are discussed in [4]. If the data-carrying vehicle does not find a potential forwarder, the vehicle keeps checking whether or not it is heading towards the destination initial location. If it is, it keeps holding the packet until a better forwarding opportunity arises. Otherwise, it sets the forwarding flag of the packet to ON and forwards it to RCS_{cur} to increase its chances of being forwarded towards the destination (lines 34-37). If, according to the header information, RCS_{cur} is the same as the RCS_{max} (i.e., the RCS with the maximum centrality), the reply-carrying vehicle sends a replica of the packet to RCS_{cur} to be cached and sets the caching flag to ON (line 39). If the vehicle is heading away from RCS_{max} , which implies that the packet may not have the opportunity of encountering the caching RCS, it forwards a replica of the packet to the latter using the basic forwarding procedure (line 41).

D. CADD-MR at an Intersection RCS

RCSs are used for three main purposes in CADD-MR. They handle caching, forwarding; especially when the packet-carrying vehicles are heading away from the destination, and keep track of the movement of requesting vehicles in a registry table as the vehicles pass by them. The logic of these three purposes is illustrated in Algorithm 3.

For the caching purpose, the RCS with maximum centrality on an interest forwarding path is used to cache a replica of its corresponding reply. When an interest packet is received from a vehicle by an RCS, the latter checks whether or not it has a replica of the requested data packet in its cache. If a matching entry is found, the RCS sends the data packet back to the requesting vehicle. Otherwise, the interest packet is directed towards the AoI. Before forwarding the unmatched interest packet, an RCS checks its centrality value to determine whether or not it is eligible for being the maximum central RCS (RCS_{max}) or the second maximum central RCS (RCS_{2max}) compared to the other RCSs encountered so far on the interest path. The ID and location of RCS_{max} and RCS_{2max} specified in the interest packet header get updated by each passing-by RCS that views itself as a better candidate

than the recorded ones. Such interest handling and forwarding procedure can be found in [4].

In order to keep track of the movement of a requesting vehicle and help navigate reply packets towards their destination, an RCS can receive a registry packet from a mobile requester when it gets passed-by. Once a registry packet is received by an RCS, it extracts the vehicle ID, the interest packet ID, and the heading of this requesting vehicle from the header of the packet. It then searches its registry table for a matching tuple for the (vehicle ID, interest ID) pair. If a matching entry is found, the heading registered in the table is compared with the heading specified by the registry packet. If both headings are not the same, the heading entry of the corresponding pair is updated. If there is no match, the RCS creates a new entry in the registry table for the received (vehicle ID, interest ID) pair and associates it with the requesting vehicle's specified heading (lines 14-20).

Algorithm 3 : CADD-MR at an RCS

```

1: Input:
2: Interest packet  $i$ 
3: Reply packet  $p$ 
4: Destination Vehicle  $D$ 
5: Set of neighboring road segments  $\mathcal{S}$ 
6: Neighborhood list  $N$ 
7: Registry Packet  $r$ 
8: Registry Table  $T$ 
9:
10:  $interest\_received(i)$  //Same as in CADD [4]
11:
12:  $registry\_received(r)$ 
13: Begin
14: extract header information of  $r$ 
15: if an entry exists in  $T$  for this  $r$ 's (vehicle ID, interest ID) pair then
16:     if heading in  $T$  does not match the heading in  $r$  then
17:         update heading in  $T$ 
18:     else // No matching entry in  $T$ 
19:         create a new entry for this (vehicle ID, interest ID) pair in  $T$ 
20:         add the heading in  $r$  to the newly added entry
21: End
22:
23:  $reply\_received(p)$  //Same as in CADD [4]
24:
25:  $forward\_packet(k)$  //  $k$  can be a reply packet or an interest packet
26: Begin
27: if  $k$  is a flagged reply packet then
28:     check registry table  $T$ 
29:     if there exists a corresponding entry in  $T$  for  $D$  then
30:         get the registered heading in  $T$ 
31:         select the road segment  $s$  that has the same heading of  $D$ 
32:         if list of neighboring vehicles on  $s$  is not empty then
33:             send  $k$  to the furthest vehicle on  $s$ 
34:         else // No corresponding entry is found
35:             drop  $k$  // destination is no longer interested in  $k$ 
36:     else if  $k$  is a non-flagged reply packet or an interest packet then
37:         follow the same forwarding procedure of CADD
38: End
39:
40:  $cache\_reply(p)$  //Same as in CADD [4]
41:
42: //The following two functions are called periodically upon the firing of
    corresponding timers. Their details are presented in CADD [4].
43:
44:  $calculate\_centrality()$ 
45:  $calculate\_popularity()$ 

```

When a reply packet is received by an RCS, the latter checks its forwarding and caching flags. If the caching flag is ON, the caching mechanism is executed as in CADD. If the maximum capacity of the RCS cache is reached, cache replacement is considered. In order to give the replaced packet another chance to be kept in the network instead of being dropped, it gets forwarded towards its corresponding RCS_{2max} for one more caching chance. Details about the adopted centrality-based, popularity aware caching mechanism are discussed in [4].

