On-Road Caching Assistance for Ubiquitous Vehicle-Based Information Services

Sherin Abdelhamid, *Member, IEEE*, Hossam S. Hassanein, *Senior Member, IEEE*, and Glen Takahara, *Senior Member, IEEE*

***Abstract*—Smart vehicles are considered major providers of ubiquitous information services. In this paper, we propose a solution for expedited and cost-effective access to vehicular public sensing services. The proposed caching-assisted data delivery (CADD) scheme applies caching on the delivery path of the collected data. Cached information can be used for later interests without having to request similar data from vehicles. For the operation of CADD, we introduce a lightweight road caching spot (RCS) to work as an on-road caching and forwarding assistant. CADD utilizes these caching spots along with vehicles on roads for handling the data collection and delivery processes. The proposed scheme involves a novel caching mechanism that utilizes real-time information for selecting the caching RCSs while considering popularity in cache replacement. A data chunk to be replaced may be forwarded to another less-loaded RCS. CADD considers vehicles' headings to direct communication toward the destination. Mathematical analysis and a simulation-based evaluation of the scheme are conducted. The evaluation results show significant improvements achieved by CADD in terms of access cost, delivery delay, and packet delivery ratio, compared with other schemes that do not involve caching assistance and do not take vehicles' headings into consideration.**

***Index Terms*—Caching, content popularity, data delivery, vehicular resources.**

I. Introduction

SMART vehicles with their abundant on-board resources, such as the sensing, computing, data storage, and relaying resources, are promising solutions for providing ubiquitous information services [1]. These services are not only confined to other vehicles on the road but to remote third parties as well. Vehicle-based services have broad potential for enhancing the public sensing domain [2]. A vehicle on the road can be considered a mobile sensor collecting data on the go, processing such data, and sharing it with others for supporting a wide range of information services [3]. Examples of such services include reporting road and weather conditions to interested authorities, reporting how crowded it is near points of interest, monitoring medium/long-lasting events on the road (e.g., fire), and reporting pollution and noise levels. These supported services can benefit a wide scope of consumers that include municipalities, governmental authorities, news and weather centers, end users, and other vehicles.

Utilizing vehicular resources for providing sensing-based services faces two major concerns. The resource owners need to get rewarded each time their resources are accessed, which brings forward an access cost challenge. Moreover, when data are needed from a specific area of interest (AoI), a sensing request needs to be forwarded toward that area, which imposes an access delay challenge. In this paper, we present a solution that handles the aforementioned access concerns through applying caching on the delivery path of the service data (i.e., caching the collected sensing data needed for providing the service).

We propose a caching-assisted data delivery (CADD) scheme that depends on utilizing caching spots deployed on the road for assisting in collecting vehicular data and providing vehicle-based information services. Our argument is that caching of collected data somewhere on the road helps in resolving later interests in similar data without having to access vehicular resources again. In addition, bringing the data of interest closer to the requesters through caching aids in reducing the access delay.

For achieving the targeted functionality of the proposed scheme, three design elements need to be addressed: 1) where caching can be applied on roads; 2) which caching mechanism to be used to cope with the dynamic nature of the vehicular network generating the data; and 3) how to achieve optimized forwarding of the interest and reply packets.

Some data delivery schemes that utilize assistance from roadside entities are available in the literature. These schemes depend on utilizing powerful roadside units (RSUs) for forwarding assistance without caching consideration. These RSUs can be adjusted and utilized for supporting the proposed caching concept; however, the cost of ubiquitous deployment will be high since the off-the-shelf RSU models available are not inexpensive. As a solution, we propose the use of a simple lightweight device that can complement RSUs when ubiquitous RSU deployments are not feasible. Our proposed road caching spot (RCS) consists of an 802.11p radio for communication, which comes with an embedded processor, a memory chip for caching, and a power port. Having only the components needed for forwarding and caching assistance, our proposed RCS is priced lower than an RSU.

As a default caching mechanism in caching-assisted schemes, ubiquitous caching is used for caching all content everywhere. Apparently, this mechanism is not efficient, and other caching mechanisms are proposed to solve its inefficiency [4]. One approach gaining popularity is the centrality-based

Manuscript received April 14, 2015; revised July 15, 2015; accepted September 2, 2015. Date of publication September 22, 2015; date of current version December 14, 2015. This work was supported by the Natural Sciences and Engineering Research Council of Canada. The review of this paper was coordinated by Guest Editors.

S. Abdelhamid and H. S. Hassanein are with the School of Computing, Queen's University, Kingston, ON K7L2N8, Canada (e-mail: sherin@cs.queensu.ca; hossam@cs.queensu.ca).

G. Takahara is with the Department of Mathematics and Statistics, Queen's University, Kingston, ON K7L2N8, Canada (e-mail: takahara@mast.queensu.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2015.2480711

0018-9545 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

caching approach [5], which aims at finding a subset of caching nodes that are the most central nodes in the network and directing caching to these nodes as these are the spots that are more likely to get many interests passing by and, hence, more cache hits. The most central nodes in a network are usually determined based on the locations of the nodes in the network graph such that those that can be encountered the most on the set of shortest paths between all pairs of nodes in the network are claimed to be the most central. Unfortunately, most of the caching mechanisms utilizing the centrality-based approach use such static computations for the centrality values as they target networks with static topologies such as the Internet backbone. As a vehicular network is dynamic in nature, we propose a dynamic centrality-based caching mechanism as part of the proposed CADD scheme. The proposed mechanism considers data popularity in cache replacement, favoring data types with more interests from end users. It is preferred that a popular data type with more generated interests than other types stay longer in the caching entities for the sake of increasing the chance of cache hits. In cache replacement, the proposed mechanism works on giving the to-be-replaced data chunk another caching opportunity instead of dropping it as commonly followed by other caching mechanisms.

For optimized forwarding of the interest and reply packets, the proposed data delivery scheme is *heading aware* in the sense that vehicles carrying packets to be forwarded check at intersections whether they are heading toward the packet destination or not. If a vehicle finds that it is heading away from the destination and it has no potential neighboring vehicle to forward the carried packet to, it seeks forwarding assistance from the neighboring RCS by offloading the packet to that RCS, giving it a better delivery chance. The proposed CADD scheme inherits this heading awareness feature from our previously proposed infrastructure-assisted data delivery (IADD) scheme [6]. Taking vehicles' headings into consideration improves the data delivery ratio and lowers the delivery delay through saving the carried packets from going away from their destinations.

Our contributions are summarized as follows.

- Proposing the CADD scheme aiming at reducing the access cost of vehicular resources and expediting the access time. To the best of our knowledge, CADD is the first data delivery scheme that considers caching assistance from roadside entities to support location-based vehicular information services;
- Introducing the RCS as a cost-effective assistant to the highly priced RSUs;
- Proposing a decentralized, dynamic, and centrality-based caching mechanism with data popularity in cache replacement for favoring highly interested data;
- Improving the commonly used cache replacement concept through considering recaching a replaced data chunk at another caching spot instead of dropping it.

We mathematically analyze the performance of the proposed CADD scheme in terms of its delay and access cost through introduced assessment models and a comparison to another scheme. First, we introduce a mathematical model to analyze and compare CADD to a scheme with interconnected RSUs. The assessment results show that the access delay of the connected RSU scheme falls between those of the best and worst cases of CADD, whereas its access cost is the same as CADD in its worst case. Second, we present another mathematical model to compute the estimated delay to reach an AoI from a considered gateway, providing a means for analyzing and assessing the scheme performance in a considered region. In addition to the mathematical analysis, we evaluate the performance of the proposed CADD scheme using the NS-3 simulator and compare it with the popular store–carry–forward (SCF) mechanism with no roadside assistance. Simulation results show that CADD achieves significant improvements in terms of access cost, delay, and delivery ratio.[1]

The remainder of this paper is organized as follows. In Section II, we discuss some related work on roadside-assisted data delivery for vehicular ad hoc networks (VANETs), vehicular data gathering schemes, and data caching. We present the proposed CADD scheme in Section III, including the caching mechanism. In Section IV, we mathematically analyze the performance of the scheme through introduced assessment models. In Section V, we present a simulation-based performance evaluation of the CADD scheme comparing it with the SCF scheme. We highlight some practical considerations related to the operation of the proposed scheme in Section VI. Finally, we conclude this paper and present our future work in Section VII.

II. Related Work

A. Roadside-Assisted Data Delivery for VANETs

Among the various VANET data delivery schemes available in the literature [8]–[10], some are proposed with roadside assistance to enhance the forwarding performance. All these schemes utilize assistance from RSUs deployed at intersections. Some of these schemes depend mainly on multihop communication through vehicles for forwarding [6], [11], and others assume that RSUs are interconnected through the Internet and utilize that for delay-critical forwarding [12], [13]. An example of the former category is our IADD scheme [6] that seeks RSU forwarding assistance in handling cases when the data carrying vehicles are moving away from the intended destination for the sake of improving both the data delivery ratio and delivery delay. Another example is the static-node-assisted adaptive routing protocol (SADV) [11] that utilizes RSUs at intersections for reducing the delivery delay. In SADV, a data packet can be stored at an RSU until a forwarding vehicle is encountered on the best delivery path for expedited forwarding. An example of the latter category is the infrastructure-assisted georouting scheme [12] that utilizes interconnectivity of RSUs for improved end-to-end performance by reducing the number of hops and, hence, the delivery delay. Another example is the infrastructure-assisted routing scheme proposed in [13] that follows the same concept proposed in [12], with the focus being on the RSUs' buffer allocation and management challenges.

Although the aforementioned schemes succeed in achieving performance improvements, they build on an assumption of having an RSU deployed at each intersection. As previously mentioned, the RSU models available on the market are quite pricey and, therefore, do not readily support such ubiquitous

[1] A simplified version of the work presented in this paper was introduced in our previous work in [7], which does not include the mathematical analysis section and has preliminary simulation results.

deployments. In addition, the assistance sought from RSUs in these schemes is only for forwarding purposes without caching considerations. Our proposed CADD scheme tackles these limitations through the introduction of the RCS priced lower and using it for both forwarding and caching assistance.

B. Vehicle-Based Data Gathering

Some platforms and schemes are proposed in the literature to utilize the resources of vehicles for gathering on-road data on-demand and handling interest and reply dissemination. The vehicular information transport protocol (VITP) [14] is proposed for the retrieval of vehicular information over VANETs through directing queries to AoIs and retrieving resolved replies with both query and reply dissemination being handled by intermediate vehicles via multihop communication. The VITP is only responsible for specifying the syntax and semantics of messages carrying location-dependent queries and replies between the nodes of a VANET. It works independently of the underlying VANET transmission and routing protocols.

