Wireless Pers Commun (2015) 85:1789–1814
DOI 10.1007/s11277-015-2867-3

Cooperative Gateway Cache Invalidation Scheme for Internet-Based Vehicular Ad Hoc Networks

Rajeev Tiwari[1] · **Neeraj Kumar**[1]

Published online: 30 June 2015

Abstract Internet-based vehicular ad hoc network is a fast growing technology with an aim to provide uninterrupted services such as regular safety alerts, entertainment, and resource sharing to the on board passengers even on-the-fly. On board passengers avail all these services even on-the-fly either in dense urban regions, or on highway using an Internet connection. Reduction in delay with an increase in the accuracy for accessing all these services from anywhere are the most challenging tasks to be performed in this environment. In literature, it has been found that caching the most relevant content at some of the intermediate sites may increase the overall performance of the network in this environment. In this direction, an important issue is to invalidate the cache when the original data items are updated and these data items are irrelevant for the end users. In this paper, we have been motivated and inspired from our earlier solution (Kumar and Lee in IEEE Syst J 8(4):1136–1144, 2014), in which we proposed a peer-to-peer cooperative caching scheme among the vehicles for urban scenario. But, in the current proposed scheme, we have enhanced our existing solution by proposing a *Cooperative Gateway Cache Invalidation (CGCI)* based upon cooperative cache invalidation technique. The designed scheme has the advantages of cooperation of Gateways in different regions along with the underlying location management scheme to reduce the number of broadcast operations, lesser Uplink requests, and query delay with an increase in the cache hit ratio. To test the effectiveness of the proposed scheme, we carried out extensive simulations by varying query arrival rate, object update rate, and cache size. Moreover, to evaluate the cost of query communication among the vehicles, an analytical model is also included in the proposed scheme. The results obtained confirm that the proposed scheme yields a reduction in the query delay with an increase in the cache hit ratio as compared to other state-of-the art existing schemes in literature.

✉ Neeraj Kumar
nehra04@yahoo.co.in; neeraj.kumar@thapar.edu

Rajeev Tiwari
errajeev.tiwari@gmail.com

[1] Department of Computer Science and Engineering, Thapar University, Patiala, Punjab, India

Keywords Cache invalidation · Cooperative caching · Cache hit ratio · VANET

1 Introduction

Vehicular ad hoc networks (VANETs) have emerged as a new class of efficient data dissemination from the past few years. These networks establish communication among high speed mobile vehicles, and various other devices connected to the internet for data sharing and communication [1–16]. In VANETs, vehicles act as an intelligent machine for data collection and processing from various other devices connected to the internet using IEEE 802.11 a/b/g standards along with short, medium and long range transmissions such as WiFi, Bluetooth, WiMAX, and long term evolution (LTE). For connecting any device such as-laptop, PDA, and Mobile Nodes (MN) using these standards with Internet, the device must be uniquely identified on global positions. All these devices are considered as objects on the Internet and their interconnection with each other for communication and data sharing is called as Internet of Things (IoT) [2–16]. Connecting vehicles to Internet requires connecting infrastructure with wireless MNs. The mutual exchange of safety information between vehicles for safe driving is one of the applications of IoT in Internet-based Vehicular Ad Hoc Network (IVANET) [3, 4].

Wireless communication among vehicles is one of the most challenging tasks for providing safety to the on board driver and passengers [1, 13, 16–19]. Vehicles have became intelligent machines in modern era because of advances in communication technologies. Now global positioning system (GPS), digital maps and wireless interfaces [14, 17, 20], large cache memory is embedded in vehicles from all leading companies, so that user can access various resources from the internet easily. There are many state-of-the-art protocols available for caching in VANETs, but these solutions are not sufficient to provide quality of service (QoS) to various applications in these networks. This is because of cache inconsistency, more communication costs, and longer delays of query reply. Moreover, vehicles have high mobility as nodes in VAENTs which leads to constant topological changes [2, 21–24]. So, maintaining a cache consistency with high mobility and constant topological changes is one of the most challenging tasks which requires a special attention in this environment.

