Computer Communications 35 (2012) 380–391

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Cache invalidation strategies for Internet-based vehicular ad hoc networks

Sunho Lim [a,*], Chansu Yu [b,d], Chita R. Das [c]

[a] Dept. of Computer Science, Texas Tech University, Lubbock, TX 79409, USA
[b] Dept. of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH 44115, USA
[c] Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA
[d] WCU Division of IT Convergence Engineering, POSTECH, Pohang, S. Korea

ARTICLE INFO

Article history:
Received 21 January 2010
Received in revised form 11 October 2011
Accepted 30 October 2011
Available online 6 November 2011

Keywords:
Cache invalidation
Data access
Internet-based vehicular ad hoc networks

ABSTRACT

Internet-based vehicular ad hoc network (IVANET) is an emerging technique that combines a wired Internet and a vehicular ad hoc network (VANET) for developing an ubiquitous communication infrastructure and improving universal information and service accessibility. A key design optimization technique in IVANETs is to cache the frequently accessed data items in a local storage of vehicles. Since vehicles are not critically limited by the storage/memory space and power consumption, selecting proper data items for caching is not very critical. Rather, an important design issue is how to keep the cached copies valid when the original data items are updated. This is essential to provide fast access to valid data for fast moving vehicles. In this paper, we propose a *cooperative cache invalidation* (CCI) scheme and its enhancement (ECCI) that take advantage of the underlying location management scheme to reduce the number of broadcast operations and the corresponding query delay. We develop an analytical model for CCI and ECCI techniques for fasthand estimate of performance trends and critical design parameters. Then, we modify two prior cache invalidation techniques to work in IVANETs: a poll-each-read (PER) scheme, and an extended asynchronous (EAS) scheme. We compare the performance of four cache invalidation schemes as a function of query interval, cache update interval, and data size through extensive simulation. Our simulation results indicate that the proposed schemes can reduce the query delay up to 69% and increase the cache hit rate up to 57%, and have the lowest communication overhead compared to the prior PER and EAS schemes.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

A vehicular ad hoc network (VANET) consists of a set of high-speed mobile vehicles equipped with communication facilities [1]. It supports inter-vehicle communications through a multi-hop message relay without the assistance of any fixed infrastructure. A variety of applications built for VANETs will be available in the near future as drivers request for more information regarding, for example, safety and traffic conditions. Drivers can monitor/share real-time traffic conditions, exchange files, and send/receive emergency warning messages so that they can pro-actively avoid accidents such as intersection collisions or chained collisions [2,3]. This requires that drivers should be able to access the Internet service and information on wheel [4] through a roadside unit (e.g. an access point (AP), an infostation [5], or a message relay box [6]) located along the road that acts as a gateway to an infrastructure network.

Therefore, in order to provide a flexible connectivity, accessibility, and a rich set of services, it is imperative to consider the integration of a VANET with a wireless infrastructure, such as a wireless local area network (e.g. IEEE 802.11) and a wireless wide area network (e.g. 3G). It is envisaged that such an *Internet-based vehicular ad hoc network*, or IVANET, will prevail to become an ubiquitous communication infrastructure in the near future.

A key optimization technique in improving the communication performance of IVANETs is to cache the frequently accessed data items in a local storage. In an IVANET, it is less of a problem to determine which data items to cache because the storage space in a vehicle is not critically limited. However, a critical design issue is the *cache invalidation* scheme, since applications require data consistency with the server to avoid using any stale data. When a data item in a server is updated, it is necessary to invalidate the cached copies of this data item by broadcasting an *invalidation report* (IR). However, due to fast roaming vehicles, cache invalidation schemes developed for cellular networks and mobile ad hoc networks (MANETs) may not work well. Unlike these conventional networks, energy conservation is not an issue in IVANETs because a vehicle is supported by its own built-in battery. Rather, our concerns are query delay and cache hit ratio in the presence of mobility and frequent data updates.

* Corresponding author. Tel.: +1 806 742 3527.
E-mail addresses: sunho.lim@ttu.edu (S. Lim), c.yu91@csuohio.edu (C. Yu), das@cse.psu.edu (C.R. Das).

0140-3664/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2011.10.013

The following observations characterize the IVANETs in the context of *cache invalidation* and support our research motivation.

- Due to high-speed mobility, vehicles reside in a coverage area for a short period of time. For example, the connection time within a coverage area ranges from 5 to 24 s at city driving [7]. Therefore, when a data server broadcasts an IR, it is not straightforward which coverage area (s) to target. And, since multiple coverage areas are involved in a broadcast operation, the cost of broadcasting IRs becomes non-negligible.
- Since it is unlikely that adjacent vehicles have common data items in a real IVANET environment, it is wasteful to broadcast the same content of IRs to different vehicles because most of contents may not be relevant to the vehicles.
- It is possible to have a proxy server between a data server and vehicles to reduce the invalidation cost. However, in order to support a scalable invalidation operation, it is essential for a data server or a proxy server to coordinate with location management agents, which increases complexity.

As compared to MANET, the cache invalidation problem in an IVANET is unique because of vehicles' high-speed mobility and requirement of scalable cache invalidation operation.

To address these issues, we propose and analyze a *cooperative cache invalidation* (CCI) scheme and its enhancement, called *enhanced CCI* (ECCI), which can effectively deal with fast moving vehicles without incurring significant overhead. Our contributions are the following:

- First, we suggest a hierarchical network model for IVANETs that consists of access points (APs), gateway foreign agents (GFAs), home agents (HAs), and data server (s). This model facilitates scalable cache invalidation operations as well as efficient location management by having the GFAs to take care of micro-mobility and micro-invalidation in their respective local areas.
- Second, we propose a cooperative approach, where both a server and location management agents coordinate for cache invalidation. By maintaining a list of data items and the access history by vehicles, a server asynchronously sends an IR to a HA rather than blindly broadcasts to vehicles. Then the HA judiciously refines and distributes the IR to appropriate GFAs, which can answer the validity of the queried data item to reduce the unavoidable query delay, witnessed in most IR-based techniques.
- Third, we develop an analytical model for predicting the performance trends of the proposed two cache invalidation schemes.
- Fourth, we modify two previous cache invalidation schemes, poll-each-read (PER) and extended asynchronous (EAS), to work in IVANETs and compare them against our proposed schemes.

We conduct simulation-based performance evaluation of the four invalidation techniques along with the base case *no-cache* model to observe the impact of query interval, cache update interval, and data size on system performance, and communication overhead. According to the simulation results, the proposed schemes reduce the query delay up to 69%, increase the cache hit rate up to 57%, and incur lower communication overheads compared to two previous schemes. Overall, our study indicate that the proposed schemes provide better performance at reduced overhead and thus, are a viable approach for adopting in IVANETs.

The rest of paper is organized as follows. The prior work is reviewed and analyzed in Sections 2 and 3, respectively. The proposed schemes and their analytical models are presented in Sections 4 and 5, respectively. Section 6 is devoted to performance evaluation, comparison, and discussion. Finally, we further exploit several research issues for future directions and conclude the paper in Sections 7 and 8, respectively.

2. Background and related work

Most of cache invalidation schemes deploy one or combination of following design aspects: whether to maintain the cache status or not, who initiates cache validity, and level of cache consistency. First, a server may or may not maintain any cache status. In a stateful approach [8–11], a server keeps track of which mobile node caches which data item. Whenever a data item is updated, the server broadcasts an IR to inform the mobile nodes of the cache update. In a stateless approach [12–14], however, the server is not aware of any cache status and thus, it periodically broadcasts an IR including an update history. Depending on the IR size and broadcast interval, there is a tradeoff between the efficiency of cache invalidation and query latency.

Second, the push/pull/hybrid techniques specify who initiates cache validity. In the push approach [15], the server initiates the cache validity, for example, by broadcasting an IR. In the pull approach [15], however, whenever a query is generated which can be answered from a cached data item, a query request packet is sent to the server to verify the validity of the cached data item before it answers the query. Several push- and pull-based schemes [16] and combining both schemes [17,18] have been proposed considering the tradeoffs between query delay and communication overhead.

Third, depending on a consistency level of cached data items with the server, there could be strong/weak/hybrid consistencies. In the strong consistency [12,14,15], a query is answered by the latest updated data item from either a cache or the server. In the weak consistency [19], however, it needs not reflect the current update of a data item. Thus, it may reduce the communication cost and query delay, but increase the communication overhead due to accessing stale data items. To balance strong and weak consistencies in terms of scalability, complexity, and communication cost, a hybrid cache consistency scheme [17] is proposed, where mobile nodes can flexibly specify their consistency requirements for different types of data items.