The forwarding procedure adopted by an RCS in CADD-MR checks whether the forwarded packet is an interest or data packet. If the packet to be forwarded is a data packet flagged as destined for a missing requesting vehicle, the RCS checks whether there is an entry for the destination vehicle in its registry table or not. If an entry is found, the registered heading is checked. The RCS then navigates the packet towards the direction of the destination by selecting the road segment corresponding to the same heading. Once the segment is determined, the RCS chooses the furthest neighboring vehicle in the selected road to forward the packet to (lines 27-33). In case a corresponding (vehicle ID, interest ID) pair is not found for a flagged packet, the RCS assumes that the interest initiator (requesting vehicle) is no longer interested in the packet and drops it (lines 34 & 35). If the packet to be forwarded is an interest packet or a non-flagged data packet, the RCS uses the forwarding procedure discussed earlier followed by a forwarding vehicle at an intersection before it reaches the destination's initial position (i.e., road segments are prioritized and sorted based on their densities and direction priorities and checked for a potential forwarder in the prioritized order).

IV. PERFORMANCE EVALUATION

We assess the performance of CADD-MR by comparing it to a modified SCF (M-SCF) scheme. In M-SCF, each interest initiator attaches its velocity at the time of the interest initiation, along with its position to the header of the interest packet. Once its initial position is reached and the destination cannot be found, a predicted position of the destination is estimated based on its position and velocity at the time of the interest initiation, and the time elapsed since the interest was issued until the data packet reaches the destination's initial position. A search region is then created from the current position of the reply packet to the estimated predicted position. A greedy forwarding procedure is used to forward the packet in that search region towards the estimated position. The reply-carrying-vehicles check whether the destination is in their neighborhood until the destination is found, the packet expires, or the end of the search region is reached.

We evaluate CADD-MR and M-SCF in terms of three performance metrics: 1) the average delay experienced from the moment the interest is generated until the reply packet is received, 2) the packet delivery ratio which is the ratio of reply packets that are successfully delivered to their corresponding destinations to the total number of reply packets sent, and 3) the ratio of received replies through accessing vehicular

resources to the total received replies, which is an indicator of the incurred access cost.

A. Simulation Setup

The NS-3 network simulator [6] is used for the implementation of CADD-MR and M-SCF over a 6×6 road grid topography. The SUMO vehicular simulator [7] and MOVE mobility generator [8] are used to generate realistic mobility traces with the maximum vehicular speed set to 40 km/h. The IEEE 802.11p WAVE standard is adopted for communication in the vehicular network. The beacon interval is set to 0.5 seconds and the transmission range is set to 150 m. Four vehicles are selected to act as interest initiators and requesting vehicles. The interest generation is uniformly distributed among the four requesters. The rate of interest generation is set to 25 seconds.

The schemes are evaluated over different vehicle densities, ranging from 200 to 1000 vehicles for a simulation duration of 2000 seconds each. In CADD-MR, an RCS is deployed at each intersection. The centrality period needed for the centrality calculation procedure is set to 250 seconds and the popularity period needed for the cache replacement procedure is considered to be the entire simulation time. The α and β weights needed for calculating segments' priorities are set to 0.2 and 0.8, respectively, giving a higher weight at each RCS to the segment whose direction brings a packet closer to the destination [4]. The cache size is set to 20 (allowing a maximum number of 20 packets to be cached).

B. Simulation Results and Analysis

First, we compare CADD-MR and M-SCF in terms of average delay over different vehicle densities. As depicted in Figure 2, CADD-MR shows a significant improvement over M-SCF. This can be attributed to the use of caching assistance of RCSs to bring data closer to the requesters, and to the heading-awareness feature of CADD-MR. This heading-awareness feature saves the interest and reply packets from going farther from their destinations. In addition, the capability of an RCS to track the movement of a destination vehicle through the registry table that it maintains contributes to the improved performance possessed by CADD-MR over M-SCF. It is evident that delay decreases as the number of vehicles increases since the larger the density of vehicles, the greater the forwarding opportunity of the packets towards their destinations.