One of the key objectives for improving the performance of Internet-based VANETs (IVANETs) is to cache the frequently accessed data items at a local storage. Hence, the critical design issue is the cache invalidation scheme, since all applications require consistency with the server to avoid any inconsistency wheneve data items are updated at the server. When a data item in a server is updated, it must be invalidated in cache of vehicles by broadcasting an invalidation report (IR). However, due to fast roaming of vehicles, this task is the most challenging as vehicles may change their positions quickly. During the data access by the vehicles from the server, query delay, and Uplink requests need to be minimized whenever frequent data updates occur.

From the above discussion, it is clear that delay is a major issue in accessing services when MNS have high mobility in IVANET environment. One of the feasible solutions to reduce delay is to make use of caching of frequently accesses data items to minimize the network traffic and enhance the performance. So, caching of popular data in mobile terminals (MTs/MNs) or at nearest access points (APs) may facilitate query response quickly, as query request can be replied from local nodes instead of remote servers. However, data consistency must be maintained between MTs and servers to prevent clients from answering to those queries which are invalidated from the server. A data item is invalidated if a query to access the same has been generated from the clients and it has been updated by the server and clients are still using the older data for query replies [25].

In literature, there exist various cache invalidation techniques (CIT) for VANETs which are described as follow. Time stamping technique (TS) [26] collects all updates pages for IR and broadcasts it to MTs after a fixed broadcast time (BT) interval of L secs. Due to mobility if MT missed IR, then queried data cached in MT is validated from remote server which incurs a longe delay. In update invalidation report (UIR) [28] technique, a smaller fixed size m UIRs are sent between BT times and at BT time, fixed IR is sent. In selective adaptive sorted (SAS), a group Invalidation report (GIR) [29] is generated which has a longer delay without cooperation with gateway. These proposals are based on invalidation reports (IR) to handle frequent disconnections [26]. Cooperative cache invalidation (CCI) and extended cooperative cache invalidation (ECCI) [27] are schemes which are cooperative and stateful, where a server and network agent coordinate together for invalidation of cache data, but these techniques missed cooperation of gateway foreign agents (GFA) which may reduce a longer delay for many queries generated from the clients in this environment.

1.1 Motivation

Most of the above discussed techniques [26, 28–30] suffer from longer query delay and large bandwidth consumptions due to the longer time spent in validating cached data items from server. So, in such a case, MTs can't reply for query till their cached data is validated by IR, sent from server which may increase the query response time for MTs. These schemes have not exploited the advantage of cooperation of proxy servers or other network components. Authors in [28] has proposed an update invalidation report (UIR)-based cache technique in which a server sends UIRs after every L/mth seconds, where $m < L$. The sending of UIRs may decrease query response time to the extent of hit occurrences, due to UIRs on MN. But, the replication of UIR instead of IR consumes a lot of bandwidth and also increases the congestion in the network. Due to this, query response may take a longer time with a large number of query missed with respect to the UIR report. Due to high mobility in vehicles, they may reside in multiple coverage area of different GFAs. So, they may frequently hands off the IR sent by server which may results a longer query delay. This type of delay can be reduced by cooperation among gateways.

Moreover, when one GFA is not able to validate the queried data of vehicles, then other neighboring GFAs may have valid copy of queried data. So, there is a need of cooperation among the multiple GFA to increase the efficiency of any cache management scheme in this environment.

We have also motivated from our earlier solutions [1, 31], in which we have designed cache management scheme for peer-to-peer (P2P) mobile clients in vehicular environments where the cooperation of network components like GFAs has not been considered. But, in the current solution, we have considered this factor in different environment and tested the performance of the designed scheme using cooperation of GFAs.

1.2 Contributions

Based upon the above discussion and challenges in IVANET environment, in this paper, following contributions are presented.