In addition, there have been many research efforts on developing various caching schemes for MANETs [18,20–25] and Internet-based MANETs, called IMANETs [16,17], where MANET technologies are combined with an Internet to provide a flexible information accessibility and availability for users. For MANETs, Hara and Madria suggested a replication scheme for periodically updated data items based on the previously suggested replica allocation method [20]. Replicated data items are periodically re-allocated depending on the access frequency, remaining time to next update, and network topology. However, estimation of optimal reallocation period and remaining time to the next update are not practically feasible. There are few invalidation approaches [16,17] for IMANETs. In [16], several push and pull-based cache invalidation schemes are proposed based on the aggregate cache [26], in which the aggregated local caches of the nodes can be considered as an unified large cache for the IMANET.

In summary, little effort has been devoted in exploring the cache invalidation issue for IVANETs, where the high-speed mobility and scalability algorithm are primary concerns.

3. Revisit cache invalidation strategies

In this section, we analyze the prior cache invalidation strategies in terms of single- and multi-cell based designs, and formalize the problem in an IVANET environment.

3.1. Single-cell based approach

Most of the prior IR-based schemes are variation of the timestamp (TS) scheme [12], where a base station (BS) broadcasts an

IR every L seconds. Since the IR is periodically broadcasted, there is an unavoidable delay before answering a query. Depending on the length of broadcast interval, L, there is a tradeoff between the communication overhead of cache invalidation and query latency. Although an updated invalidation report (UIR) [14] is proposed to further reduce the query latency by adding updated items between IRs, query delay of half of the broadcast interval $(\frac{L}{2})$ still exists. Since the TS scheme does not consider mobility, it cannot be directly deployed in IVANETS. For example, when a vehicle generates a query and waits for an IR, it may miss the IR if it has moved to the next adjacent cell. To verify this, we experimented with a single cell environment, where 20 vehicles are randomly located and moved with a given velocity. In Fig. 1(a), as the velocity increases, more vehicles handoff before receiving the IR and thus, the IR miss ratio increases which is sensitive to L.

3.2. Multi-cell based approach

To reduce the IR miss ratio due to mobility, multi-cell based designs extended from the asynchronous stateful (AS) scheme [8] are proposed, where a network component (e.g. mobile switching center (MSC) or server) located in a higher network hierarchy than a BS executes cache invalidation operations. In [9], when a mobile node handoffs, it either simply drops the cached data items or checks them with the server for validity. Since the server maintains a list of which data items are accessed by which mobile nodes, whenever a data item is updated, it pro-actively unicasts IRs into multiple cells, where the updated data item is cached. In VANETS, however, vehicles stay in a cell for a short period of time and travel along the large number of cells and thus, it is wasteful to examine cached data items with the server for validity upon every handoff. Also, the server's pro-active IR transmission to multiple cells is neither a scalable solution nor efficient, if the updated data item is not queried.

To verify this, we experiment with a multi-cell environment, where 50 vehicles are randomly located in five cells and move under a wrap around one-dimensional network topology. We compare two schemes extended from [12,9], namely an extended TS (ETS) scheme and an extended AS (EAS) scheme. In ETS, a stateless server broadcasts an IR to every cell in the system whenever vehicles cache an updated data item. In the EAS scheme, however, a stateful server selectively broadcasts an IR to the cell(s). Also, upon handoff, vehicles send all the valid cached data items' *ids* to the server for validity and receive an invalidation check packet, which is counted as the IR traffic. In Fig. 1(b), when the cache update interval (T_u) is 10 (sec), due to the IR broadcasts to all the cells, the ETS scheme shows the same amount of IR traffics and the impact of velocity on the IR traffic is minimized. When T_u is 100 (sec), the IR traffic reduces. However, the ETS scheme is not scalable in terms of the cache update rate and network size. In the EAS scheme, as the velocity increases, it shows very high IR traffic regardless of cache update interval because of additional invalidation check packets upon every handoff (here, the mean query interval (T_q) is set 50 (sec)).

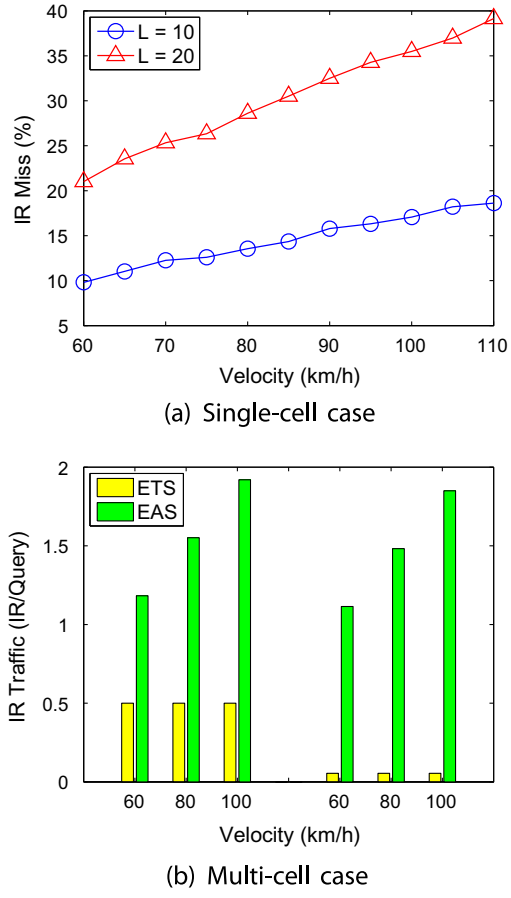

Fig. 1. Effect of velocity (L = 10 or 20 (sec) [14,9]): (a) IR miss ratio (b) Average IR traffic per query (T_q = 50 (sec), and T_u = 10 (Subfigure (a)) or 100 (sec) (Subfigure (b))).

3.3. Summary

Although the ETS scheme reduces the impact of mobility, there still exists a non-negligible query delay and the stateless server blindly broadcasts an IR to entire cells in the system, resulting in a scalability issue. On the other hand, the EAS scheme has no query delay when a queried data item is cached and marked as valid, but it incurs heavy IR traffic mainly due to mobility. Thus, reduction of the mobility impact and IR traffic are conflicting requirements, and optimization of both is extremely complex. In light on this, a server is required to keep track of a vehicle's where abouts under the coordination of other network agents not only to reduce the impact of mobility and IR traffic, but also to provide a scalable solution in IVANETS.

4. The proposed cache invalidation strategy

In this section, first we briefly introduce a system model and then present our cache invalidation technique.

4.1. System model

As illustrated in Fig. 2, a hierarchical network model for IVANET is motivated by a cellular network, and it consists of access points (APs), gateway foreign agents (GFAs), home agents (HAs), and servers of data item source. First, vehicles are equipped with communication facilities such as an IEEE 802.11-based dedicated short range communication (DSRC) transceiver [1]. Thus, they can either communicate with other vehicles or connect to the Internet flexibly. Since providing a location tracking capability in vehicles is becoming popular, we assume that vehicles are aware of their current location. Also, vehicles have a built-in navigation system via GPS, in which a digital map is loaded to show roads around current location and direction, the shortest path to the destination, traffic conditions, location-dependent information, and other information. In addition, unlike cellular and MANET environments,

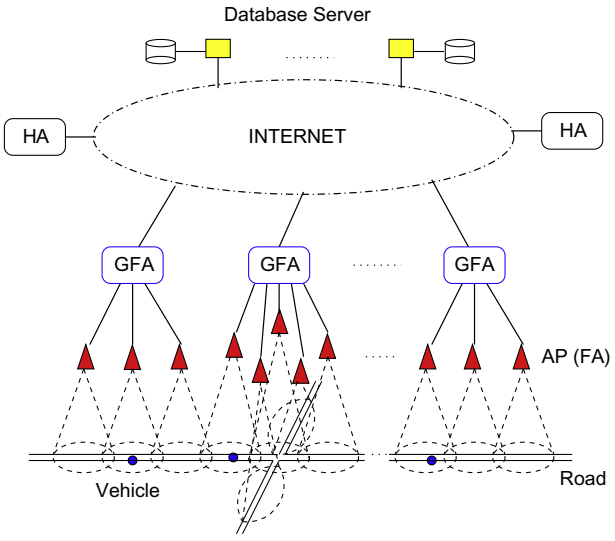

Fig. 2. A system model of IVANET.

where nodes move without restrictions, vehicles are bounded to the underlying fixed roads with speed limits and traffic lights in VANETs.