Second, we compare the performance of CADD-MR and M-SCF in terms of packet delivery ratio. As depicted in Figure 3, CADD-MR significantly improves the packet delivery ratio compared to M-SCF. The reason is that M-SCF relies solely on the existence of the mobile requester in the neighborhood of the vehicles in the established search region; which does not guarantee that the requester can be found especially if it changes its heading after attaching it to the interest packet. On the contrary, CADD-MR seeks the assistance of RCSs for keeping track of the heading of the requesters and using such tracking registries for forwarding the reply packets towards their mobile requesters. In such a way, CADD-MR keeps the

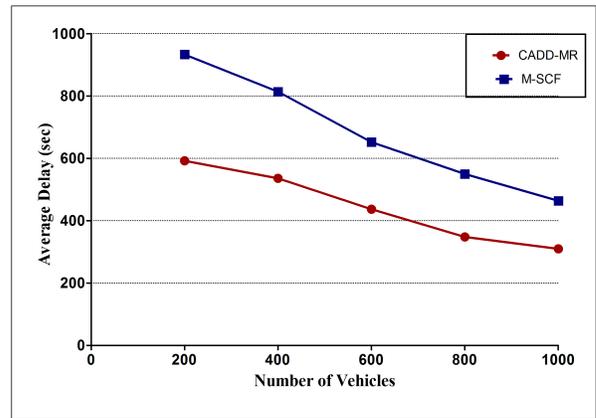


Fig. 2. Average Delay of CADD-MR and M-SCF with varying densities.

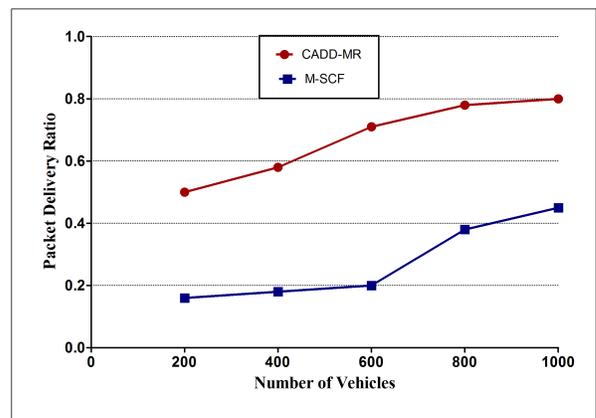


Fig. 3. Packet Delivery Ratio of CADD-MR and M-SCF with varying densities.

reply packets following their destined requesters until they eventually reach their destinations, which is the reason for the improved delivery ratio. In addition, the heading-awareness feature of CADD-MR saves packets from eventual dropping due to going away from their destinations.

Finally, we compare CADD-MR and M-SCF in terms of the ratio of vehicular resource-accessed replies to the total received replies. This metric is an indicator of the access cost such that the higher this ratio is, the higher the incurred access cost is. Since M-SCF is not caching-assisted, this ratio is always equal to 1 as all the replies come from accessed vehicular resources. As shown in Figure 4, CADD-MR significantly reduces this ratio due to its underlying caching assistance that introduces chances for cache hits.

V. CONCLUSIONS

In this paper, we proposed the caching-assisted data delivery with mobile requesters (CADD-MR) scheme to support having vehicles as requesters and consumers of road information. CADD-MR employs a caching concept that targets solving access cost and delay concerns through introducing cache hits. The proposed scheme utilizes road caching spots (RCS)

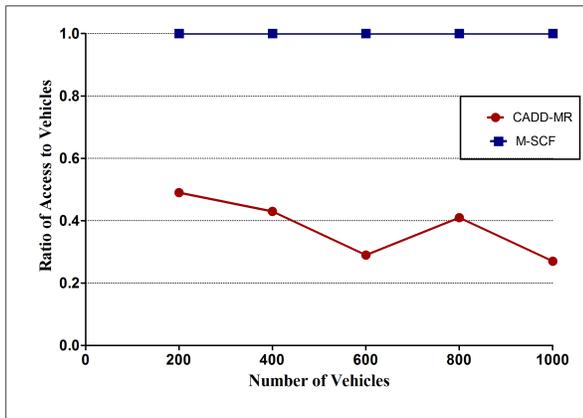


Fig. 4. Ratio of access to vehicular resources of CADD-MR and M-SCF with varying densities.