(i) A hierarchical network model for IVANETs is presented which consists of access points (APs), gateway foreign agents (GFAs), home agents (HAs), and data server (s) with cooperation from neighboring GFAs. The proposed scheme consists of a stateful, location management with cooperation from the neighboring GFAs.

(ii) A cooperative scheme is proposed in which both the server and Home Agents(HA) coordinate for cache invalidation and server asynchronously sends an IR to a HA rather than blindly broadcasts to vehicles. Then, the HA judiciously filters and distributes the IR to appropriate GFAs and cooperation among neighboring GFAs is maintained which answer the validity of the queried data item to reduce the unavoidable query delay which was one of the major issues in most of the existing invalidation techniques. [26, 28–30]

(iii) An analytical model is developed for predicting the query delay of the proposed Cache Invalidation Algorithm with cooperation from the neighboring GFAs.

(iv) We have modified the previous cache invalidation schemes, TS [26], CCI and ECCI [27, 37, 38] to make them suitable for IVANETs environment. The proposed scheme is compared with the above discussed schemes with respect to various performance evaluation metrics. The results obtained prove the effectiveness of the proposed scheme in comparison to the above discussed schemes.

1.3 Organization

The rest of the paper is organized as follows. Section 2 discusses the most relevant related work done with similarity and difference to the current proposal. A detailed comparison of existing protocols with respect to various parameters is also described in this Section using Table 1. Section 3 presents the network model, mobility model, and problem formulation with objective function. In Sect. 4, a detailed description of the proposed scheme is provided. Section 5 provides the analytical model of the proposed scheme. Simulation results and discussion are provided in Sect. 6. Finally, conclusion and future insights are provided in Sect. 7.

2 Related Work

There are a number of research proposals exist in literature addressing the issue of cache invalidation in IVANET. Few of these proposals have used cooperation among nodes and others have used partially the cooperation among the nodes in IVANETs. Two of the most common strategies for IR-based cache invalidation are-Object-based, and Group-based. Various research proposals in these categories are described as follows.

Barbara et al. [26] has proposed first broadcasting timestamp (TS)-based effective CIT technique. This strategy was the first object-based, stateless-server, synchronous IR-based invalidation strategy. In this strategy, at every time interval equal to L seconds, the server broadcasts an IR. Any IR_i at time T_i consists of a list of (Oid, T_d^r) pairs, where O_id is the identifier of the changed data item and T_d^r is the time stamp of the most recent data item changes, such that $Ti - wL < T_d^r < Ti$ holds. MN maintains a variable T_{lb}, in which it stores the time of the last IR. In addition, for each cached data item O_i, it stores the item's identifiers (O_{id}) and the time stamp for the item which was last invalidated (T_d^c). When MN receives the IR_i report at T_i, then depending upon state of MN, following possibilities

Table 1 Comparison table of various existing schemes

Protocols	Year	Report type	Server type	TS based	Uplink requests	Congestion	Cooperation	Query-response	Bandwidth used
TS [26]	1994	IR	Stateless	Yes	More	More	No	Slow	More
UIR [28]	2000	IR+UIR	Stateless	Yes	More	More	No	Slow	More
SAS [29, 30]	2010	GIR+IR	Stateless	Yes	More	More	No	Slow	More
CCI [27]	2012	IR	Statefull	Yes	More	Little	Yes	Fast	Little
ECCI [27, 38]	2012	IR	Statefull	Yes	More	Little	Yes	Fast	Little
P2PCCI [2]	2014	–	Stateless	Yes	More	Little	Yes	Fast	Little
Proposed	2015	Adaptive	Stateless	Yes	Less	Little	Yes	Fast	Little