Second, in order to support global wireless access of an Internet, a mobile IP [27] is deployed, in which vehicles are able to access the Internet and maintain their on-going communications, while they move at a high-speed. When a vehicle accesses the Internet, it searches the nearest located AP, which is a gateway to the Internet, to access any information and service. An AP is located in a communication area and can be mounted on the top of a signal light or a road lamp along the road. Also, the AP plays an additional role as a foreign agent (FA). Thus, when a vehicle moves from one coverage area to another, it performs a location update/register operation that involves an AP, a GFA, and an HA. To avoid frequent location register/update operations for fast roaming vehicles, we use an IP micro-mobility support, in which vehicles report the HA only when a major change occurs such as changes to a regional network. In this paper, a 2-way handshake [28] is used to implement location updates by exchanging either a home or regional request/reply packet between a vehicle and a GFA (or HA) through an AP (or GFA).

Third, a database may be attached to any components in the proposed system, e.g. an AP, a GFA, an HA or a database server. However, due to the consistency issue among replicas stored in multiple databases, we assume that the database is attached to a database server. We also assume that the database consists of a total of D data items. A data item (d_x) is the basic unit of an update or a query operation, where $1 \leqslant x \leqslant D$. The database is only updated by the server, while a query is generated by the vehicles for read-only requests.

4.2. The cooperative approach

In this paper, we propose a *cooperative stateful* approach, where a server and network agents of location management coordinate together for cache invalidation operations. The server and network agents maintain a list of which data item is accessed by which vehicle and vehicles' current locations, respectively. Unlike most of the prior stateful designs, the server does not keep track of a vehicle's current location for IR broadcast. Thus, whenever a data item is updated, it sends an IR to an HA rather than blindly broadcasting the IR to multiple cells, where the updated data item is cached in a vehicle. Then the HA judiciously refines and re-distributes the IR to appropriate GFA (s), where they can answer the validity of a queried data item, requested from a vehicle. Here, GFAs do not pro-actively send the IR to the individual vehicles, but reply a vehicle's validity request by a on-demand-basis.

The rationale behind this approach is the following: (i) Since a vehicle's location is implicitly maintained by both a GFA and an HA, the server can avoid keeping track of the vehicles' current location, which is in fact a duplicated operation with the location management; (ii) Also, since a vehicle has a low probability of finding common data items in adjacent vehicles, broadcasting the same IR to different vehicles is inefficient in IVANETs; and (iii) In addition, since the query rate of vehicles is independent of the cache update rate of the server, whenever a data item is updated, a blind IR broadcast incurs heavy IR traffic. The proposed cache invalidation mechanism avoids these issues.

4.3. Proposed cache invalidation technique

In this subsection, we describe the detail operation of the cooperative cache invalidation (CCI) scheme and its enhanced scheme (ECCI). First, a server maintains a list of which data item ($d_x, x \in D$) is accessed by which vehicle in a registry (r_s), $[id(d_x), [vid(v_{x,y}), t_{x,y}]]$, where $v_{x,y} = \{v_y |$ request $(d_x) \wedge (y \in N)\}$, vid is a vehicle's id, and $t_{x,y}$ is the access time of d_x by v_y. Here, D and N are the number of data items and vehicles, respectively. Whenever a data item is updated, the server generates an IR $(\mathcal{R}_s)$, $< id(d_x), vid(v_{x,y}), t_{cur} >$, and it is not directly broadcasted to the vehicles but sent to an HA. Here, t_{cur} is the current timestamp. Upon receiving the $\mathcal{R}_s$, since the HA maintains location information about which vehicle is currently roaming under which GFA (g_i) in a registry (r_h), $[vid(v_y), g_i]$, it compares the v_y with $\mathcal{R}_s$ and r_h. Then the HA creates a new IR $(\mathcal{R}_h)$, $< id(d_x), t_{cur} >$, based on the matched g_i and sends it to the appropriate GFA. Depending on the number of matched GFAs, more than one $\mathcal{R}_h$ can be created. Also, each $\mathcal{R}_h$ may have a different size that is less than or equal to the original $\mathcal{R}_s$ received from the server. Upon receiving the $\mathcal{R}_h$, the GFA updates its registry and thus, it has the most recent updated IR with the server.

Second, when a vehicle (v_y) initially connects to a server, it sends all the *ids* of cached data items (d_k^c) and their timestamps, $[vid(v_y), id(d_k^c), t_k]$, and then the server updates its registry. Here, k is $1 \leqslant k \leqslant c$ and c is the total number of cache slots. Whenever the server receives a request packet for validating or accessing a data item, it also updates its registry. Also, when a vehicle handoffs to the coverage area which is under a different GFA, then the HA updates its registry and forwards the $\mathcal{R}_h$ to the new GFA when a cached data item is updated in the vehicle. The $\mathcal{R}_h$ may be forwarded to the appropriate APs to further reduce the query latency, but it is an overhead for the GFA, if a vehicle frequently moves among the coverage areas.

According to the proposed cache invalidation mechanism, let us examine the detail operation of a query request. First, when a vehicle (v_y) generates a query for a data item (d_x) which is not cached, it sends a request packet for accessing the data item to the server. The packet contains the vehicle's care-of address (CoA), an *id* of the requested data item, and a single bit flag (f) representing whether the queried data item is cached or not, $[CoA_y, id(d_x), f_x]$, and it is sent to the nearest AP. Here, f_x is set to 0. As the packet is forwarded to the server through the AP and GFA, CoAs of the AP and GFA are appended in the packet header, $[CoA_{GFA}, [CoA_{AP}, [CoA_y, id(d_x), f_x]]]$, to keep the route information. Upon receiving the request packet, the server attaches the queried data item to the ack packet with the route information. The ack packet is replied back to the AP

via the GFA. Then the AP unicasts the packet and thus, the corresponding vehicle can receive the data item.

Algorithm 1 Vehicle side

1: (**A**) When v_y generates a query for d_x:
2: $f_x \leftarrow 0$;
3: **if** $d_x \in L_y$ **then**
4: $f_x \leftarrow 1$;
5: attach $[f_x, t_x^c]$ to a request packet;
6: **else**
7: attach f_x to a request packet;
8: **end if**
9: append $[CoA_y, id(d_x)]$ to the request packet;
10: send the request packet to the nearest AP;
11: (**B**) When v_y receives an ack packet:
12: **if** d_x^u **then**
13: cache $d_x \leftarrow d_x^u$ in L_i;
14: **end if**
15: save $t_x^c \leftarrow t_{cur}$ in L_i;
16: answer the query;

Second, when a query is generated which can be answered by a cached copy of data item, the vehicle sends a request packet to the GFA for checking validity through the nearest AP. The GFA compares the *id* of queried data item with the $\mathcal{R}_h$ and replies an ack packet, if the cached copy is valid. If the cached copy is not valid, then the GFA forwards the packet to the server. When the server receives the request packet, it verifies the status of the queried data item and replies an ack packet with the route information, if it is a valid copy. If not, it uses the same procedure as the first case to supply the data item. The pseudo codes for the proposed scheme are presented in Algorithms 1–3, where L_y is a local cache of v_y.

Algorithm 2 Server side

1: (**A**) When a server receives a request packet for d_x from v_y:
2: **if** $f_x \wedge (t_x^c \geqslant t_x^u)$
3: reply an ack packet to v_y;
4: **else**
5: attach d_x^u to an ack packet and reply it to v_y;
6: **end if**
7: update R_s, $[id(d_x), [vid(v_{x,y}), t_{x,y}]]$;
8: (**B**) When d_x is updated:
9: send an IR, $<id(d_x), vid(v_{x,y}), t_{cur}>$, to an HA;

Third, to further enhance the CCI scheme, called *enhanced CCI* (ECCI), the GFAs cache data items in their local storage for future query requests from vehicles. Since an updated data item attached to an *ack* packet is replied back to vehicles from the server, a GFA intercepts the packet and caches the updated data item. Based on the IR received from the server through HA, GFAs also can maintain a set of valid data items and reply *ack* packets to vehicles directly, if the queried data item is cached, without forwarding the query request to the server.

In summary, based on the cooperative approach, the server and network agents of the underlying location management coordinate the cache invalidation operations, where the server does not blindly broadcast an IR but sends it to the HA. Then the HA judiciously refines and re-distributes the IR to the GFAs, where the queried data items can be validated or replied.

Algorithm 3 AP, GFA, or HA side

1: (**A**) When an AP receives a request packet from v_y:
2: append CoA_{AP} to the request packet and forward it to a GFA;
3: (**B**) When a GFA receives a request packet from an AP:
4: **if** $f_x \wedge (t_x^c \geqslant t_x^u)$ **then**
5: reply an ack packet to v_y;
6: **else**
7: append CoA_{GFA} to the request packet and forward it to the server;
8: **end if**
9: (**C**) When an HA receives an IR $(\mathcal{R}_s)$ from the server:
10: **for** $vid(v_k) \in \mathcal{R}_s$ **do**
11: find a CoA_{GFA} matched to a CoA_k;
12: create an IR $(\mathcal{R}_h), < id(d_x), t_{cur} >$, send it to the GFA;
13: **end for**

5. Analytical model

In this section, we analyze the proposed cache invalidation scheme and its enhancement in terms of a cost function. Let C_{prot} be the average communication protocol cost, which consists of the communication costs of query operation (C_{query}) and cache invalidation operation ($C_{invalid}$):

$$C_{prot} = \lambda_q \cdot C_{query} + \lambda_u \cdot C_{invalid}, \tag{1}$$

where the arrival process of the query and update is modeled as a Poisson processes with an arrival rate of λ_q and λ_u at a vehicle and a server, respectively. A set of notations used for analysis is summarized in Table 1.