Unlike the VITP, the delay-bounded vehicular data gathering (DB-VDG) solution [15] supports geographical vehicle-based data gathering services with taking into consideration the routing of query and reply messages. DB-VDG depends on vehicles as mobile sensors and data relays and on a fixed base station to create queries and collect replies back. Based on a delay bound, vehicles in DB-VDG decide on either forwarding the packets immediately or carrying them while moving, aiming at reducing the communication overhead and aggregating data from multiple sources.

Although the aforementioned solutions share with the proposed CADD scheme its basic target of accessing the vehicular sensing resources, they include very simplistic data gathering mechanisms that do not take the access cost into consideration.

C. Data Caching

Different data caching mechanisms have been proposed in the literature to handle, for instance, the web caching [16] and information-centric network caching [4] components. Examples of such mechanisms include the ubiquitous, probabilistic, and centrality-based caching. Among the aforementioned mechanisms, centrality-based caching has proven to be highly efficient in terms of cache hits with reasonable storage requirements [5]. Unfortunately, the centrality-based caching mechanisms proposed in the literature are designed for use in networks with static topologies, i.e., mainly the Internet. For example, the betweenness centrality-based caching mechanism [5] computes a node's centrality value as the number of times that node lies in the set of shortest paths between all pairs of nodes in the network, with the argument that if a node lies in many delivery paths, it is more likely to experience a cache hit. Apparently, this approach cannot be directly applied to a dynamic network such as a VANET. Therefore, there is a need for a mechanism that utilizes this centrality-based concept with support for highly dynamic networks.

In the area of caching in VANETs, a few schemes are proposed in the literature, which utilize the caching concept considering the mobile vehicles themselves to be the caching entities. In [17], Loulloudes *et al.* extended the location-aware VITP [14] to enable in-vehicle caching. The caching-enabled VITP allows the intermediate nodes to cache the replies on their way to the requesting nodes. While propagating the queries to the AoIs, the VITP-enabled nodes check their local cache for the possibility of cache hits. If a matching replica is not found locally, the query is forwarded to a neighboring vehicle. The authors used the time-to-live (TTL) value of the messages as the metric for cache replacement and management. A message is removed from the cache once its TTL reaches a predefined value. Another example is the CRoWN framework [18] that brings the content-centric networking concept [19] to the vehicular environment implemented on top of the IEEE 802.11p standard. In CRoWN, communications are driven by the contents instead of host addresses while exploiting in-network caching and replication to achieve fast content retrieval. Although these schemes have achieved performance improvements compared with their counterparts that do not consider caching, with the very dynamic nature of VANETs, caching replicas on mobile nodes that can be reached fortuitously limits the opportunities of cache hits. In addition, with the large-scale nature of VANETs, finding a vehicle with a replica of interest requires querying a huge number of vehicles. Thus, a solution that utilizes static nodes for on-road caching is much more desirable to increase the opportunities of cache hits and minimize the interest dissemination overhead.

III. Caching-Assisted Data Delivery

One of the main goals of the proposed CADD scheme is to minimize the cost of accessing the vehicular resources through deploying RCSs, one at each intersection, that can cache previously asked for data for satisfying later interests. The second goal is to reduce the roundtrip delay of the interest–reply cycle through utilizing the caching concept for bringing data closer to requesters and introducing cache hits on the interest dissemination path. To achieve these goals, the CADD scheme depends on the caching and forwarding capabilities of the deployed RCSs and vehicles on the road that work as carriers of both interests and replies.

As the caching concept is a main part of our proposed scheme, we introduce the RCS that works as our caching and delivery assistant to be deployed at each intersection. An RCS has only the components necessary for forwarding and caching; therefore, it is less expensive than the regular RSU models on the market. Details about the RCS structure and deployment are presented in Section VI-A.

Since our CADD scheme aims at providing vehicle-assisted information services to remote end users, a connection needs to be established between those end users and the vehicular network to be able to inject the service requests and get the data replies. As part of our system, we deploy gateway entities to work as the points of attachment (PoAs) of end users to the vehicular network. A single gateway will be deployed at each area of a city/deployment region, which will be divided into main nonoverlapping areas. An end user interested in a service communicates with a gateway through the Internet. End users, in the subscription stage, select which gateway will be their PoA based on the users' preference and the area from which they will be interested in getting services more often.

In addition to the *caching-assisted* feature of the proposed CADD scheme, another main feature is being *heading aware*. CADD builds on the heading awareness concept of our

TABLE I
LIST OF NOTATIONS

List of Notations	
General Notations	
r	Interest packet
p	Reply packet
p'	Packet with the lowest popularity in cache
v	Forwarding vehicle
s	Road segment
G_y	Gateway with ID y
RCS_l	Road caching spot with ID l
RCS_{max}	Road caching spot with maximum centrality on an interest forwarding path
RCS_{2max}	Road caching spot with second maximum centrality on an interest forwarding path
A_k	Area of interest with ID k
N	Neighborhood list of a node (either a vehicle or an RCS)
ρ_a	Density assessment period
ρ_c	Centrality period
ρ_p	Popularity period
ρ_r	Interest generation period
For Mathematical Modeling	
NI	Total number of generated interests
NS	Number of segments
c	Cost of getting a single reply through accessing vehicular resources
τ	Propagation delay over a road segment
$ADens_l$	Average density of RCS_l's incoming road segments
ADG_l	Average distance between RCS_l and the gateways
$\mathcal{S}_{y\to k}$	Set of RCSs on the full path from gateway G_y to area A_k
$\mathcal{P}_l$	Set of interest paths going through RCS_l
RS	Cache size of an RCS
P_{cache_r}	Probability of cache hit for interest r
P_{cmax_r}	Probability of cache hit at RCS_{max} of interest r
P_{c2max_r}	Probability of cache hit at RCS_{2max} of interest r
$DP^l_{r\_p}$	Degree of popularity of interest path $r\_p \in \mathcal{P}_l$
D_{max}	Distance from gateway G_y to $RCS_{max_{y\to k}}$
D_{2max}	Distance from gateway G_y to $RCS_{2max_{y\to k}}$
D_{A_k}	Distance from gateway G_y to area A_k
DC_l	Degree of centrality of RCS_l
C_{Th}	Caching threshold for cache probability

previously proposed IADD scheme [6], which considers vehicles' headings for the goals of improving the data delivery ratio and end-to-end delay.

In the following sections, we discuss the general operation of CADD through an illustrating scenario; then, we describe the detailed operations of the three main entities involved in the scheme: a vehicle on the road, an RCS at an intersection, and a gateway.

Table I summarizes the list of notations to be used throughout this paper.

A. CADD General Operation

The service acquisition process starts with an end user sending a request to the designated gateway that formulates a corresponding interest and then injects it into the vehicular network. The interest is disseminated in the network through vehicles and RCSs toward the AoI defined in the interest packet. While passing by RCSs in its path, the interest is checked at each passed-by RCS for a cached match (a cache hit). Unless a cache hit happens, the interest is forwarded toward the AoI using heading-aware geographical forwarding, as detailed later in this section.

Once the interest reaches a vehicle in the AoI, this vehicle generates a data reply packet resolving the interest parameters. On the way back to the requesting gateway, the reply is cached at the RCS with the maximum centrality among all the RCSs in the interest-followed path. Details about the proposed centrality-based caching mechanism are discussed in Section III-C5. With caching the collected data on the reply delivery path, the scheme brings chances for cache hits to be encountered by subsequent interests in the same data.

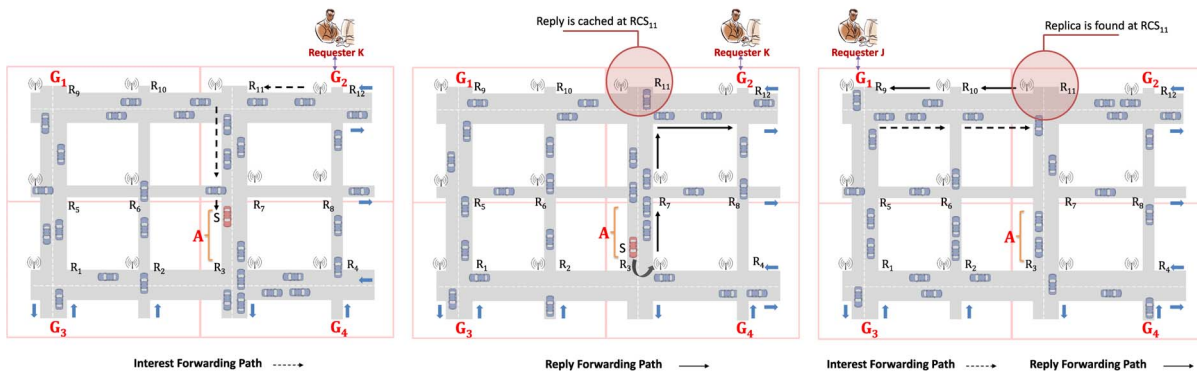
While propagating the interest and reply packets, the data carrying vehicles keep checking their headings at any approached intersection. If it happens that a vehicle does not find a neighboring vehicle as a potential forwarder, it keeps carrying the packet if it finds that it is going toward the packet's destination; otherwise, it leaves the packet at the neighboring RCS to give it a chance for a better forwarding opportunity toward its destination, which means a higher delivery probability for that packet.

All vehicles and RCSs exchange periodic beacon packets carrying their IDs, positions, and velocities (for vehicles) to announce their presence. Such beacons are used for neighbor discovery needed by the geographical forwarding procedure.

The scenario shown in Fig. 1 shows the benefits of caching assistance in reducing access to the vehicular resources, implying reducing the service access cost, and minimizing the data retrieval delay. In Fig. 1(a), requester K needs information on an event in AoI A. Since he has registered to gateway G_2, his request goes to it through the Internet. G_2 generates a corresponding interest, i.e., r_k, and sends it to the closest neighboring vehicle. The carrying vehicles follow the CADD heading-aware forwarding procedure detailed later to get the interest toward A resulting in the forwarding path shown in Fig. 1(a). During the forwarding process, the RCSs that are passed by the interest packet check the availability of a matching reply in their caches. In addition, they store information in the interest header about the maximum central and second maximum central RCSs to be used for the caching purposes. In the case shown, no match has been encountered; hence, the interest went to A.

In Fig. 1(b), r_k is received by a vehicle S in A. S generates a reply packet, i.e., p_k, with corresponding data matching the parameters of r_k, and caching-related fields matching those carried in r_k. S follows the CADD forwarding procedure to send p_k back to G_2, resulting in the path shown in Fig. 1(b). At each intersection, the carrying vehicle of p_k checks if the neighboring RCS is that with the maximum centrality stored in the packet header. In the illustrated scenario, it happened that RCS_{11} was the maximum central node on the interest path, and therefore, it is where a replica of p_k is cached on the way back. Note that an RCS at an intersection x is abbreviated as R_x in the figure for simplicity.