may occur-If MN has disconnected before the last broadcast time, i.e., $T_{lb}(T_{lb} < T_i - wL)$, then it drops its entire cache because the disconnection time is more than wL, and all the updates may not be covered in the IR_i. Also, when $T_{lb}! < T_i - wL$, i.e., disconnection interval is less than wL seconds, then MN starts invalidation process. For all cached data objects, it checks timestamp value with the IR_i timestamp value. If $T_d^r > T_d^c$, then data item is marked as invalid. Otherwise, it is marked as valid. The main advantage of this approach was that the MNs can be disconnected for w seconds and can invalidate its cache at next IR broadcast. But, the disadvantages are that it has a larger query delays, because for query response, client has to wait till next IR from server, as it is a non cooperative scheme. Instead of looking for valid data from neighboring server, MT waits for next IR from server which may generate a large query delay and when update rate increases, the report size increases significantly and consumes more bandwidth in this scheme. Also, when data is not invalidated on MTs, then data is queried from remote server by making an Uplink request which further increases the delay incurred.

Cao et al. [28] introduced a new type of IR report generation scheme called as update invalidation report (UIR). In UIR, the broadcast interval L is divided into $(m - 1)$ sub intervals $(m > 0, m < L)$, and UIR is broadcasted after every L / m seconds. The server broadcasts an IR_i at T_i containing the history of changes in last wL seconds where, $w > 0$. IR is an OIR-based scheme having data id and timestamp $< O_{id}, t_{id}^r >$, such that $T_i - wL < t_{id}^r < T_i$ holds. At time $T_{i,k}$, server broadcasts the k_th UIR report denoted by $UIR_{i,k}$ so, it has different broadcast times for UIRs. This UIR is a list of data items identifiers which have been changed after last broad casted IR. With UIR, when a client has validated its cache then, it replies fast to the queries raised by other clients so, response time in this scheme is fast but, due to number of UIR along with IR broadcast, it consumes more bandwidth. Main disadvantage of this scheme was that if one IR is lost then, entire scheme collapses and suffers from false invalidations. Moreover, when any data is invalidated on MT, then due to non-cooperation among the infrastructure components, valid data is brought from remote server which may generate a longer query delay and a large number of Uplink requests, making more congestion to the Uplink channel.

Safa et al. [29, 30] proposed a hybrid technique called as selective adaptive sorted (SAS) cache invalidation technique, which uses both type of IRs object and group-based schemes for cache invalidation. It was an improvement over SCI [32] technique as size of IR increases but, it has historical IR content which supports disconnected MNs. So, only current changes in current Broadcast time are sorted as per time stamps and are sent to MNs. So, MN has to scan new contents only. Due to its selective, adaptive and sorted nature, it takes lesser response time for the query reply. As the object update rate becomes high, it may generate a higher IR so it becomes unfavorable scheme in IVANET environment. Moreover, it consumes more bandwidth than the other schemes. There is no cooperation among the network components, so it has longer query delay.

Neeraj et al. [2] proposed a cache invalidation scheme for P2P cooperating mobile clients in vehicular environment.To minimize the load on the infrastructure, traffic information among vehicles is shared in a P2P manner using a Markov chain model with three states. The replacement of existing data to accommodate newly arrived data is achieved in a probabilistic manner. The probability is calculated using the time to stay in a waiting state and the frequency of access of a particular data item in a given time interval. In the initial state, a query is accepted from a query generator, which is a client in this case. Depending upon the request in the query, the data for the query may be available in a vehicle's cache. If they are available, the query result is passed at once to the client, and the process is over. However, if the data are not available, it is passed to some other vehicles until the reply of the query from other vehicle cache is not available, and the process enters into the waiting state. After the query result is passed to the requesting vehicle, the state of the system is changed to the update state or to the initial state to receive the next query. Also, some other cooperation-based schemes [33–36] are proposed in literature.