5.1. The query cost

Whenever a query is generated, a vehicle accesses the server for validity before answering the query even though the queried data item is cached. Thus, C_{query} includes the communication cost between a vehicle and an AP (c_x), AP and GFA (c_y), and GFA and the server (c_s), and the query processing cost at the AP (q_a), GFA (q_g), and server (q_s). We assume that the communication cost is proportional to the distance between a source and destination, and it is represented in terms of the average number of hops ($\bar{h}$) and a constant factor of each hop (κ) [28]. Thus, c_y and c_s are $\overline{h_y} \cdot \kappa$ and $\overline{h_s} \cdot \kappa$, respectively. We also assume that the wireless communication cost is ω times higher than the wired communication cost and thus, c_x is $\omega \cdot \kappa$.

The C_{query} varies in following three cases: (i) When a queried data item is not cached; (ii) When a cached data item is not valid; or (iii) When a cached data item is valid. Both (i) and (ii) cases are treated as a cache miss, and the query request is forwarded to the server. Then the query cost of cache miss, C_{query}^{miss}, is calculated as:

$$C_{query}^{miss} = 2 \cdot (q_a + q_g) + q_s + \kappa \cdot (1 + \eta) \cdot (\overline{h_y} + \overline{h_s} + \omega), \tag{2}$$

where we assume that a data item is transmitted by a number of packets (η) [11]. The third case is a cache hit and the query request is validated in the GFA. Then the query cost of cache hit, C_{query}^{hit}, is calculated as:

$$C_{query}^{hit} = 2 \cdot q_a + q_g + 2 \cdot \kappa \cdot (\overline{h_y} + \omega). \tag{3}$$

In Eqs. (2) and (3), both C_{query}^{miss} and C_{query}^{hit} are represented in terms of the processing and communication costs. By combining Eqs. (2) and (3), C_{query} is expressed as:

Table 1
Notation.

λ_q, λ_u	The arrival rate of query and update at a vehicle and the server, respectively
t_q, t_u	The query and update arrival time, respectively
c_s, c_w, c_x, c_y, c_z	The communication cost between GFA ↔ server, HA ↔ server, vehicle ↔ AP, AP ↔ GFA, and GFA ↔ HA
q_a, q_g, q_s	The query processing cost at the AP, GFA, and server
$\bar{h}$, κ	The average number of hops, and the constant factor of each hop
$\overline{h_s}$, $\overline{h_w}$, $\overline{h_x}$, $\overline{h_y}$, $\overline{h_z}$	The number of hops between GFA ↔ server, HA ↔ server, vehicle ↔ AP, AP ↔ GFA, and GFA ↔ HA
ω	The constant factor of the wireless communication cost
η, m	The number of packets and the number of GFAs, respectively
δ, γ	The percentage of the database for local cache size in a vehicle and GFA, respectively
ξ, ξ'	The percentage of hot and cold data items in the database, respectively
a_s, a_h, a_z	The update processing cost at server, HA, and GFA, respectively

$$C_{query} = P_{hit} \cdot C^{hit}_{query} + P_{miss} \cdot C^{miss}_{query}, \tag{4}$$

where P_{hit} and $P_{miss}(=1-P_{hit})$ are the probability of cached data item being valid and invalid in the GFA, respectively.

The cache hit occurs when a queried data item is cached and accessed before it is updated at the server. Here, let us assume that the number of data items stored in the database is D and a local cache size is δ% of D. We also assume that the data items are divided into hot (popular) and cold (unpopular) data subsets, and they are ξ% and ξ'% of D, respectively, where ξ' is $100-\xi$. There is a high probability that the data item is chosen in the hot set ($P_{hot} < 1$). Here, $\delta < \xi$ and $P_{cold} = 1 - P_{hot}$. Then the probability of the queried data item, which is cached in a vehicle, P_{cache}, is expressed as:

$$P_{cache} = P_{hot} \cdot \frac{\delta \cdot P_{hot}}{\xi} + (1 - P_{hot}) \cdot \frac{\delta \cdot (1 - P_{hot})}{\xi'}. \tag{5}$$

If a cached data item is queried after it is updated, then it is an invalid data item. The probability of data item being updated during the period from the current time to the query arrival time is expressed as:

$$P_{update} = P(t_u < t_q) = \int_{t_q=0}^{\infty} \int_{t_u=0}^{t_q} f(t_q) g(t_u) dt_u dt_q = \frac{\lambda_u}{\lambda_q + \lambda_u}, \tag{6}$$

where t_u and t_q are the update and query arrival times, respectively. Then we can obtain the hit rate, P_{hit}, by combining Eqs. (5) and (6) as:

$$P_{hit} = P_{cache} \cdot (1 - P_{update}). \tag{7}$$

5.2. The invalidation cost

The $C_{invalid}$ includes the communication cost between the server and an HA (c_w), and the HA and GFA (c_z). Also, the IR update processing costs at server (a_s), HA (a_h), and GFA (a_g) are added as:

$$C_{invalid} = a_s + \overline{h_w} \cdot \kappa + a_h + \overline{h_z} \cdot \kappa \cdot m + a_g \cdot m, \tag{8}$$

where m is the average number of GFAs under an HA, and the communication cost between an HA and the server (c_w) is similarly derived as $\overline{h_w} \cdot \kappa$.

5.3. Model for the enhancement case

The average communication protocol cost $\left(C^g_{prot}\right)$ and query cost $\left(C^g_{query}\right)$ of the enhancement scheme are the same as Eqs. (1) and (4), respectively except that P_{hit} needs to be recalculated. The P_{hit} in the GFA, P^g_{hit}, occurs in following two cases: (i) A queried data item is cached in a vehicle and it is validated as a valid copy in the GFA; or (ii) A queried data item is not cached in a vehicle but its valid copy is cached in the GFA.

The first case is the same as the Eq. (7). In the second case, the probability of the queried data item, which is cached in the GFA, P^g_{cache}, needs to be calculated and it is expressed as:

$$P^g_{cache} = P_{hot} \cdot \frac{\gamma \cdot P_{hot}}{\xi} + (1 - P_{hot}) \cdot \frac{\gamma \cdot (1 - P_{hot})}{\xi'}, \tag{9}$$

where we assume that the cache size of a GFA is γ% of D. Here, $\gamma < \xi$. Then P^g_{hit} is given by,

$$\begin{cases} P_{cache} \cdot (1 - P_{update}) & (i) \text{ case} \\ (1 - P_{cache}) \cdot P^g_{cache} \cdot (1 - P_{update}) & (ii) \text{ case} \end{cases}$$

By combining both the cases, P^g_{hit} is expressed as:

$$P^g_{hit} = (P_{cache} + (1 - P_{cache}) \cdot P^g_{cache}) \cdot (1 - P_{update}). \tag{10}$$

Therefore C^g_{query} is expressed as:

$$C^g_{query} = P_{hit} \cdot C^{hit}_{query} + \left(P^g_{hit} - P_{hit}\right) \cdot C^{hit'}_{query} + P^g_{miss} \cdot C^{miss}_{query}, \tag{11}$$

where $C^{hit'}_{query} = 2 \cdot q_a + q_g + \kappa \cdot (1 + \eta) \cdot (\overline{h_y} + \omega)$ and $P^g_{miss} = 1 - P^g_{hit}$. Compared to Eq. (4), additional processing and communication costs $\left(C^{hit'}_{query}\right)$ between a vehicle and GFA are added in Eq. (11). This is because when a queried data item is not cached in the vehicle but cached in the GFA, it should be delivered to the vehicle in the enhancement scheme. Thus, C^g_{query} includes the costs for validating a query request in the GFA and delivering a queried data item from the server or GFA to the vehicle.

We show the numerical results of both the average communication protocol cost and cache hit in Fig. 3. The query processing costs at the AP, GFA, and server are set to 10, 15, and 25, respectively. Similarly the update processing costs at the GFA, HA, and the server are set to 10, 15, and 25, respectively. The local cache sizes in a vehicle and a GFA are set to 5% of database, respectively. In Fig. 3(a), both communication protocol costs increase as both query and update rates increase, but they are more sensitive to the update rate. In Fig. 3(b), the cache hit increases as the query rate increases but it decreases as the update rate increases. The enhancement case achieves better performance because of additional caching in GFAs, that helping consuming more query requests directly without forwarding them to the server.