A moment later, a requester J, which is registered with gateway G_1, gets interested in information similar to that requested by requester K. G_1 generates a corresponding interest r_j and sends it toward A through the vehicular network. As r_j is checked at the passed-by RCSs for a matching reply, it happens that it hits the replica of p_k cached in RCS_{11}. Since p_k matches the parameters of r_j, a replica of it is forwarded to G_1. In this case, r_j is kept from going all the way toward A, minimizing the roundtrip access delay. Since the reply sent to J was a replica of a previously generated data packet, no access for vehicular



(a) (b) (c)

Fig. 1. Scenario showing the benefit of caching on the service data delivery path. Requester K needs information on an event in area A. He sends a request to G_2 that generates an interest following the path shown in part (a). A reply is generated by vehicle S in A and sent back to G_2 following the path shown in part (b) with a replica cached at RCS_{11}. A moment later, requester J asks for similar data from A. His interest meets the cached replica at RCS_{11}, resulting in the shortened interest and reply paths shown in part (c).

sensing resources was required, saving the cost that would have been incurred if caching was not considered. The interest and reply paths of this case are shown in Fig. 1(c).

B. CADD at a Vehicle on the Road

In CADD, vehicles work as the main carriers for both interest and reply packets. We detail the forwarding logic followed by a vehicle in Algorithm 1 for both forwarding an interest (lines 9–24) and forwarding a reply (lines 27–51). On a road, a vehicle can be in one of two modes: a segment mode or an intersection mode. In the *segment mode*, a vehicle is moving on a road segment and not close to any intersections, whereas in the *intersection mode*, a vehicle is approaching an intersection and, hence, has the opportunity of getting RCS assistance. We discuss the detailed forwarding logic below.

1) Interest Forwarding: When a vehicle gets an interest packet that has yet to expire, it checks if it is in the defined AoI. If so, it generates a reply packet; sets the corresponding caching values in the header, which it retrieves from the interest header; and triggers the reply forwarding procedure (lines 11–13). Otherwise, the vehicle checks its current mode of operation to start forwarding the interest. If it is in the *segment mode* (lines 17–21), it anchors the packet toward the next RCS $(\text{RCS}_{\text{next}})$ through the use of greedy forwarding. It checks its neighboring vehicles to find that which is closest to RCS_{next}, and if this potential forwarder is closer to RCS_{next} than the vehicle itself, it forwards the packet to it; otherwise, it keeps holding the packet until a better forwarder is encountered or it approaches RCS_{next}.

When a vehicle gets a beacon from an approached RCS, it activates the *intersection mode* of forwarding (lines 15 and 16). The carrying vehicle sends the interest to RCS_{cur}, which checks the possibility for a cache hit, if it has a matching reply in its cache, or continues the forwarding process toward the AoI otherwise.

2) Reply Forwarding: The reply forwarding procedure gets called when a vehicle generates a reply packet itself (if it gets an interest while it is in the corresponding AoI) or receives a generated packet to be forwarded. If the vehicle is in the *segment mode*, it forwards the packet toward RCS_{next} in the same way as previously defined in interest forwarding (lines 46–50). If it is in the *intersection mode*, it goes through many check points as follows.

a) **The vehicle checks if it has a candidate neighbor for forwarding the packet**. The beacon packet sent from RCS_{cur} carries the RCS's real-time assessment of the densities of its linked road segments. The density of a road segment s assessed by an RCS is defined as the number of beacons heard by that RCS from vehicles on s during an assessment period (ρ_a), i.e.,

$$\text{density}(s) = \text{no. of received\_beacons}_s \,|_{t}^{t+\rho_a} \tag{1}$$

where t is the start time of the current period. The vehicle uses the received densities for prioritizing the neighboring segments through computing a weighted priority (wpriority) for each segment as follows:

$$\text{wpriority}(s) = \alpha \times \text{density}(s) + \beta \times \text{dpriority}(s) \tag{2}$$

where $\text{dpriority}(s)$ is the priority of road segment s in terms of its direction toward the packet destination. The parameters α and β are tunable weights with $\alpha + \beta = 1$. The vehicle uses the prioritized segment list for finding its next-hop forwarder. It checks the availability of neighboring vehicles on the segments in a prioritized order, and if it finds any, it stops looping on the segments and selects the next forwarder on the segment with availability in a greedy fashion (lines 32–36).

b) **If not a), the vehicle checks if it is going toward the packet destination**. If it is, it keeps carrying the packet until it finds a better forwarder (lines 37–39). Otherwise, it sets a designated forwarding flag in the packet to ON and then sends it to RCS_{cur} to give it a better forwarding opportunity (lines 40 and 41).

c) **The vehicle checks if RCS_{cur} is the RCS with maximum centrality $(\text{RCS}_{\max})$ stored in the header for caching**. If so, the vehicle generates a replica of the reply packet, marks its caching flag as ON and sends it to RCS_{cur} to be cached (lines 42 and 43).

d) **If not c), the vehicle checks if it is moving far from $\text{RCS}_{\max}$ so that the packet may not pass by the caching RCS on its way**. If this is the case, the carrying vehicle generates a replica of the packet, sets its

destination to $\mathrm{RCS}_{\max}$, and forwards it toward $\mathrm{RCS}_{\max}$ using the same forwarding procedure used for forwarding a reply (lines 44 and 45). The idea behind this is to maintain the caching opportunity for that packet regardless of the path followed by the original packet itself.

Algorithm 1: CADD at a Vehicle

```
 1: Input:
 2: Forwarding vehicle v
 3: Interest packet r
 4: Reply packet p
 5: Gateway G
 6: Neighborhood list N
 7:
 8: forward_interest(r)
 9: Begin
10:   if r is not expired then
11:     if v is in the AoI defined in r then
12:       generate a reply packet p and set its RCS_max and
          RCS_2max fields as carried in r
13:       forward_reply(p)
14:     else
15:       if Intersection_Mode = true then
16:         send r to RCS_cur
17:       else    //In the segment mode
18:         if N is empty then
19:           keep holding r
20:         else
21:           send r to the neighbor closest to RCS_next if it
              is closer than v
22:   else
23:     drop r    // r is expired
24: End
25:
26: forward_reply(p)
27: Begin
28:   if v is a neighbor to G then
29:     send p to G
30:   else
31:     if Intersection_Mode = true then
32:       prioritize neighboring segments according to the
          density and direction priority
33:       for all segment s ∈ the prioritized segment set do
34:         if list of neighboring vehicles on s is not empty
            then
35:           send p to the farthest vehicle on s
36:           break
37:       if p is not relayed then
38:         if v is heading toward G then
39:           keep holding p
40:         else
41:           send p to RCS_cur with the forwarding flag ON
42:       if RCS_cur = RCS_max then
43:         send a replica of p to RCS_cur with the caching
            flag ON
44:       else if v is heading away from RCS_max then
45:         send a replica of p to RCS_max using the same
            procedure of the forward_reply(p) function
46:     else    //In the segment mode
47:       if N is empty then
48:         keep holding p
49:       else
50:         send p to the neighbor closest to RCS_next if it is
            closer than v
51: End
```

C. CADD at an Intersection RCS

RCSs are used mainly for caching replicas of the reply packets with the aim of resolving later relevant interests so that there will not be a need to access vehicular resources every time an interest is injected into the vehicular network. They are also used for forwarding assistance in cases when the packet-carrying vehicles are heading away from the destination. The logic followed by an RCS for both caching and forwarding is detailed in Algorithm 2 and discussed in the following.

1) Receiving an Interest: When an RCS receives an interest packet from a vehicle, if this interest is not yet expired, it checks its cache for a matching replica. If it finds a match, it sends a reply back to the requesting gateway (lines 10 and 11); otherwise, it forwards the interest toward the AoI using the forwarding procedure discussed later in this section (line 17).

Before forwarding a received interest, an RCS checks its centrality value relevant to the interest type and area, as discussed later in this section, to determine if it is a candidate for the maximum central RCS ($\mathrm{RCS}_{\max}$) or the second maximum central RCS ($\mathrm{RCS}_{2\max}$) among the RCSs encountered on the interest-traversed path so far (lines 13–16). In the interest header, the ID and location of $\mathrm{RCS}_{\max}$ and $\mathrm{RCS}_{2\max}$ are recorded along the path and are updated by a passed-by RCS if it is a better candidate than any of the recorded ones.

2) Receiving a Reply: After receiving a reply packet by an RCS, it starts checking the forwarding and caching flags carried in the reply header. If the caching flag is ON, the RCS calls the caching procedure for handling both the storage and cache replacement (lines 26 and 27). If the forwarding flag is ON, the RCS inserts the packet into the forwarding queue to be forwarded, as discussed below (lines 24 and 25).

3) Forwarding a Packet: The packet forwarding procedure is called by an RCS when it has to forward either an interest or a reply packet. An RCS's forwarding logic has similarities with that used by a vehicle for forwarding a reply. When a packet needs to be forwarded, the carrying RCS uses (2) for computing a weighted priority for all of its linked segments based on the density and direction criteria, as previously discussed. Afterward, the RCS searches for a neighboring vehicle on these road segments in the order of their weighted priorities (lines 32 and 33). If neighbors are found on a segment while searching, the RCS sends the packet to the farthest vehicle on that segment (lines 34–36). If no potential forwarder is encountered, the RCS keeps carrying the packet and marks it to be retransmitted (lines 37 and 38).

4) Caching a Reply: When an RCS receives a reply packet to be cached, it checks the availability of a vacant spot in its cache; if there is any, it caches the packet right away (lines 43 and 44). If the RCS's cache is full, it considers replacing a previously cached packet (lines 45–51), as discussed below.

One of the main contributions of this paper is the proposed cache replacement mechanism, which, compared with other caching mechanisms that drop the replaced packet, gives this packet another caching chance to stay longer in the network and increase the chances for a cache hit. In contrast to many caching mechanisms that consider picking the to-be-replaced packet using the least recently used (LRU) policy while ignoring the popularity of the different packets, our replacement mechanism considers this popularity criterion in picking the replacement

candidate. Considering popularity favors the packets with more interests from end users, which enhances the quality of service and increases the chance of cache hits.