Suno et al. [27] proposed a cooperative cache invalidation Technique. CCI technique works by cooperation among different levels of components and neighbors. So, a variety of cooperative schemes have been proposed such as- [37–41]. Lim et al. [27] proposed a cooperative stateful approach for wireless network. Some other techniques such as [42, 43] have been proposed for IVANETs. The architecture proposed by the authors has a server and agents for location management which works together in coordination for cache invalidation operations. The server and network agents maintain a list of data items such that vehicle that acts as mobile client and can access the data item as per their requirements. As it was a stateful approach so, it provided unlimited disconnection times for MTs. So, whenever a data item is updated, it sends an IR to a home agent (HA) rather than blindly broadcasting to multiple cells to increase the congestion. Then, the HA judiciously refines and re-distributes the IR to appropriate GFA, where they can answer the validity of a queried data item requested from a vehicle. GFAs do not send the IR to an individual vehicle but, reply a vehicle's validity request on-demand basis. Since a vehicles location is implicitly maintained at both GFA and HA, the server can avoid keeping track of the vehicle's current location, which is in fact a duplicated operation with the location management. Also, since a vehicle has a low probability of finding common data items in adjacent vehicles, broadcasting the same IRs to different vehicles is inefficient in IVANETs. So, whenever there is query on MN (Vehicle), it would be answered by validating the cache with IR multi-casted by GFA so, query response time in this scheme is average but, it has considerably reduces the number of Uplink requests to the server.

In enhancement of CCI scheme, called enhanced CCI (ECCI) [27, 38], the GFAs cache data items in their local storage for future query requests from vehicles. Updated data is piggy backed along with acknowledgment packet from the server. If GFA catches the packet and stores the updated page within its cache and if it maintains the list of updated pages then in future, it can serve the requests by checking validity at GFA and can be replied quickly to minimize the query delay and reduces the Uplink request considerably.

Table 1 gives the comparison of the existing protocols discussed above with respect to various parameters.

3 System Model

This section illustrates the network model used mobility model used and problem definition.

3.1 Network Model

The hierarchical network model for IVANET is motivated by a cellular network which consists of infrastructure support such as-APs, GFAs, HAs, and servers for data residency (S). In this model, GFAs have coperation with one another. Vehicles have IEEE 802.11-based Dedicated Short Range Communication (DSRC) transceiver. So, vehicles can either communicate with one another or connect to the Internet. For providing a location tracking capability in vehicles, they should have a built-in navigation system via GPS, with a digital map to know their locations. In this paper, we have considered a hybrid architecture in which APs and GFAs are deployed in such a manner so as to connect MTs to clients in different network domains. APs and GFAs are the part of Distributed Service Set (DSS) protocol standard with IEEE 802.11, while MTs are included in basic service set (BSS). These two are statically connected to S as shown in Fig. 1.

3.2 Mobility Model

In the mobility model, vehicles are bounded to the underlying fixed roads, mostly linearly with speed limits and traffic lights in VANETs. Here, Vehicles can communicate on-the-go with other vehicles or to fixed infrastructure components like AP. On high-speed, when a vehicle accesses the Internet, it looks for its geographical region AP, which is a gateway to the Internet for accessing information and services. Also, the AP plays an additional role to perform a location update/register operation as when a vehicle moves from one coverage area to another that involves an AP, a GFA, and HA. So, whenever hand off occurs, location of vehicle is maintained in GFA, and HA. Hence, when HA has to re-distribute IRs to vehicles, it sends it to new GFA's location under which a vehicle resides.

3.3 Problem Formulation

There are number of challenges for validating the cache consistency of vehicles in an IVANET. In a vehicle initiated cache invalidation approach, MN sends Uplink requests to server to invalidate the cache. So, as the number of MNs increases, it generates more Uplink