deployed at intersections to work as caching and forwarding assistants, and moving vehicles to work as data carriers and generators. The scheme considers vehicles' headings in forwarding interest and reply packets to direct them towards their destinations. CADD-MR utilizes the deployed RCSs for keeping track of the headings of the data requesting vehicles to make sure that the corresponding non-expired replies eventually reach their mobile destinations. Such a feature helps CADD-MR support having mobile requesters; hence, expand the scope of road information consumers. Performance evaluation was conducted and the results showed that CADD-MR achieves significant improvements in terms of the access delay, cost, and delivery ratio compared to another scheme that does not seek assistance from RCSs.

Our future work involves expanding the performance evaluation to assess the proposed scheme under more different practical scenarios with varying simulation parameters.

ACKNOWLEDGMENT

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

We thank Hesham Farahat for his assistance with the NS-3 simulation.

REFERENCES

- [1] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," in *IEEE Network Magazine*, vol. 29, no. 1, pp. 12-17, 2015.
- [2] A. T. Campbell, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, S. B. Eisenman, and G.-S. Ahn, "The rise of people-centric sensing," in *IEEE Internet Computing*, vol. 12, no. 4, pp. 12-21, 2008.
- [3] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a mobile sensor," in the *International Conference on Future Networks and Communications (FNC '14)*, Elsevier Procedia Computer Science, vol. 34, 2014, pp. 286-295.
- [4] 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.

- [5] H. T. Cheng, H. Shan, and W. Zhuang, "Infotainment and road safety service support in vehicular networking: from a communication perspective," *Mechanical Systems and Signal Processing*, Special Issue on Integrated Vehicle Dynamics, vol. 25, no. 6, pp. 2020-2038, Aug. 2011.
- [6] "The NS-3 network simulator." [Online]. Available: <https://www.nsnam.org/> (Accessed 2015, May 26).
- [7] "SUMO (simulation of urban mobility)." [Online]. Available: <http://sumo-sim.org/> (Accessed 2015, May 26)
- [8] Laboratory for Experimental Network and System (LENS). "MOVE (Rapid Generation of Realistic Simulation for VANET)," Available: http://lens1.csie.ncku.edu.tw/Joomla_Version/index.php/research-projects/past/18-rapid-vanet/ (Accessed 2015, February 17th)
- [9] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode, "VITP: an information transfer protocol for vehicular computing," in the 2nd ACM international workshop on Vehicular ad hoc networks, Germany, 2005, pp. 30-39.
- [10] C. E. Palazzi, F. Pezzoni, and P. M. Ruiz, "Delay-bounded data gathering in urban vehicular sensor networks," *Pervasive and Mobile Computing*, vol. 8, no. 2, pp. 180-193, 2012.
- [11] W. Ali, S. M. Shamsuddin, and A. S. Ismail, "A survey of web caching and prefetching," in *Int. J. Advance. Soft Comput. Appl.*, vol. 3, no. 1, pp. 18-44, 2011.
- [12] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: a survey," in *Computer Networks*, vol. 57, no. 16, pp. 3128-3141, 2013.
- [13] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache less for more in information-centric networks," ser. NETWORKING 2012, Lecture Notes in Computer Science. Springer, 2012, pp. 27-40.
- [14] N. Loulloudes, G. Pallis, and M. D. Dikaiakos, "Caching dynamic information in vehicular ad hoc networks," ser. Euro-Par 2010-Parallel Processing. Springer, 2010, pp. 516-527.
- [15] M. Amadeo, C. Campolo, and A. Molinaro, "CRoWN: Content-centric networking in vehicular ad hoc networks," in *IEEE Communications Letters*, vol. 16, no. 9, pp. 1380-1383, 2012.
- [16] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in the 5th international conference on Emerging networking experiments and technologies, 2009, pp. 1-12.
- [17] B. Brik, N. Lagraa, A. Lakas, H. Cherroun, and A. Cheddad, "ECDGP: extended cluster-based data gathering protocol for vehicular networks," in *Wireless Communications and Mobile Computing*, Wiley Online Library, 2015.
- [18] B. Brik, N. Lagraa, M. B. Yagoubi, and A. Lakas, "An efficient and robust clustered data gathering protocol (CDGP) for vehicular networks" in the 2nd ACM international symposium on Design and analysis of intelligent vehicular networks and applications, 2012, pp. 69-74.