Since the proposed model is intended only to estimate the performance trend, we conduct detail performance evaluation through extensive simulation in the following section. The performance optimization could improve the usefulness of the analysis but it is beyond the scope of this paper. For example, the query cost can be reduced by caching more valid data items at GFAs. However, the invalidation cost may increase due to frequent validation operations between the server and GFAs. Thus, reduction of the query and

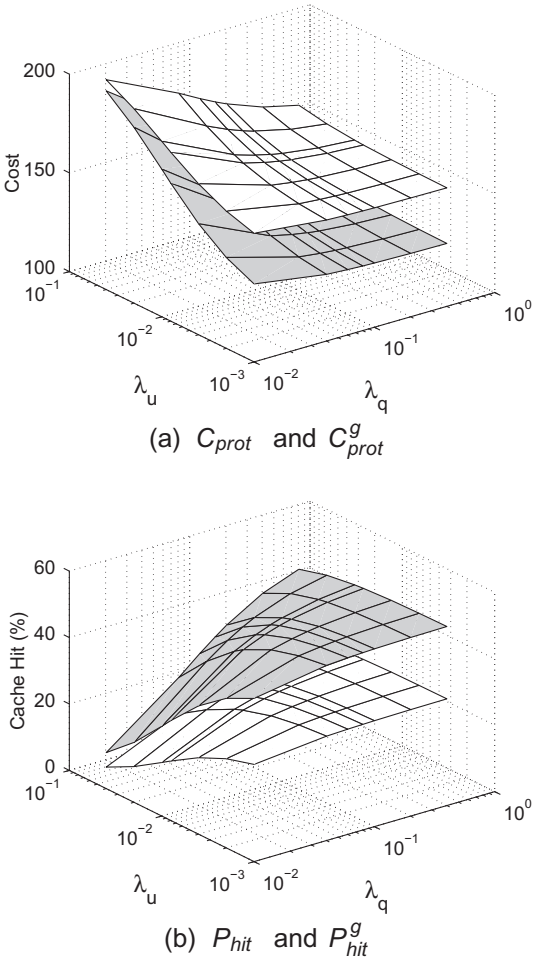

Fig. 3. The query cost and cache hit as a function of query and update rates. The enhancement case is shown in shade.

invalidation costs is conflicting requirements, and optimization of both is admittedly extremely complex.

6. Performance evaluation

6.1. Simulation testbed

We develop a customized discrete-event simulator using CSIM19 [29] to conduct our experiments. To examine the proposed idea, we use a simple wrap around one-dimensional network topology (see Fig. 4) with a velocity suitable for a city driving under emphasis on the data caching and its invalidation. In this paper, we neither consider a high speed mobility model nor deploy an advanced location management scheme (i.e., [28]) to clearly see the effect of our cache invalidation strategies on the performance.

We deploy 25 APs[1] to cover the area and assume that each AP is located in the center of a coverage area with 2 Mbps bandwidth. Five GFAs are deployed in the area and each GFA consists of 5 APs. A set of vehicles is randomly located in the area, travels with a given velocity suitable for a city driving, and communicates with the APs. The communication delay is dependent on the distance among wired network agents (e.g. AP, GFA, HA, and server) represented by the

[1] The AP (e.g. an infostation [30,5], or a message relay box [6]) to cover the area, and they are located along the road and act as a gateway to an infrastructure network. We envision that drivers should be able to access not only roadside Wi-Fi stations but also high-speed wide area 3G cellular networks and thus, the diameter of coverage area will become wider than that of we have assumed based on [31] in this paper.

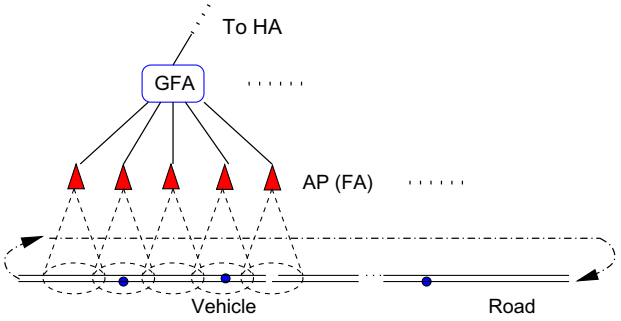

Fig. 4. A wrap-around one-dimensional network topology. A set of vehicles moves along the given road and continues to move to the opposite side of road, when it reaches to the end.

Table 2
Simulation parameters.

Parameter	Value
Network size (km)	18.85
Diameter of AP coverage (m)	750 [31]
Velocity (km/h)	60
Number of vehicle	250
Database size (items)	1000
Data item size (KByte)	1–500
Hot data items	50
Cold data items	Remainder of DB
Hot data item access prob.	0.8

number of hops. Based on [28], the number of hops between an AP and a GFA, a GFA and HA, a GFA and the server, and the server and HA are set to 5, 10, 25, and 25, respectively. We assume that the bandwidth of the wired network is 100 Mbps.

Both update and query inter arrival times follow the exponential distribution. The entire data items stored in the database is classified into two subsets, hot and cold data items. We assume that 80% and 20% of update requests are uniformly distributed within the hot and cold subsets, respectively. The Zipf distribution model [32] is often used to model a skewed access pattern, where θ is the access skewness coefficient. The access probability of the ith data item is represented as $\frac{\Omega}{i^{\theta}}$, where $\Omega = \left(\sum_{j=1}^{n} \frac{1}{j^{\theta}}\right)^{-1}$, and $0 \leqslant \theta \leqslant 1$. Setting $\theta = 0$ corresponds to the uniform distribution. Here, we set θ to 0.8 based on the real Web traces [33].

Each vehicle caches 5% of the data items in the database. When a cache is full, we use the least recently used (LRU) cache replacement policy. We do not deploy an advanced cache admission or replacement policy proposed in [26] to clearly see the effect of the proposed scheme on the performance. The simulation parameters are summarized in Table 2.

6.2. Performance comparison

To compare the performance of proposed schemes, we modified two prior cache invalidation techniques to work in IVANET: a poll-each-read (PER) scheme, and an extended asynchronous (EAS) scheme. The PER[2] scheme [10] is an on-demand approach extended from [34], where a query is forwarded to the server for either validating a cached data item or receiving the queried data item. In

[2] A server-based poll-each-read (SB-PER) scheme [10] is proposed to further improve the performance of PER under the assumption of available global access and update information in a server. However, obtaining the global update information is practically infeasible and thus, we do not consider the SB-PER scheme in this paper.

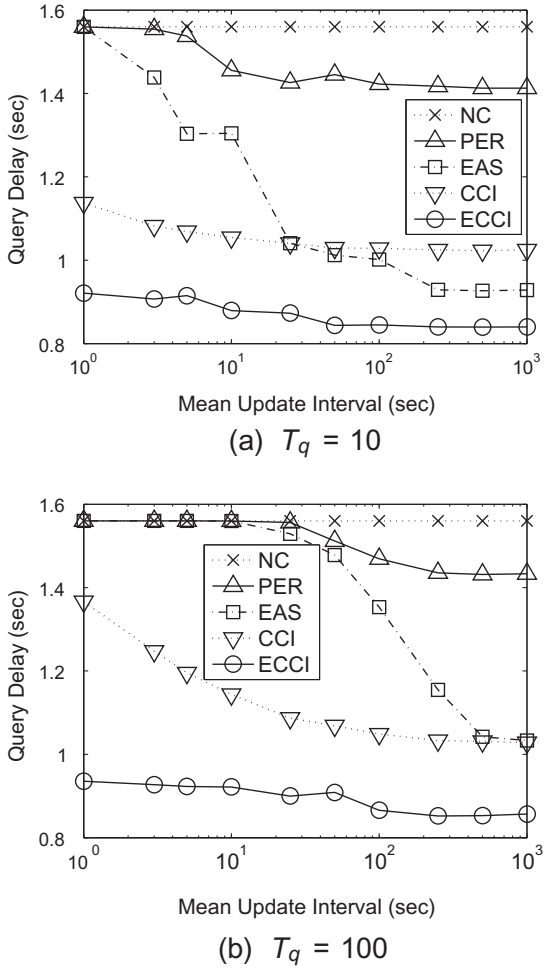

Fig. 5. Query delay as a function of mean update interval.