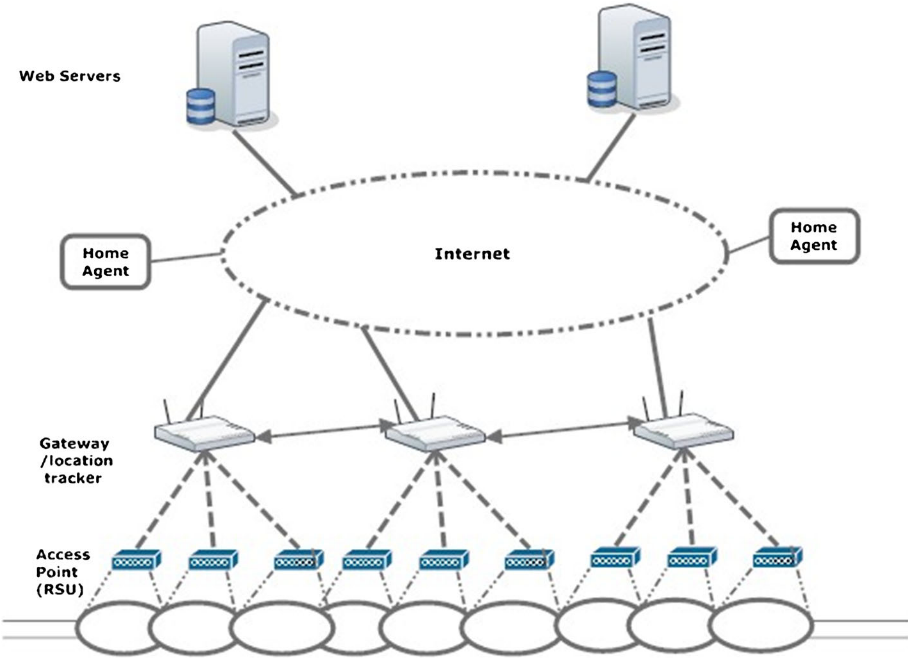


Fig. 1 Network model used

requests results in generation of congestion to the Uplink channel. Due to the mobility, vehicle may hand off from its current GFA location to neighboring GFA. By considering hot data access of a geographical region, its popularity would be same in neighboring GFAs. So, there are fair chances to get the requested data from neighboring GFAs. Hence, we have exploited GFA cooperation in IVANET environment, which has been missed from earlier CIT techniques [26, 28–30]. As in ECCI, the GFAs cache data items in their local storage for future query requests from vehicles. So, GFA can serve the requests by checking validity at GFA and can reply to vehicles query quickly to minimize the query response time and reduces the Uplink request considerably. So, hot data pages are cached on GFAs. Hence, by making cooperation of neighboring GFAs, queries can be replied faster as round trip time of MN to GFA is lesser than round trip time of MN to server.

$Minimize\ \{Query\ Response\ Time(QRT)\ \&\ UplinkRequests(UR)\}$

$$SubjectTo: \begin{cases} P_{hit} + P_{miss} = 1 \\ P_{hit} \geq P_{miss} \geq 0 \\ P_{GFA}^{hit} \geq 0 \\ \tau_{GFA} \leq \tau_{server} \end{cases} \tag{1}$$

Equation 1 is used to minimize the query response time for each query reply because we have not considered any other state in cache except hit or miss, so probability of hit and miss is collectively one. If we consider the hot data, then probability of hit is greater than probability of miss, so by considering Least Recently Used page replacement algorithm, hot data is placed in cache. So, we have some finite value of P_{hit}, i.e., $P_{hit} \geq P_{miss} \geq 0$ which results $P_{GFA}^{hit} \geq 0$. Moreover, as GFA is located nearer to roads as compared to servers deployed remotely, so Round Trip Time from Vehicle to GFA is lesser than from Vehicle to Server. Hence, query replies are generated between vehicle cache, GFA, and neighboring cache by using the cooperation among these entities. Hence, lesser Uplink requests are generated towards server which generates a lesser overhead to maintain the cache at various levels in vehicular environment.

Table 2 describes various symbols and their meaning used in this paper.