Algorithm 2: CADD at an RCS

```
1: Input:
2: Interest packet r
3: Reply packet p
4: Gateway G
5: Neighborhood list N
6:
7: interest_received(r)
8: Begin
9:   if r is not expired then
10:     if there is p matching r in Cache then
11:       forward p to G by calling forward_packet(p)
12:     else
13:       if centr_{r_(tp,a)} of the RCS > centr of RCS_max
          then
14:         update RCS_max, RCS_2max, and their stored
            locations
15:       else if centr_{r_(tp,a)} of the RCS > centr of
          RCS_2max then
16:         update RCS_2max and its stored location
17:       forward_packet(r)
18:   else
19:     drop r   // r is expired
20: End
21:
22: reply_received(p)
23: Begin
24:   if the packet's forwarding flag is set to ON then
25:     forward_packet(p)
26:   if the packet's caching flag is set to ON then
27:     cache_reply(p)
28: End
29:
30: forward_packet(k)   // k can be a reply packet or an
    interest packet
31: Begin
32:   prioritize neighboring segments according to the density
      and direction priority
33:   for all segment s ∈ the prioritized segment set do
34:     if list of neighboring vehicles on s is not empty then
35:       send k to the farthest vehicle on s
36:       break
37:   if a forwarder is not found then
38:     keep holding k to be retransmitted
39: End
40:
41: cache_reply(p)
42: Begin
43:   if Cache is not full then
44:     store p in Cache
45:   else   //Cache is full, consider replacement
46:     p' ← the packet with the lowest popularity in Cache
47:     if p.pop < p'.pop then
48:       forward p to p.RCS_2max
49:     else
50:       forward p' to p'.RCS_2max
51:       store p in Cache
52: End
53:
54: //The following two functions will be called periodically
    upon the firing of corresponding timers.
55: calculate_centrality()   //as in (3)
56: calculate_popularity()   //as in (4)
```

When a replacement needs to be considered, the RCS with the full cache picks the least popular packet among those stored in its cache (line 46). Details on popularity computation are discussed later in this part. When it happens that an RCS finds many packets with the same lowest popularity value, it picks the LRU one among those of equal popularity. Then, the RCS compares the popularity of the candidate packet (p') to the to-be-cached packet (p). That with the higher popularity will be the caching winner, and the other will be given another caching chance. Each reply packet carries in its header information the second maximum RCS ($\mathrm{RCS}_{2\max}$) encountered on the interest path, and this is where the other caching chance will be targeted. The packet with the lower popularity between p' and p will be offloaded to the $\mathrm{RCS}_{2\max}$ recorded in its header using the same forwarding procedure (lines 47–51). A packet is kept in an RCS's cache until the packet expires or gets replaced.

5) Centrality Computation: Another main contribution in this paper is our dynamic decentralized mechanism for computing the centrality of the RCSs. These centrality values are used for caching purposes through directing a reply replica to the maximum central node encountered on the interest forwarding path. A replaced replica is directed to the second maximum central node encountered on its interest path, as previously discussed.

Among the different caching mechanisms proposed in the literature, we build on the centrality-based caching mechanism due to its demonstrated ability to achieve high levels of cache hits with less overhead compared with many other mechanisms [5]. The concept of centrality-based caching aims at directing caching to the nodes that are most central in the network as they are more likely to get many interests passing by and, hence, more cache hits. Unlike the static centrality-based caching mechanisms that compute the centrality value of a node based on its position in the static network topology, our proposed mechanism handles the computation in a dynamic way based on the number of interests received by a node such that the more interests a node receives, the more central it is in the network, and the more likely it will get upcoming interests.

In CADD, all the potential interests are classified into main types (e.g., traffic conditions, road conditions, and crowd). Each RCS maintains an Interest Type_Area table that lists these predefined interest types, each associated with the different predefined main areas of the whole region in (type, area) 2-tuples. The RCS computes its *centrality* for each tuple and uses it when it receives an interest of that tuple for checking its candidacy to be the $\mathrm{RCS}_{\max}$ or the $\mathrm{RCS}_{2\max}$ on that interest forwarding path.

As shown in (3), an RCS's centrality of a specific tuple is computed as the number of interests of that type toward that area received during a predefined centrality period (ρ_c), i.e.,

$$\text{centrality}_{r\_(\text{type,area})} = \text{no. of interest}_{r\_(\text{type,area})}\,\Big|_{t}^{t+\rho_c} \quad (3)$$

6) Popularity Computation: Using a similar dynamic mechanism as that used for computing the centrality, an RCS computes a popularity value for all of the different packets it carries in its cache as the popularity of its corresponding interest tuple. A tuple *popularity* is computed as the number of interests of that type toward that area received during a popularity period ρ_p, as follows:

$$\text{popularity}_{r\_(\text{type,area})} = \text{no. of interest}_{r\_(\text{type,area})}\,\Big|_{t}^{t+\rho_p} \quad (4)$$

Algorithm 3: CADD at a Gateway

```
1: Input:
2: Neighborhood list N
3:
4: request_data()
5: Begin
6:   generate interest r with the corresponding parameters
7:   initialize the centrality-related caching fields of the
     header
8:   send r to the nearest neighbor in N
9:   keep track of r until either getting a reply or it expires
10: End
```

The RCS uses the computed popularity for picking the caching replacement candidate, as previously discussed.

D. CADD at a Gateway

In CADD, a gateway is used as the PoA of the requesters to the vehicular network. Its main job in the scheme is injecting the service requests into the network and waiting for the replies, which is presented in Algorithm 3 through the $request\_data$ procedure. When a gateway receives a service request through the Internet, it generates an interest packet with the corresponding parameters defined in the request, including the interest type, AoI, and the expiry times of the interest and its requested reply. Before injecting the interest into the network, the gateway initializes the caching-related fields of the interest header: the centrality and IDs and locations of $\text{RCS}_{\max}$ and $\text{RCS}_{2\max}$. Then, it sends the generated interest to its closest neighbor of the vehicular network. A gateway keeps track of all the interests it sent until it gets matching replies or the interests expire. Once a gateway receives a reply matching with a stored interest, it sends this reply back to the requester through the Internet.

IV. Mathematical Analysis of the Proposed Caching-Assisted Data Delivery Scheme

Here, we analyze the delay and access cost of the proposed CADD scheme through mathematical modeling and assessment.

A. CADD and Connected_RSUs

Here, we present a mathematical model to analyze and compare CADD to a scheme utilizing RSUs with interconnection (CRSU). We consider an $m \times n$ grid topology with an RSU deployed at each corner working as a gateway for both schemes. In CRSU, the four RSUs can communicate with one another through the Internet such that they can move the interests and replies closer to their destinations through their backhaul connection. For CADD, an RCS is deployed at each intersection point of the grid.

In assessing the performance of CADD, we consider its best case (CADD_B) when a cache hit is encountered at the next intersection RCS and its worst case (CADD_W) when no cache hits happen on the interest path; hence, the interest has to be resolved by a vehicle at the AoI.

In our assessment scenario, we consider each segment on the grid as a possible AoI with one interest generated to it by an RSU/gateway. In an $m \times n$ grid, the number of segments (NS), which is equal to the total number of generated interests (NI), is computed as: $NS = NI = 2mn - m - n$.

As a first assessment metric, we consider the total cost $(TCost)$ incurred by each scheme for getting a reply to each of the NI interests. Assume that the cost of getting a single reply through accessing vehicular resources is c. In both CADD_W and CRSU, each interest is resolved by a vehicle in the AoI; hence, their $TCost$ is equal to $NI \times c$. For CADD_B, as all interests are resolved through cache hits, its $TCost$ is equal to 0.

As a second metric, we compute the average delay $(ADelay)$ of reaching all the segments from the requesting RSU. We assume that the propagation delay over a road segment is τ and is the same for all segments. Since in CADD_B, a cache hit is encountered at the next intersection (one segment away), the delay of getting a reply targeting any segment is equal to τ, except for the two segments directly linked to the requesting gateway where the delay is 0. Consequently, the $ADelay$ of CADD_B is computed as

$$ADelay(\text{CADD}_B) = (NI - 2) \times \tau / NI \tag{5}$$

For CADD_W, the delay of reaching an AoI/segment starting at point (i, j) on the grid and the average delay can be computed as in (6) and (7), respectively shown below. Thus

$$Delay(\text{CADD}_W)_{(i,j)} = (i + j)\,\tau \tag{6}$$

$$ADelay(\text{CADD}_W) = \frac{\left[2\sum_{i=0}^{m-2}\sum_{j=0}^{n-2}(i+j) + \sum_{i=0}^{m-2}(i+n-1) + \sum_{j=0}^{n-2}(j+m-1)\right]\tau}{NI} \tag{7}$$

To reach an AoI starting at point (i, j) in CRSU, since the RSUs are connected, the delay is the minimum delay of reaching this segment from the four RSUs, as computed in (8). Equations (9) and (10), shown below, show the computation of the CRSU average delay for all different cases of the m and n values. Thus

$$Delay(\text{CRSU})_{(i,j)} = min[(n-1-j+i),(i+j),(m-i+n-j-2),(m-1-i+j)]\times\tau \tag{8}$$

$$ADelay(CRSU) = \frac{Total\_Delay(CRSU)}{NI} \tag{9}$$

$$Total\_Delay(CRSU) = \begin{cases} W \times \tau, & \text{if both } m\&n \text{ are odd} \\ [W+X+Z]\times\tau, & \text{if } m \text{ is even and } n \text{ is odd} \\ [W+Y+Z] \times \tau, & \text{if } m \text{ is odd and } n \text{ is even} \\ [W+X+Y+(Z\times 4)] \times \tau, & \text{if both } m\&n \text{ are even} \end{cases} \tag{10}$$

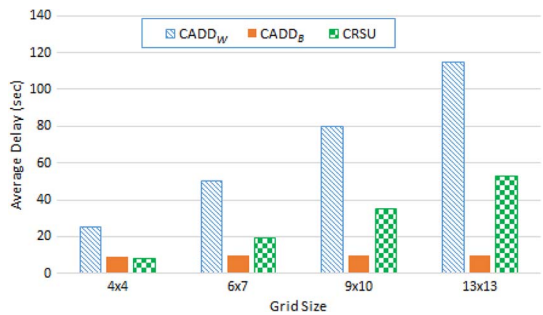


Fig. 2. Average delay of CADD and CRSU with varying grid sizes.

where

$$W = \left(\left(8 \times \sum_{i=0}^{U-1}\sum_{j=0}^{V-1}(i+j)\right) + \left(2 \times \left(\sum_{i=0}^{U-1}(i+V) + \sum_{j=0}^{V-1}(j+U)\right)\right)\right)$$

$$X = 4 \times \sum_{i=0}^{U-1}(i+V), \quad Y = 4 \times \sum_{j=0}^{V-1}(j+U)$$

$$Z = U + V, \quad U = \left\lfloor\left(\frac{m-1}{2}\right)\right\rfloor, \quad V = \left\lfloor\left(\frac{n-1}{2}\right)\right\rfloor.$$

Fig. 2 shows the assessment results of CADD$_W$, CADD$_B$, and CRSU in terms of the average delay computed based on (5), (7), and (9) with varying grid sizes and τ equal to 10. The results show that in small areas (small grid sizes), the average delay of CADD$_B$ is almost equal to that of CRSU. In larger areas, the average delay of CRSU is significantly higher than the delay of CADD$_B$ and getting closer to that of CADD$_W$ as many of the segments get farther from the corner RSUs.