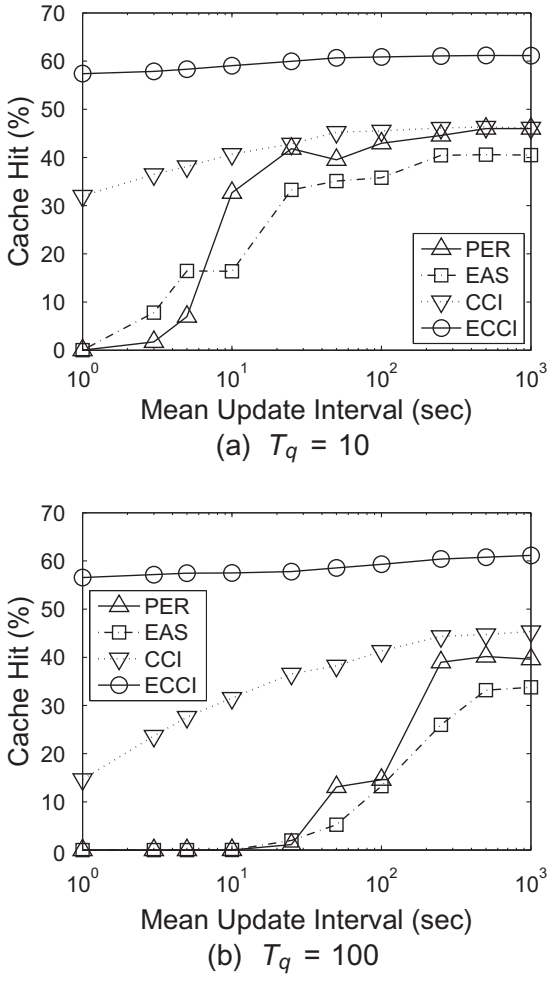

Fig. 6. Cache hit as a function of mean update interval.

the EAS[3] scheme, as we mentioned in Section 3, a stateful server maintains a list of *ids* of the cached data items in vehicles, and broadcasts an IR to inform the vehicles of any cache update. Thus, a queried data item, which is cached, is used to answer the query directly without validating with the server. When a vehicle handoffs, it sends the entire valid cached data item *ids* to the server for validity check and receives an invalidation check packet from the server. For comparison, we add the proposed cooperative approach in these schemes, where the impact of mobility on the performance is minimized. In addition, we include the base case, no cache (NC), where a query request is forwarded to the server always.

6.3. Simulation results

We evaluate the impact of update interval (T_u), query interval (T_q), and size of data items, and examine the communication overheads of the cache invalidation strategies.

6.3.1. Impact of update interval

First, we evaluate the query delay and cache hit rate of cache invalidation strategies as a function of update interval. In Fig. 5(a), as the update interval increases, the query delay decreases. The NC scheme is not affected by the update and query intervals because every query is forwarded to the server for access. In Fig. 5(b), when the query interval is high, the overall query delays increase because more cached data items become invalid before they are queried. Both PER and EAS schemes show high query delays closer to the NC scheme in the low update intervals, but the EAS scheme exhibits lower query delay with high update intervals, because more valid cached data items are found without additional validation delay with the server. Both our CCI and ECCI schemes show better performance than other two schemes. Especially the ECCI scheme shows the lowest query delay for the entire update intervals due to additional caching in the GFAs.

In Fig. 6(a), as the update interval increases, the cache hit rate increases. Both CCI and ECCI schemes show higher cache hit rates than other two schemes because a query can be validated or answered in the GFA before it is forwarded to the server. The detail analysis of cache hit is presented in Fig. 9(a) later. In Fig. 6(b), when the query interval is high, the overall cache hits decrease. However, the ECCI scheme is not affected much by both query and update intervals and shows the highest cache hit rate for the entire update intervals.

6.3.2. Impact of query interval

Second, we examine the query delay and cache hit of the cache invalidation strategies as a function of query interval. In Fig. 7(a), when the update interval is low, both PER and EAS schemes show almost similar performance to the NC scheme except for the low query intervals. This is because, due to frequent update of data

[3] Both asynchronous (AS) [9] and call-back (CB) [35] schemes are similar in the sense that a stateful server broadcasts an IR to the mobile nodes for cache updates. Since the CB scheme is designed for a single-cell environment, we extend the AS scheme for comparison in this paper.

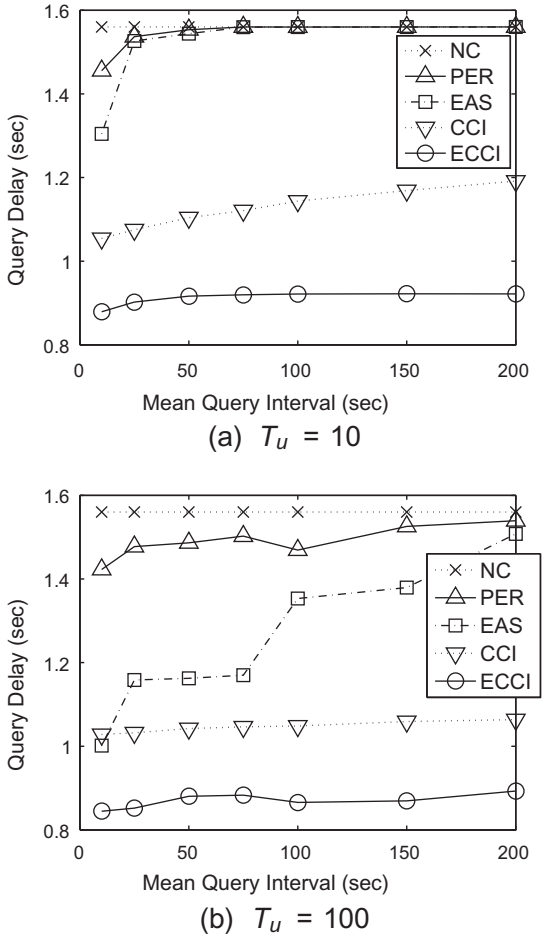

Fig. 7. Query delay as a function of mean query interval.

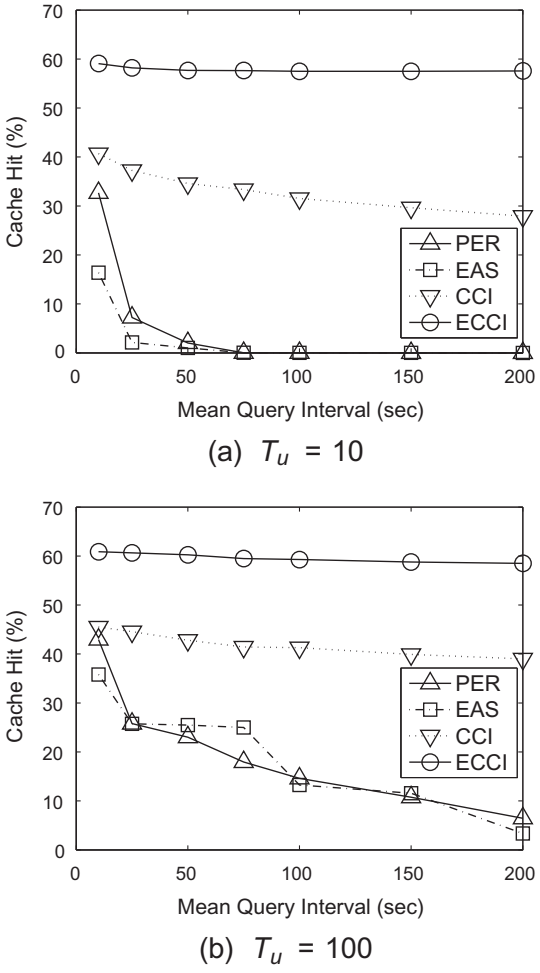

Fig. 8. Cache hit as a function of mean query interval.

items, more queried cached data items become invalid and thus, more queried requests are forwarded to the server. In Fig. 7(b), when the update interval is high, the overall query delays decrease. The EAS scheme shows lower query delay than the PER scheme because a valid cached data item can directly be used for answering a query, but higher query delay than that of both CCI and ECCI schemes because a query is forwarded up to the server when the queried cached data item is either invalid or unavailable.

In Fig. 8(a), when the update interval is low, both PER and EAS schemes show almost no cache hit except for the low query intervals due to frequent update of data items. In Fig. 8(b), when the update interval is high, both PER and EAS schemes still suffer from the low cache hit resulting in the high query delay. Regardless of query and update intervals, both the CCI and ECCI schemes achieve higher cache hits than the other two schemes.

6.3.3. Impact of cooperative approach

Third, we analyze the proposed schemes in terms of the local cache hit and a remote cache hit. The local cache hit is the ratio that a queried data item is locally cached and validated by the server, while the remote cache hit is the ratio that a queried data item is validated or found from the GFAs. In the ECCI scheme, the remote cache hit can be enhanced when a queried data item is cached in GFAs. In Fig. 9(a), when the pair of query and update interval is 10/100 (sec), both CCI and ECCI schemes show higher cache hits than other two schemes. The CCI scheme achieves the highest remote cache hit because many queries are validated in the GFAs before they are forwarded to the server. However, the ECCI scheme shows higher local cache hit than other three schemes because many queried data items found in the GFA brought to the query request senders for caching. Then these cached data items are used for answering the next query requests, resulting better local cache hits. In case of 100/10 (sec), due to frequent update of data items, both PER and EAS schemes show no cache hit, while both CCI and ECCI schemes heavily rely on the remote cache hit.

6.3.4. Impact of data size

Next, we measured the query delay as a function of data size. In Fig. 9(b), the ECCI scheme outperforms the others and achieves the lowest query delay for all data sizes. The performance gap between the NC and other schemes is larger as the data size increases. Due to additional communication with the server on every query request for validating or accessing the data items, both NC and PER schemes show higher query delays than the other three schemes.

6.3.5. Communication overhead

Finally, we evaluate the communication overhead of the cache invalidation strategies in terms of the average number of packet transmissions only from the network components including the query request, reply, and IR packet transmissions. In Fig. 10, both NC and PER schemes require six packet transmissions from a vehicle, AP, GFA, and server for answering a query. Although additional IR transmission from the server and HA is required in both CCI and ECCI schemes, they achieve the less number of packet transmissions than

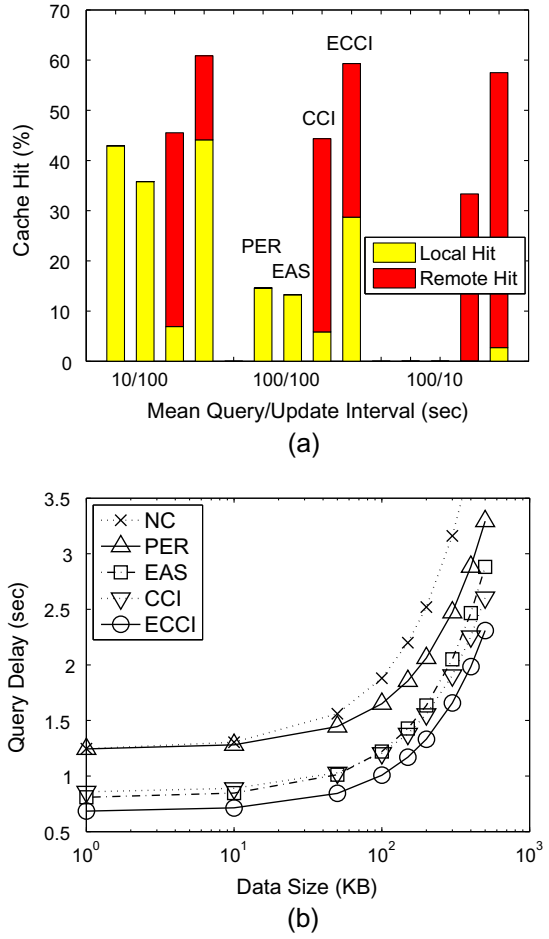

Fig. 9. The left figure shows the cache hit against the pair of mean query and update intervals, where the PER, EAS, CCI, and ECCI schemes are plotted from left to right. The right figure shows the query delay against data size (T_q = 10 and T_u = 50).

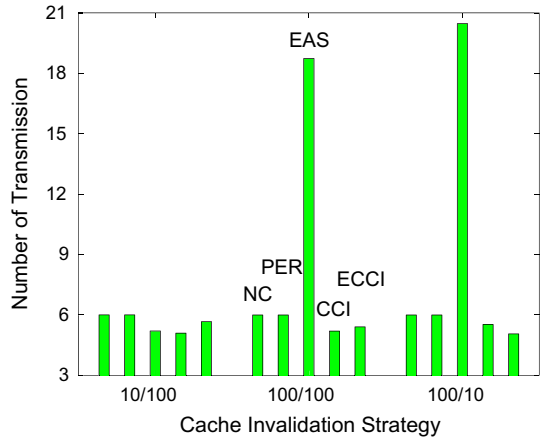

Fig. 10. Average number of transmissions per query is shown against the pair of mean query and update intervals. The NC, PER, EAS, CCI, and ECCI schemes are plotted from left to right.

that of NC and PER schemes because some of queries can be validated in GFAs before they are further forwarded to the server. The EAS scheme may reduce the query delay, but it incurs maximum communication overhead, especially in low update intervals (e.g. 100/10 (sec)) mainly due to excessive IR packet transmissions.

In summary, the proposed cooperative cache invalidation schemes provide better performance and less overhead. The cooperative approach reduces the impact of high-speed mobility and provides a scalable cache invalidation operation, because a server and network agents of location management coordinate together for cache invalidation operation. Thus, the proposed schemes are a viable approach for implementing IVANETs.

6.4. Discussion

In this subsection, we discuss how the simulation parameters and assumptions affect the caching performance. We first investigate how data access pattern affects the caching performance. In this paper, a skewed access pattern is considered and the Zipf distribution model [32] is deployed based on the real Web traces (i.e., $\theta = 0.8$). Since frequently accessed data items (i.e., hot data items) are highly preferred to be cached, a non-uniform data access pattern is implicitly assumed in most caching schemes. If the data access pattern is uniform (i.e., $\theta = 0.0$), the benefits of caching in terms of low query delay and communication overhead reduce. Also the problem of data caching can be turned into data replication [20,36], where multiple copies of data item are stored in local caches and a performance tradeoff between data accessibility and access latency is a major performance matrix.

Second, we investigate how packet collision or loss affects the caching performance. In this paper, no packet (i.e., IR) collision or loss is assumed between vehicles and APs. We can relax this assumption by considering limited broadcast bandwidth, unreliable wireless link, or high vehicle density often witnessed in a rush hour or a city event. In case of IR packet collision or loss, the query delay increases because a vehicle cannot answer a query until it receives the IR. Due to lack of receiving IRs, long-disconnection problem [13] can be occurred and the entire cached data items can be thrown away for validation. Thus, overall caching performance will decrease.

7. Research issues

We further exploit several issues open for investigation toward future research directions. First, we envision that vehicles will be able to access not only roadside Wi-Fi stations but also 3G networks. Here, in contrast to 3G networks, roadside Wi-Fi stations have advantage of easy to deploy with low cost but provide high bandwidth. When each vehicle has a connection supported either by 3G networks or satellite techniques, its high-speed mobility still becomes an issue for cache invalidation operation. In light of this, coordination with location management scheme is essential to reduce the cost of IR broadcast. Since vehicles are bounded to move under a fixed road with speed limits and traffic lights, prediction of future movement considering such restrictions will also reduce the impact of mobility.

Second, as compared to MANETs and IMANETs, the unique challenges in IVANETs such as scheduling [37] and data dissemination [38] will motivate further research in this area. Since vehicles moves with high-speed, they have a short period of time to contact a roadside unit for download and upload requests. As the number of vehicles increases in a rush hour, it is not trivial to design an efficient scheduling scheme for the server to process the requests in a timely manner. Due to vehicles' high-speed mobility, the conventional broadcast technique for data dissemination cannot directly be applied to IVANETs. In IVANETs, the roadside unit (e.g. data source) broadcasts data items to passing by vehicles and they exchange the data items with other vehicles located within the communication range. Also the roadside unit periodically broadcasts data items at the intersection, and vehicles can buffer and re-broadcast the data items at the intersection for easy of data dissemination. These

Table 3
Acronym.

VANET	Vehicular ad hoc network
IVANET	Internet-based vehicular ad hoc network
MANET	Mobile ad hoc network
IR	Invalidation report
UIR	Updated invalidation report
TS	Time stamp
BS	Base station
MSC	Mobile switching center
AP	Access point
FA	Foreign agent
GFA	Gateway foreign agent
HA	Home agent
CoA	Care-of address
DSRC	Dedicated short range communication
NC	No cache
PER	Poll-each-read
SB-PER	Server-based poll-each-read
AS	ASynchronous
CB	Call-back
EAS	Extended ASynchronous
ETS	Extended time stamp
CCI	Cooperative cache invalidation
ECCI	Enhanced cooperative cache invalidation
LRU	Least recently used

techniques can improve the data accessibility and availability and reduce the network traffic.

Third, in this paper, we implicitly assume a strong consistency model, in which a query should be answered by the latest updated data item from either a local cache or the server. However, we need to relax this assumption to support growing diversity in applications and users' demands requiring a certain consistency level with the server. Note that ensuring a strong consistency is not always a prompt and critical requirement. For example, map, video clip, and weather information are not update sensitive and thus, occasional inconsistency between the source and its cached copy would be acceptable. Little effort has been devoted in developing a flexible consistency-based technique [17,39]. We expect that this technique can effectively balance among the cost of accessing data item, complexity of consistency maintenance, and query latency.