B. CADD to a Specific AoI

Here, we consider analyzing the delay to reach a specific AoI by an interest packet initiated from a specific gateway. First, we consider this analysis under the assumptions of having a fixed traffic density for each road with non-uniform density distribution for the whole set of segments (i.e., each segment has a fixed density that may differ from other segments' densities) and having a uniform popularity for all the interest tuples. Second, we consider the analysis under the fixed non-uniform density distribution assumption with non-uniform popularity distribution for the interest tuples.

1) Analysis Considerations: We consider an $m \times n$ grid topology with a gateway deployed at each corner and an RCS deployed at each intersection point of the grid. The delay is estimated for an interest packet going from a gateway to the starting point of an AoI (road segment) on the grid. Note that the location points of the gateway, AoI reaching point, and the RCSs are mapped to points on the grid in terms of m and n.

The density of road segments, popularity of interest tuples, and locations of RCSs on the grid are known and used as inputs to the assessment model. Based on this information, the full paths from the considered gateway (G_y s.t. $y = 1, \ldots,$ no. of Gateways) to each candidate AoI (A_k s.t. $k = 1, \ldots,$ no. of AoIs) are computed *a priori* based on the CADD forwarding procedure as a list of junction/intersection points. Therefore, the set of RCSs on the full path from G_y to A_k is known and referred to as $\underset{y\to k}{\mathcal{S}}$.

2) Analysis Model:

Case I: Non-uniform Density and Uniform Popularity: First, based on the inputs previously mentioned and the computed full path between the considered G_y and A_k, we compute the locations of RCS$_{\max}$ and RCS$_{2\max}$ on that path. For each RCS$_l \in \underset{y\to k}{\mathcal{S}}$, we compute a corresponding degree of centrality DC_l that indicates the candidacy of that RCS to be the most central node. RCS$_{\max}$ on a path represents the node with the highest number of interests passing by. In case there is more than one RCS encountering the same highest number of interests, RCS$_{\max}$ is chosen to be that closest to the gateways among those nodes with the maximum centrality. Based on this centrality notion, the DC_l of RCS$_l$ is computed as

$$DC_l = ADens_l \times \frac{1}{ADG_l} \tag{11}$$

where $ADens_l$ is the average density of RCS$_l$'s incoming road segments, and ADG_l is the average distance between RCS$_l$ and the gateways considering only those having RCS$_l$ between the gateway and AoI (i.e., the gateways that might have RCS$_l$ on the path to the considered AoI).

According to the computed degrees of centrality, we determine $\underset{y\to k}{\text{RCS}_{\max}} \in \underset{y\to k}{\mathcal{S}}$ and $\underset{y\to k}{\text{RCS}_{2\max}} \in \underset{y\to k}{\mathcal{S}}$ of the path from G_y to A_k as follows:

$$\underset{y\to k}{\text{RCS}_{\max}} \longleftarrow \text{RCS}_l \text{ with } \max(DC_l)$$
$$\underset{y\to k}{\mathcal{S}} \longleftarrow \underset{y\to k}{\mathcal{S}} \setminus \underset{y\to k}{\text{RCS}_{\max}}$$
$$\underset{y\to k}{\text{RCS}_{2\max}} \longleftarrow \text{RCS}_l \text{ with } \max(DC_l).$$

Consider the probability of having a cached replica on the path to be $P_{\text{cache}} = P_{\text{cmax}} + P_{\text{c2max}}$, where P_{cmax} is the probability of having a cached replica at RCS$_{\max}$, and P_{c2max} is the probability of having a cached replica at RCS$_{2\max}$. Note that according to the scheme, a cached replica can only be found at either RCS$_{\max}$ or RCS$_{2\max}$, but not in both. Accordingly, the estimated distance to be traversed by an interest r initiated through G_y and directed toward A_k can be computed as

$$\underset{y\to k}{EDist_r} = [P_{cmax_r} \times D_{\max}] + [P_{c2max_r} \times D_{2\max}] + [(1 - P_{cache_r}) \times D_{A_k})] \tag{12}$$

where $D_{\max}$, $D_{2\max}$, and D_{A_k} are the distances from G_y to the determined $\underset{y\to k}{\text{RCS}_{\max}}$ and $\underset{y\to k}{\text{RCS}_{2\max}}$, and A_k, respectively. Note that the distances in the model are expressed in the number of road segments between a pair of points on an $m \times n$ grid. For example, the distance between a specific G_y at point $(0, n-1)$ and an A_k reachable through point $(m-1, 2) = [|m-1-0| + |2-(n-1)|] = m - n + 2$ segments.

The last term in (12) represents where the requested interest r has not been previously requested, or was requested and

reached its expiry time before initiating the current interest packet, as the worst case of CADD where the interest packet has to go all the way toward the AoI. The probability of a cache hit, i.e., $P_{\text{cache}_r} = P_{\text{cmax}_r} + P_{c2\text{max}_r}$, can be computed as detailed below.

We consider the popularity of the interest tuples and the cache sizes of the RCSs for computing P_{cache}. Based on the full path set computed *a priori* from the gateways to the AoIs, for each RCS$_l$, we compute the set of interest paths going through it, referring to it as $\mathcal{P}_l$. Accordingly, with the assumption that all interest tuples have the same popularity, $P_{c\,\text{max}_r}$ and $P_{c2\text{max}_r}$ of an interest r from G_y to A_k are computed as in (13) and (14), respectively, shown below. Thus

$$P_{\text{cmax}_r} = \frac{1}{\left|\mathcal{P}_{\underset{y\to k}{\text{RCS}_{\max}}}\right|} \times RS \tag{13}$$

$$P_{c2\text{max}_r} = \frac{1}{\left|\mathcal{P}_{\underset{y\to k}{\text{RCS}_{2\max}}}\right|} \times RS \times (1 - P_{\text{cmax}_r}) \tag{14}$$

where RS is the cache size of an RCS. In cases where the values of P_{cmax_r} and $P_{c2\text{max}_r}$ are greater than 1 due to having $|\mathcal{P}_{\underset{y\to k}{\text{RCS}_{\max}}}|$ much smaller than RS, P_{cmax_r} and $P_{c2\text{max}_r}$ are assigned the probability upper bound 1.

The probabilities of caching obtained through (13) and (14) are plugged into (12) to compute $\underset{y\to k}{EDist_r}$. Finally, following the general assumption considered in Section IV-A of having the propagation delay over a road segment to be τ, we can compute the estimated delay for an interest r initiated through G_y and targeting data about A_k, as shown in

$$\underset{y\to k}{EDelay_r} = \underset{y\to k}{EDist_r} \times \tau. \tag{15}$$

Case II: Non-uniform Density and Non-uniform Popularity: The same model and formulation discussed under Case I applies to this case as well, except for the computation of P_{cmax_r} and $P_{c2\text{max}_r}$. Since non-uniform popularity is considered (i.e., each interest tuple may have a different popularity), we rank the different interest paths in $\mathcal{P}_l$ of each RCS$_l$ in ascending order according to the popularity of its corresponding interest tuple, subject to that the interest tuple with the highest popularity gets a rank of 1. Each interest path $r\_p \in \mathcal{P}_l$ is assigned a degree of popularity, i.e., $DP^l_{r\_p}$, equivalent to its rank in $\mathcal{P}_l$. P_{cmax_r} and $P_{c2\text{max}_r}$ of an interest r from G_y to A_k following a path r_p are computed as in (16) and (17), respectively, shown below. Thus

$$P_{\text{cmax}_r} = \begin{cases} 1, & \text{if } DP_{r\_p}^{\underset{y\to k}{\text{RCS}_{\max}}} \le RS \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$P_{c2\text{max}_r} = \begin{cases} 0, & \text{if } DP_{r\_p}^{\underset{y\to k}{\text{RCS}_{2\max}}} > RS \\ & \text{or } P_{\text{cmax}_r} \text{ is equal to 1} \\ 1, & \text{otherwise.} \end{cases} \tag{17}$$

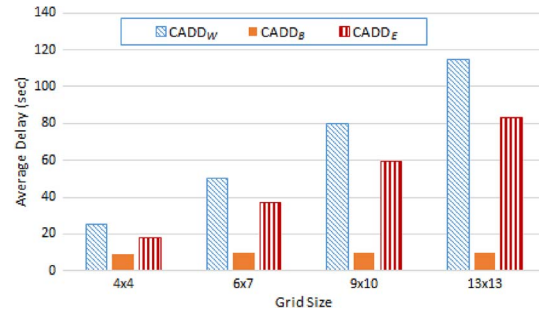


Fig. 3. Average delay of CADD$_E$, CADD$_W$, and CADD$_B$ with varying grid sizes.

It is worth noting that in the two cases of the given analysis, we consider the popularity values at the time of computing the estimated delay. Such values may vary at different times due to the dynamic nature of popularity, but this does not affect the computation at a specific time.

3) Numerical Results: Below are the results of computing the average estimated delay for reaching data about all the segments in the $m \times n$ grid topology, considering each segment on the grid as a possible AoI and one of the corner gateways to be the initiator of all the interests targeting the segments. To generalize and assess the model for different grid sizes, we adopt uniform traffic density with uniform interest popularity for all the segments in the grid.

Since having uniform density over the segments leads to having multiple potential paths between gateway G_y and AoI A_k, we adopt the following strategy to form a path from G_y to A_k. Among the set of shortest paths from G_y to A_k, we select that giving priority in segment selection in a counterclockwise fashion (e.g., considering the gateway in the top right, the selection strategy picks all the segments to the left until reaching the same grid column of A_k and then moves down selecting all the segments on the way until reaching A_k). Due to the use of uniform density as well, the use of the $ADens_l$ parameter in (11) is relaxed, leading to considering only the average distance to gateways (ADG_l) in computing the DC_l of RCS$_l$.

For each AoI/road segment, we generate a random value to indicate if an interest to that AoI was requested before or not (i.e., there is a probability of a cache hit or not). If this generated value is greater than a predefined threshold (C_{Th}), it will be assumed that an interest r to that AoI will not meet a cached replica anywhere on the grid, leading to $P_{\text{cache}_r} = P_{\text{cmax}_r} = P_{c2\text{max}_r} = 0$; hence, having the estimated delay converges to that of CADD$_W$ for that AoI. If this value is less than or equal to C_{Th}, the values of P_{cmax_r}, $P_{c2\text{max}_r}$, and P_{cache_r} are computed according to the proposed model.

We then compare this model, referring to it as CADD$_E$, to CADD$_W$ and CADD$_B$ defined in the previous part in terms of the average delay to reach all the segments in the grid for different grid sizes. Fig. 3 shows the results of the comparison, considering a cache size that is equal to 20, C_{Th} equal to 0.5, and τ equal to 10. As expected, the delay of CADD$_E$ lies between these of CADD$_W$ and CADD$_B$. For larger grid sizes, the difference between CADD$_W$ and CADD$_E$ is more apparent as the distances between the gateway and some of the segments

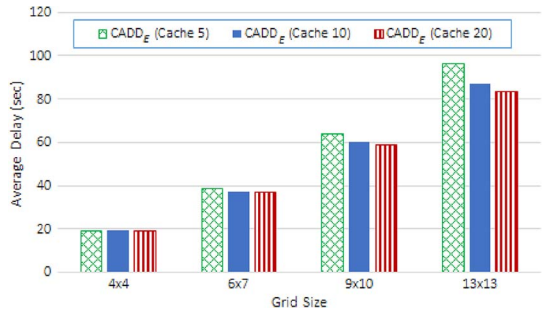


Fig. 4. Average delay of CADD$_E$ with different cache sizes over varying grid sizes.

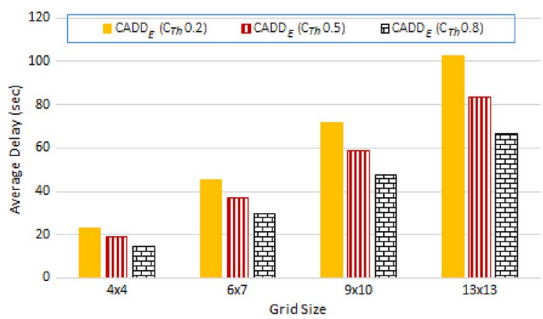


Fig. 5. Average delay of CADD$_E$ with different caching probability thresholds over varying grid sizes.

get longer, with a probability of not being traversed in CADD$_E$ due to the potential cache hits.

We also assess the average delay computed by CADD$_E$ with considering different cache sizes (RS) over different grid sizes. The results in Fig. 4 show that with increasing the cache size, the average delay decreases due to increasing the values of P_{cmax_r} and $P_{c2\mathrm{max}_r}$ per an interest r, which decreases the corresponding $EDist_r$, according to (12).

Finally, we assess CADD$_E$ with considering different values for C_{Th} over different grid sizes. The results in Fig. 5 show that with increasing C_{Th}, the delay decreases, as fewer segments would be assumed not having a cached replica leading to lower estimated distances to more segments.

V. Simulation-Based Evaluation of the Proposed Caching-Assisted Data Delivery Scheme

Here, we evaluate the performance of the CADD scheme using simulation assessment. The performance of CADD is analyzed and compared with the nonassisted SCF scheme in which a vehicle continues to carry the data if it does not find a potential next-hop forwarder even if the vehicle is moving away from the destination's direction. The two schemes are compared in terms of 1) the average round-trip access delay, since an interest is sent by a gateway until its reply is received; 2) the packet delivery ratio, highlighting the benefit of the heading awareness feature; and 3) the cache hit ratio, which is an indicator of the saved access cost.

Fig. 6. Part of Kingston city representing the simulation topography.

A. Simulation Setup

Both CADD and SCF are implemented using the NS-3 network simulator [20]. Simulations were performed over different vehicle densities for a period of 2000 s each. We considered a grid simulation topography similar to the part of Kingston city shown in Fig. 6, with one gateway deployed at each corner and a scale mapped to 1 km $\times$ 1 km area. The interest generation is uniformly distributed among the four gateways with the default injection rate equal to 1 every 20 s. In generating interests, four interest types and four targeted AoIs are considered, leading to 16 different interest tuples. The Simulation of Urban Mobility vehicular simulator [21], in conjunction with the mobility model generator for vehicular networks [22], are used to generate realistic mobility traces. The IEEE 802.11p WAVE standard is used for communication in the vehicular network with the beaconing interval set to 0.5 s and the transmission range set to 150 m.

In CADD, an RCS is added at each intersection. The parameters ρ_c and ρ_p are set to 250 s and the whole simulation time (2000 s), respectively. The α and β weights 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.

We consider two different cache sizes (the maximum number of cached packets at each RCS) to show the effect of the popularity-based cache replacement. Cache sizes of 5 (CADD-5) and 20 (CADD-20) are considered. In CADD-5, only the five most popular tuples are kept at RCS$_{\mathrm{max}}$, whereas less popular tuples are offloaded to their corresponding RCS$_{2\mathrm{max}}$. In CADD-20, an RCS can accommodate a replica of each possible interest.

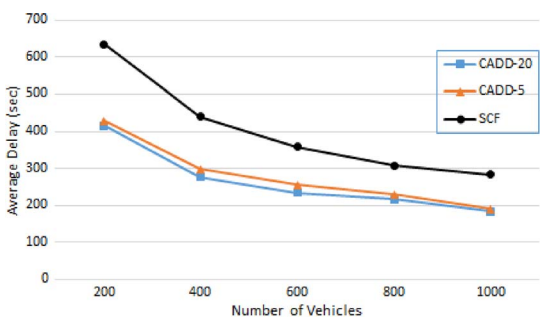
Table II summarizes the simulation parameters and the corresponding values.

B. Simulation Results and Analysis

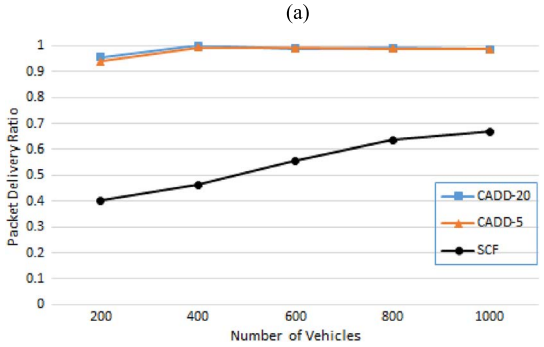
First, we compare CADD-5, CADD-20, and SCF in terms of the average round-trip delay over varying vehicular densities. As shown in Fig. 7(a), CADD decreases the delay compared

TABLE II
SIMULATION PARAMETERS AND CONFIGURATIONS

Simulation Parameter	Value
Number of vehicles	200, 400, 600, 800, 1000
Max. vehicle speed (Km/h)	40
Average road segment length (m)	300
Topography size	1km × 1km
Beaconing interval (sec)	0.5
Transmission range (m)	150
Communication technology	IEEE 802.11p
Number of interest types	4
Number of targeted areas of interest	4
ρ_c (sec)	250
ρ_p (sec)	2000
ρ_r (sec)	20, 60
α	0.2
β	0.8
RS	5, 20



(a)



(b)

Fig. 7. Comparison of CADD and SCF with varying densities. (a) Average delay of CADD and SCF with varying densities. (b) Packet delivery ratio of CADD and SCF with varying densities.

with SCF. This decrease is due to 1) the caching assistance feature that brought the replies closer to the requesters, saving the interests from having to go to their designated AoIs; and 2) the heading awareness feature that saved the interests and replies from going farther from their destinations. Comparing CADD-5 and CADD-20, we can notice a slight increase in the delay with reducing the cache size. Since our cache replacement mechanism gives a replaced packet another caching chance instead of dropping it, reducing the cache size has not greatly affected the delay as, still, there have been chances for cache hits for the replaced replicas.

We also consider a similar comparison in terms of the packet delivery ratio metric. As shown in Fig. 7(b), CADD significantly improves the delivery ratio in its two versions, namely, CADD-20 and CADD-5, compared with SCF. The main reason for such improvement is the heading awareness

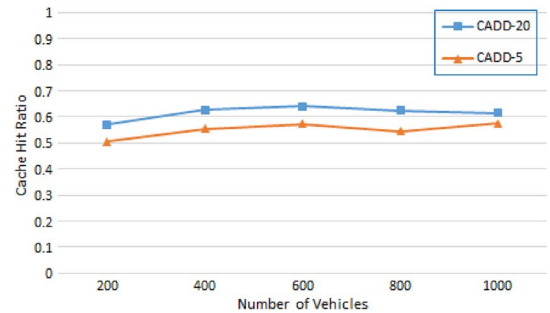


Fig. 8. Cache hit ratio of CADD with varying densities.

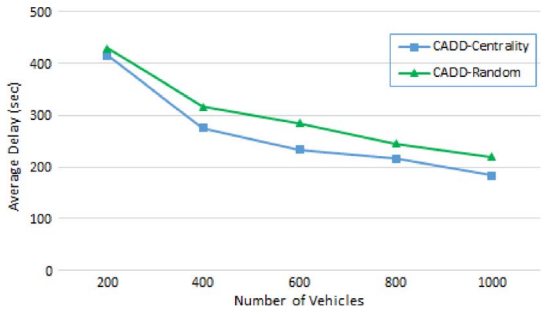
feature of CADD that saves packets from eventual dropping due to going away from their destinations.

Another main metric used to evaluate the performance of CADD is the cache hit ratio of the resolved interests. With caching assistance, CADD achieves significant savings in access cost through succeeding in achieving a high cache hit ratio, as shown in Fig. 8. For the same reason previously discussed, CADD-5 has a slight decrease in the hit ratio compared with CADD-20. Since SCF does not involve any caching assistance, the cache hit ratio does not apply.

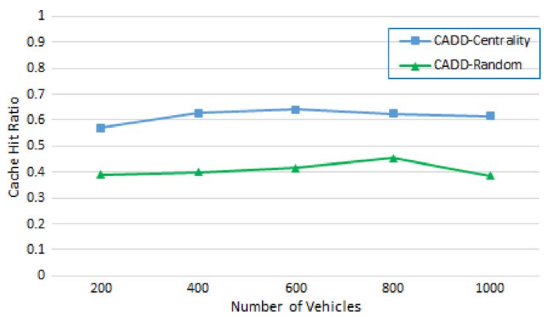
Second, we compare CADD with its underlying centrality-based caching mechanism (referring to it as CADD-Centrality) to another version with random caching (referring to it as CADD-Random) to show the effect of the proposed caching mechanism. In terms of the average round-trip delay, Fig. 9(a) shows that CADD-Centrality achieves lower delay than CADD-Random due to ensuring to have the cached replicas at the maximum central RCSs, which have higher probabilities for cache hits than randomly chosen RCSs. This is shown in Fig. 9(b), comparing CADD-Centrality and CADD-Random in terms of the cache hit ratio. We considered CADD-20 in the above comparison, as well as in the following.

Third, we compare CADD in its complete form (referring to it in this comparison as CADD-Caching) and SCF to another version of CADD with relaxing the caching feature (referring to it as CADD-NoCaching) to show the effect of the heading awareness feature independently. In terms of the average round-trip delay, as expected, Fig. 10(a) shows that CADD-NoCaching achieves a value between those achieved by CADD-Caching and SCF, highlighting the delay reduction achieved through taking vehicles' headings into consideration. Fig. 10(b) shows a similar comparison in terms of the packet delivery ratio, emphasizing significant improvements through CADD with and without caching compared with SCF due to the heading awareness feature.