8. Conclusion

In this paper, we investigate the cache invalidation issue in an IVANET, where minimizing the impact of high-speed mobility and designing a scalable algorithm are primary concerns. We proposed a hierarchical network model and a cooperative cache invalidation scheme including its enhancement, in which both the server and network agents of location management coordinate the cache invalidation operation. We developed a simple analytical model for the proposed schemes for rapid estimate of performance trends. We also compared the proposed schemes with two prior cache invalidation schemes through extensive simulation, and observed that the proposed CCI and ECCI schemes provide better performance than others with respect to the query delay, cache hit rate, and communication overhead.

Acknowledgments

This research was supported in part by Startup Grant in Dept. of Computer Science at Texas Tech University and US National Science Foundation (NSF) under the Grants CNS-0831673, CNS-1004210, and CNS-0831853, and Korea National Research Foundation (NRF) WCU Grant R31-2008-000-10100-0. This work was published in part at the 18th International Conference on Computer Communications and Networks (ICCCN) 2009 [40].

Appendix A

A set of acronyms used in this paper is summarized in Table 3. Note that the singular and plural of an acronym are always spelled the same.

References

[1] Dedicated Short Range Communications (DSRC) Home, <http://www.leearmstrong.com/DSRC/DSRCHomeset.htm>.
[2] X. Yang, J. Liu, F. Zhao, N. Vaidya, A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning, in: Proc. on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004), 2004, pp. 114–123.
[3] Q. Xu, T. Mark, J. Ko, R. Sengupta, Vehicle-to-Vehicle Safety Messaging in DSRC, in: ACM VANET, 2004, pp. 19–28.
[4] J. Ott, D. Kutscher, Drive-thru Internet: IEEE 802.11 for Automobile Users, in: Proc. IEEE INFOCOM, 2004.
[5] R. Frenkiel, B. Badrinath, J. Borras, R. Yates, The infostation challenge: balancing cost and ubiquity in diverse wireless data, IEEE Personal Commun. 7 (2) (2000) 66–71.
[6] H. Reumerman, M. Roggero, M. ruffini, The application-based clustering concept and requirements ofr intervehicle networks, IEEE Commun. Mag. (2005) 108–113.
[7] V. Bychkovsky, B.H.A. Miu, H. Balakrishnan, S. Madden, A Measurement Study of Vehicular Internet Access Using in Wi-Fi Networks, in: Proc. ACM MOBICOM, 2006, pp. 50–61.
[8] S. Khurana, A. Kahol, S. Gupta, P. Srimani, A strategy to manage cache consistency in a distributed mobile wireless environment, in: Proc. IEEE International Conference on Distributed Computing Systems (ICDCS), 2000, pp. 530–537.
[9] Z. Wang, M. Kumar, S.K. Das, H. Shen, Investigation of cache management strategies for multi-cell environments, in: Proc. 4th International Conference on Mobile Data Management (MDM), 2003, pp. 29–44.
[10] H. Chen, Y. Xiao, X. Shen, Update-based cache access and replacement in wireless data access, IEEE Trans. Mobile Comput. 5 (12) (2006) 1734–1748.
[11] W. He, I. Chen, B. Gu, A Proxy-Based integrated cache consistency and mobility management scheme for mobile IP systems, in: Proc. Advanced Networking and Applications, 2007, pp. 354–361.
[12] D. Barbara, T. Imielinksi, Sleepers and workaholics: caching strategies for mobile environments, in: Proc. ACM SIGMOD, 1994, pp. 1–12.
[13] Q. Hu, D. Lee, Cache algorithms based on adaptive invalidation reports for mobile environments, Cluster Comput. (1998) 39–48.
[14] G. Cao, A scalable low-latency cache invalidation strategy for mobile environments, in: Proc. ACM MOBICOM, 2000, pp. 200–209.
[15] Y. Lin, W. Lai, J. Chen, Effects of cache mechanism on wireless data access, IEEE Trans. Wireless Commun. 2 (6) (2003) 1240–1246.
[16] S. Lim, W. Lee, G. Cao, C.R. Das, Cache invalidation strategies for Internet-based mobile ad hoc networks, Comput. Commun. J. 30 (8) (2007) 1854–1869.
[17] Y. Huang, J. Cao, Z. Wang, B. Jin, Y. Feng, Achieving flexible cache consistency for pervasive internet access, in: Proc. Pervasive Computing and Communications, 2007, pp. 239–250.
[18] W. Li, E. Chan, D. Chen, S. Lu, Maintaining probabilistic consistency for frequently offline devices in mobile ad hoc networks, in: Proc. IEEE International Conference on Distributed Computing Systems (ICDCS), 2009, pp. 215–222.
[19] J. Gwertzman, M. Seltzer, World-Wide Web cache consistency, in: Proc. USENIX Technical Conference, 1996, pp. 141–152.
[20] T. Hara, S. Madria, Data replication for improving data accessibility in ad hoc networks, IEEE Trans. Mobile Comput. 5 (11) (2006) 1515–1532.
[21] G. Karumanchi, S. Muralidharan, R. Prakash, Information dissemination in partitionable mobile ad hoc networks, in: Proc. IEEE Symposium on Reliable Distributed Systems (SRDS), 1999, pp. 4–13.
[22] J. Luo, J. Hubaux, P.T. Eugster, PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems, in: Proc. ACM MobiHoc, 2003, pp. 1–12.
[23] Y. Sawai, M. Shinohara, A. Kanzaki, T. Hara, S. Nishio, Quorum-Based Consistency Management among Replicas in Ad Hoc Networks with Data Update, in: Proc. ACM Symposium on Applied Computing, 2007, pp. 955–956.
[24] W. Zhang, G. Cao, Defend Against Cache Consistency Attacks in Wireless Ad Hoc Networks, in: Proc. Mobile and Ubiquitous Systems (Mobiquitous), 2005, pp. 12–21.
[25] L. Yin, G. Cao, Supporting cooperative caching in ad hoc networks, IEEE Trans. Mobile Comput. 5 (1) (2006) 77–89.
[26] S. Lim, W. Lee, G. Cao, C.R. Das, A novel caching scheme for improving Internet-based mobile ad hoc networks performance, Ad Hoc Networks J. 4 (2) (2006) 225–239.
[27] C.E. Perkins, IP Mobility Support, Request for Comments (RFC) 2002–2006.
[28] J. Xie, I.F. Akyildiz, A Distributed Dynamic Regional Location Management Scheme for Mobile IP, in: Proc. IEEE INFOCOM, 2002, pp. 1069–1078.
[29] The CSIM User Guides, <http://www.mesquite.com/>.
[30] T. Small, Z. Hass, The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where There is a Whale, There is a Way), in: Proc. MobiHoc, 2003, pp. 233–244.

[31] S. Yoon, H.Q. Ngo, C. Qiao,On Shooting a Moving Vehicle with Data Flows, in: Proc. Mobile Networking for Vehicular Environments (MOVE-07), 2007, pp. 49–54.
[32] G.K. Zipf, Human Behavior and the Principle of Least Effort, Addison-Wesley, Cambridge, MA, 1949.
[33] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web Caching and Zipf-like Distributions: Evidence and Implications, in: Proc. IEEE INFOCOM, 1999, pp. 126–134.
[34] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, M. West, Scale and performance in a distributed file system, ACM Trans. Comput. Syst. 6 (1) (1988) 51–81.
[35] Y. Xiao, H. Chen, An Adaptive Callback Cache Access for Wireless Internet, in: Proc. IEEE GLOBECOM, 2005, pp. 1049–1053.
[36] Y. Zhang, S. Ray, G. Cao, T.L. Porta, P. Basu, Data Replication in Mobile Tactical Networks, in: Proc. IEEE MILCOM, 2011, pp. 797–803.
[37] Y. Zhang, J. Zhao, G. Cao, On Scheduling Vehicle-Roadside Data Access, in: Proc. ACM VANET, 2007, pp. 9–18.
[38] J. Zhao, Y. Zhang, G. Cao, Data pouring and buffering on the road: a new data dissemination paradigm for vehicular ad hoc networks, IEEE Trans. Vehicular Technol. 56 (6) (2007) 3266–3277.
[39] S. Lim, Y. Lee, M. Min,ConSens: Consistency-Sensitive Opportunistic Data Access in Wireless Networks, in: Proc. IEEE MILCOM, 2011, pp. 804–809.
[40] S. Lim, C. Yu, C.R. Das,Cooperative Cache Invalidation Strategies for Internet-based Vehicular Ad Hoc Networks, in: Proc. 12th International Conference on Computer Communications and Networks (ICCCN), 2009, pp. 1–6.