Finally, we study the effect of changing the interest generation period (ρ_r) in CADD-20 and CADD-5, showing the results in Figs. 11 and 12, respectively, for varying vehicular densities. We considered values 20 and 60 for ρ_r (generating an interest every 20 and 60 s, respectively). In terms of the average delay, Figs. 11(a) and 12(a) show that with increasing ρ_r, the delay increases because with such an increase in ρ_r, some cached replicas expire before being asked for by later interests, leading to reduced cache hits and, hence, longer delay. Figs. 11(b) and 12(b) show such a decrease in the cache hit ratio with increasing

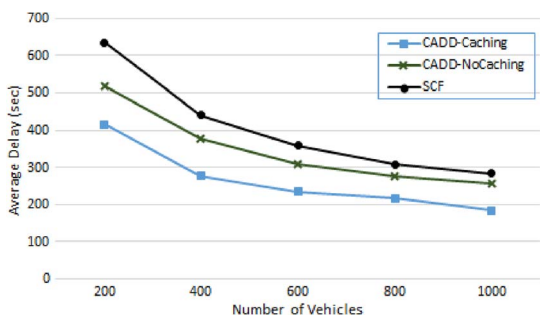


(a)

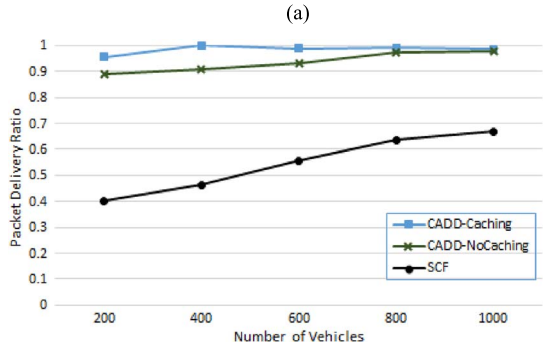


(b)

Fig. 9. Comparison of CADD with centrality-based caching versus random caching with varying densities. (a) Average delay of CADD-Centrality and CADD-Random with varying densities. (b) Cache hit ratio of CADD-Centrality and CADD-Random with varying densities.



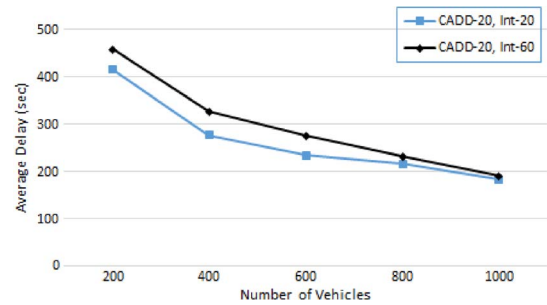
(a)



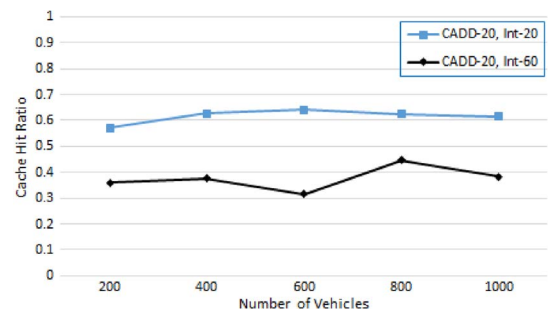
(b)

Fig. 10. Comparison of CADD with and without caching and SCF over varying densities. (a) Average delay of CADD with and without caching and SCF over varying densities. (b) Packet delivery ratio of CADD with and without caching and SCF over varying densities.

ρ_r. We can also see in Figs. 11 and 12 that the results, when ρ_r is equal to 60, is quite the same in CADD-5 and CADD-20 for the same reason previously mentioned i.e., the reduced cache hits, diminishing the effect of the cache size.

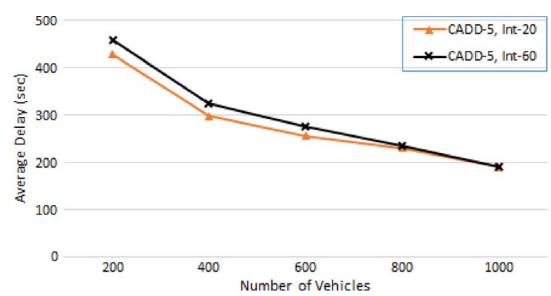


(a)

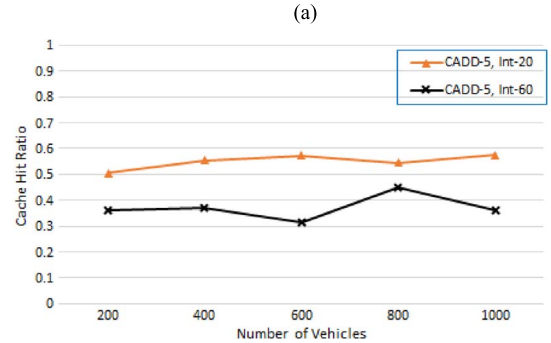


(b)

Fig. 11. Comparison of CADD-20 with different interest generation periods over varying densities. (a) Average delay of CADD-20 with different interest generation periods over varying densities. (b) Cache hit ratio of CADD-20 with different interest generation periods over varying densities.



(a)



(b)

Fig. 12. Comparison of CADD-5 with different interest generation periods over varying densities. (a) Average delay of CADD-5 with different interest generation periods over varying densities. (b) Cache hit ratio of CADD-5 with different interest generation periods over varying densities.

VI. Practical Considerations

Here, we highlight the practical considerations related to the operation and implementation of the CADD scheme and the deployment of the required components.

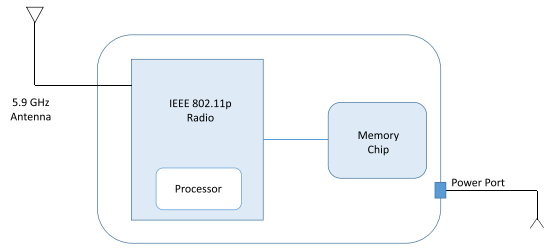


Fig. 13. Basic architecture of an RCS.

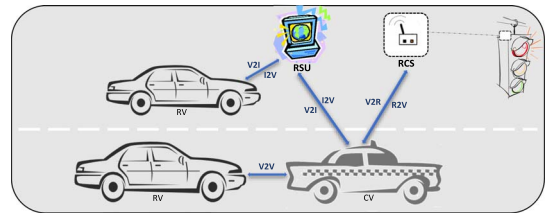


Fig. 14. Interoperability of CADD-enabled vehicles with regular vehicular networks.

A. Road Caching Spot (RCS)

Compared with the different RSU models that are on the market (e.g., the LOCOMATE devices that are used in the US pilot and test-field programs [23]), the proposed RCS is a more cost-effective solution for data delivery assistance as it only holds the components needed for the caching and multihop data delivery processes. The other RSU models include a variety of components that are not required for those processes, such as the Ethernet, M2M, and GPS modules. Our proposed RCS is a simple lightweight device that can be deployed on traffic lights or electric poles to complement RSUs for providing ubiquitous roadside assistance. It consists of an 802.11p radio for communication, which comes with an embedded processor, a memory chip for caching, and a power port. Fig. 13 shows the basic architecture of an RCS. It is worth noting that the RCS is not proposed to replace the RSU; it will be used to complement and assist the RSUs when/where their ubiquitous deployment is not feasible.

B. Communication Technologies

The IEEE 802.11p communication standard is suggested as the main communication technology between the RCSs and vehicles since each smart vehicle will be equipped by default with an 802.11p radio to support the intelligent transportation system services. Our proposed RCS is equipped with an 802.11p radio as well to support such a communication. However, the scheme operation is not confined only to the 802.11p use; other intervehicle communication technologies can be utilized without affecting the scheme. For example, some Zigbee modules [24], [25] are introduced to the market to support communication among vehicles and between vehicles and infrastructure. These modules can be easily attached to vehicles and road units. The visible light communication (VLC) technology is currently gaining interest as a candidate for vehicular communication [26], [27]. A vehicle's regular lights can be replaced with light-emitting diodes as VLC transmitters and each vehicle can be equipped as well with a VLC receiver that can be either an image sensor or a photo diode. To support such type of communication, our basic RCS can be modified to include a VLC transmitter and a receiver instead of the 802.11p radio.

It is worth mentioning that with the flexibility and ease of altering the communication interface of the RCSs, the penetration rate of the roadside assistance can be boosted through adjusting such interfaces to cope with whichever vehicular communication technology is available.

The gateways needed as the PoA to the vehicular network are equipped with two communication interfaces: 1) a broadband interface to be connected to the Internet for receiving and sending service requests and replies, respectively; and 2) a communication interface to communicate with the vehicular network supporting the same technology used by that vehicular network. A gateway either can have the same architecture as an RCS with an added capability for Internet connectivity or can be an RSU.

C. Backward Compatibility

Our scheme depends mainly on smart vehicles as carriers of interest and reply packets between RCSs and as the main resources of the sensing-based data; however, the entities involved in the scheme can be adjusted to support backward compatibility with regular/legacy vehicles having no on-board connectivity. Owners of legacy vehicles who are willing to participate in public sensing services supported by the proposed scheme can equip their vehicles with an 802.11p radio to be engaged in the forwarding loop. The owner of these vehicles can depend on the sensing resources of smartphones for potential tasks (e.g., depending on a smartphone camera for monitoring an event on the road). For the communication between the smartphone and the outside vehicular network (for getting sensing interests and replying with sensed data), a Bluetooth interface can be integrated with a standalone 802.11p radio for establishing an in-vehicle communication between the smartphone and the added communication module that is responsible for the out-of-vehicle communication.

We also highlight that the operation of the scheme does not require that all vehicles should be CADD enabled. Regular communication-enabled vehicles can also be involved in the system for basic data relaying through their default data forwarding capabilities and vehicular communication. The CADD-enabled vehicles (CVs) support interoperability with such regular vehicles (RVs), as well as communication with on-road RSUs and RCSs (see Fig. 14). RVs can be involved in regular vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-vehicle (I2V) communications. CVs can support all the aforementioned types of communication in addition to communication with RCSs, which are denoted as V2R and R2V communications.

For the backward compatibility of roadside assistance, as previously mentioned, the RCS is proposed to complement the RSU operation and not to replace it. For an intersection with a deployed RSU, the scheme does not require an RCS to be deployed; only the storage capacity of the deployed RSU can be upgraded to support caching.

TABLE III
COMMUNICATION OVERHEAD FOR RESOLVING AN INTEREST PACKET

Packet Type	Number of Generated Packets by CADD	
Control Packets	0	
Data Packets	1	in case of a cache hit
	2	otherwise

D. Distributed Storage and Processing

Currently, there is a great push toward the adoption of fog/edge computing [28], [29] and cloudlets [30], [31], aiming at providing local distributed computing, as opposed to cloud central computing to handle the torrent of data being generated by the Internet of things [32], [33]. Such trends move computing to a virtualized layer between the users and traditional cloud computing platforms, bringing content closer to consumers of local interest. Following these hot trends, our scheme builds on distributed storage and processing through RCSs deployed in a distributed fashion. In our scheme, data of interest are cached and processed locally, relaxing the need for central/cloud computing for the sake of an efficient use of the broadband bandwidth, and a fault-tolerant and load-balanced caching of the generated data, avoiding directing the caching load to one central node.

E. Advantageous Delivery Features

The proposed scheme works on utilizing the current state of vehicular density efficiently for data delivery. In cases of congested areas, the scheme utilizes the dense availability of vehicles for faster multihopping, even if the carrying vehicles are quasi-stationary, whereas in sparse environments, the scheme depends on moving vehicles and their heading for bringing packets closer to/from AoIs. Therefore, the proposed scheme can successfully operate in both cases, considering its adaptive delivery features.

F. Performance Overhead

The CADD scheme works on achieving its functionalities without imposing a communication overhead on the network and its involved entities. Vehicles and RCSs use the IEEE 802.11p WAVE standard in which, by default, the nodes periodically broadcast beacon packets to exchange ID, position, and velocity information. Our scheme mainly utilizes such exchanged beacon packets for performing geographical data forwarding without the need to exchange special control packets. In terms of the data packets, for every interest packet sent, the scheme generates a maximum of two reply packets (see Table III). If the interest packet does not meet a cached replica on its way toward the corresponding AoI, it would be resolved by a vehicle at the AoI sending a data reply packet to the requesting gateway. In addition, a replica of this reply would be generated and directed toward $\mathrm{RCS}_{\max}$ encountered on the interest path. On the other hand, when an interest encounters a cache hit at one of the traversed RCSs, only one packet would be generated, which is a copy of the cached data.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed the CADD scheme that enhances access to vehicular resources for vehicle-based public sensing services. Through applying caching on the data delivery path, CADD aims at reducing the cost and delay of accessing vehicular resources. CADD relies on the deployment of a lightweight RCS at each intersection, and vehicles work as carriers of both interests and replies between the RCSs. As part of CADD, a novel centrality-based caching mechanism was proposed, which handles the dynamic nature of vehicular networks and considers popularity in cache replacement. CADD considers vehicles' headings to direct interests and replies toward their destinations. We presented mathematical analysis of the scheme through assessment models comparing it to another scheme with connected RSUs in addition to analyzing its estimated delay to reach an AoI from a considered gateway. Simulation-based performance evaluation was conducted as well, showing that CADD achieves significant improvements in terms of access cost, delivery delay, and packet delivery ratio, compared with another scheme that does not involve caching assistance and does not take vehicles' headings into consideration. In our future work, we will consider having a targeted cache hit ratio and study the placement and distribution of RCSs to achieve this ratio.

ACKNOWLEDGMENT

The authors would like to thank H. Farahat for his help with the NS-3 simulation.

REFERENCES

[1] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," *IEEE Netw. Mag.*, vol. 29, no. 1, pp. 12–17, Jan./Feb. 2015.
[2] A. T. Campbell *et al.*, "The rise of people-centric sensing," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 12–21, Jul./Aug. 2008.
[3] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a mobile sensor," *Procedia Comput. Sci.*, vol. 34, pp. 286–295, 2014.
[4] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Comput. Netw.*, vol. 57, no. 16, pp. 3128–3141, Nov. 2013.
[5] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in information-centric networks," in *Proc. NETWORKING*, Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2012, pp. 27–40.
[6] S. Abdel Hamid, M. Abu-Elkheir, H. S. Hassanein, and G. Takahara, "Towards provisioning vehicle-based rural information services," in *Proc. IEEE 37th Conf. Local Comput. Netw. Workshops*, 2012, pp. 835–842.
[7] S. Abdelhamid, H. S. Hassanein, G. Takahara, and H. Farahat, "Caching-assisted access for vehicular resources," in *Proc. IEEE Conf. LCN*, Sep. 2014, pp. 28–36.
[8] S. Abdel Hamid, H. S. Hassanein, and G. Takahara, *Routing for Wireless Multi-Hop Networks*, ser. Springer Briefs in Computer Science. New York, NY, USA: Springer-Verlag, 2013.
[9] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Veh. Technol. Mag.*, vol. 2, no. 2, pp. 12–22, Jun. 2007.
[10] K. C. Lee, U. Lee, and M. Gerla, "Survey of routing protocols in vehicular ad hoc networks," in *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*. Hershey, PA, USA: IGI Global, 2010, pp. 149–170.
[11] Y. Ding and L. Xiao, "SADV: Static-node-assisted adaptive data dissemination in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2445–2455, Jun. 2010.
[12] D. Borsetti and J. Gozalvez, "Infrastructure-assisted geo-routing for cooperative vehicular networks," in *Proc. IEEE VNC*, 2010, pp. 255–262.
[13] Y. Wu, Y. Zhu, and B. Li, "Infrastructure-assisted routing in vehicular networks," in *Proc. 31st IEEE INFOCOM*, 2012, pp. 1485–1493.

[14] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode, "VITP: An information transfer protocol for vehicular computing," in *Proc. 2nd ACM Int. Workshop Veh. Ad Hoc Netw.*, 2005, pp. 30–39.
[15] C. E. Palazzi, F. Pezzoni, and P. M. Ruiz, "Delay-bounded data gathering in urban vehicular sensor networks," *Pervasive Mobile Comput.*, vol. 8, no. 2, pp. 180–193, Apr. 2012.
[16] W. Ali, S. M. Shamsuddin, and A. S. Ismail, "A survey of web caching and prefetching," *Int. J. Adv. Soft Comput. Appl.*, vol. 3, no. 1, pp. 18–44, 2011.
[17] N. Loulloudes, G. Pallis, and M. D. Dikaiakos, "Caching dynamic information in vehicular ad hoc networks," in *Euro-Par 2010-Parallel Processing*. Berlin, Germany: Springer-Verlag, 2010, pp. 516–527.
[18] M. Amadeo, C. Campolo, and A. Molinaro, "CRoWN: Content-centric networking in vehicular ad hoc networks," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1380–1383, Sep. 2012.
[19] V. Jacobson *et al.*, "Networking named content," in *Proc. 5th Int. Conf. Emerging Netw. Exp. Technol.*, 2009, pp. 1–12.
[20] The NS-3 Network Simulator. Accessed: Sep. 15, 2015. [Online]. Available: https://www.nsnam.org/
[21] Simulation Of Urban Mobility. Accessed: Sep. 9, 2015. [Online]. Available: http://sumo-sim.org/
[22] Rapid Generation of Realistic Simulation for VANET. Accessed: Sep. 9, 2015. [Online]. Available: http://lens1.csie.ncku.edu.tw/Joomla_Version/index.php/research-projects/past/18-rapid-vanet/
[23] Arada Systems. Accessed: Sep. 9, 2015. [Online]. Available: http://www.aradasystems.com/
[24] n-Core Sirius OBD. Accessed: Oct. 10, 2015. [Online]. Available: http://www.nebusens.com/index.php/en/products/n-core/hardware
[25] OBD2 Fleet Management System Vehicle Telematics Monitor. Accessed: Sep. 16, 2015. [Online]. Available: http://www.fleet-genius.com/obd2-fleet-vehicle-telematics-monitors
[26] C. B. Liu, B. Sadeghi, and E. W. Knightly, "Enabling Vehicular Visible Light Communication (V2LC) networks," in *Proc. 8th ACM Int. Workshop VANET*, Sep. 2011, pp. 41–50.
[27] LEDs Bring New Light to Car-to-Car Communication, 2014. Accessed: Sep. 16, 2015. [Online]. Available: http://spectrum.ieee.org/transportation/advanced-cars/leds-bring-new-light-to-car-to-car-communication
[28] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of things," in *Proc. 1st Ed. ACM MCC Workshop MCC*, 2012, pp. 13–16.
[29] Fog Computing—Cisco Systems. Accessed: Sep. 9, 2015. [Online]. Available: https://techradar.cisco.com/trends/Fog-Computing
[30] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proc. 3rd ACM Workshop MCS*, 2012, pp. 29–36.
[31] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct.–Dec. 2009.
[32] L. Atzori, A. Iera, and G. Morabito, "The Internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[33] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

Sherin Abdelhamid (S'12–M'15) received the B.Sc. and M.Sc. degrees in computer sciences from Ain Shams University, Cairo, Egypt, in 2005 and 2009, respectively, and the Ph.D. degree from Queen's University, Kingston, ON, Canada, in 2014.

She is a Postdoctoral Fellow and a Research Associate with the Telecommunications Research Laboratory, School of Computing, Queen's University. She has several publications in top venues. Her research interests include intelligent transportation systems, vehicular and wireless multihop networks, vehicular cloud computing, information-centric networks, crowdsourcing, and Internet of things.

Dr. Abdelhamid received the Best Paper Award from the Ad Hoc and Sensor Networking Symposium of the 2013 IEEE Global Communications Conference.

Hossam S. Hassanein (S'86–M'90–SM'06) received the B.Sc. degree in electrical engineering from Kuwait University, Kuwait City, Kuwait, in 1984; the M.A.Sc. degree in computer engineering from the University of Toronto, Toronto, ON, Canada, in 1986; and the Ph.D. degree in computing science from the University of Alberta, Edmonton, AB, Canada, in 1990.

He is the Founder and Director of the Telecommunications Research Laboratory, School of Computing, Queen's University, Kingston, ON, Canada. He is a leading authority in the areas of broadband, wireless, and mobile networks architecture, protocols, and performance evaluation. His record spans more than 500 publications, in addition to numerous keynotes and plenary talks at flagship venues.

Dr. Hassanein has received several recognitions and Best Paper Awards at top international conferences. He is a former Chair of the IEEE Communication Society Technical Committee on Ad Hoc and Sensor Networks. He is an IEEE Communications Society Distinguished Speaker.

Glen Takahara (SM'15) received the B.A. degree in mathematics from the University of British Columbia, Vancouver, BC, Canada, in 1988 and the M.S. and Ph.D. degrees in statistics from Carnegie Mellon University, Pittsburgh, PA, USA, in 1990 and 1994, respectively.

He is an Associate Professor in mathematics and engineering with the Department of Mathematics and Statistics, Queen's University, Kingston, ON, Canada. His research interests include communications theory, networking, applied probability, and statistics.