4 Proposed Scheme

Based upon above issues and challenges, this section illustrates the proposed solution to the problems discussed above. In the architecture shown in Sect. 3, every vehicle has cache memory with it. A Server has all the data D so that all updates of data items occurs only on the server. Here, server is a stateful and synchronous in nature. So, a hybrid cooperative scheme is proposed with cooperation of infrastructure components and neighbor GFAs. As in ECCI, when some miss occurs in cache of vehicle, queried data is requested from server and then the GFAs cache data items in their local storage. So, GFAs can serve the requests by checking validity at GFA and can reply to vehicles query quickly. In this paper, we have enhanced ECCI by considering cooperation of neighboring GFAs.

When neighboring GFAs starts cooperation among themselves, i.e., they send request and reply to each other. As in ECCI [27, 38], GFAs maintain a list of updated valid pages so that they can fulfill future needs of vehicles. So, extending this scheme to further levels is explained as follows.

Table 2 Symbols and their meanings

Symbols	Meaning
P_{hit}	Probability of finding page in cache
P_{miss}	Probability of page missing from cache
IR	Invalidation report
D	Total data on server
H	Hot data %age
C	Cold data %age
d_x	Data page Id
$Page(d_x)$	Data page d_x
w	No of broadcast intervals for carrying history
ts_{curr}	Current time stamp
l_{bt}	Last broadcast time
DRP	Data request packet
$Cache_V$	Cache of vehicle
U_{Rate}	Update rate of data on server
QP_s	Query processing delay time at server
QP_{bs}	Query processing delay time at BS
QP_{ap}	Query processing delay time at AP
QP_v	Query processing delay time at Vehicle
N_{gfa}^{s}	No of hops between server & GFA
N_{ap}^{v}	No of hops between vehicle & AP
N_{gfa}^{ap}	No of hops between AP & GFA
N_{ha}^{gfa}	No of hops between GFA & HA
k	Constant delay of each hop
n	Number of packets
w_c	Wireless communication cost
C_{size}	Cache size on vehicle
Q_{rate}	Query rate
QRT	Query response time
τ_{GFA}	Round trip time from vehicle to GFA
τ_S	Round trip time from vehicle to server
P_{GFA}^{hit}	Probability of hit of a queried data on GFA
UR	Uplink requests
U_{rate}	Update rate of current t_{bt}
COA_{AP}	Care of address of AP
COA_{GFA}	Care of address of GFA
COA_V	Care of address of vehicle

When a vehicle requests for a page and is not cached in AP, then it is looked into GFA for updated valid page. If data item is presented, then replied back otherwise, instead of looking into server (remote or far away) for the requested page. GFAs now has cooperation with one another, so requested page can be looked into neighboring GFAs cache. Neighboring GFA cache has a probability of P_{GFA}^{hit} of finding the requested page in their cache. So, if updated and validated page exists at neighboring GFA, then page is replied back to home GFA and further to its AP and then to vehicle.

4.1 Cooperative Approach

In this paper, we propose a cooperative stateful approach, where a server and HAs of location management coordinate together for cache invalidation operations along with cooperation of neighboring GFAs. The server and HAs has a list of which data item is accessed by which vehicle and is aware of vehicle's id. Thus, whenever a data item is updated, it sends an IR to an HA which further unicasts the intended data to vehicles only. In CGCI, major cooperation is of neighboring GFAs for validating the data requests and query responses. After a fixed time interval *t*, GFAs exchange their list of pages with neighboring GFAs to make a *Valid_Unified_list* wich can be refereed when any miss occurs from a vehicle cache.

4.2 Design and Structure

Each GFA has a concept of updated registry (list of updated pages). They exchange this update registry among neighboring GFAs after a fixed time, then every GFA has a consolidated list named as *Valid_Unified_List* of its own updated pages plus list of neighboring GFA updated pages. So, when a request for a non cached page comes to GFAs, then GFA checks whether it is presented in its own list or in its neighboring GFA. If present in neighbors GFA, only then request is sent to neighboring GFAs. Otherwise, it acts like ECCI [27, 38]. So, if there is any chance of negotiating query delay only then CGCI comes into action; otherwise, ECCI works.

Modifying ECCI technique [27, 38] in this way can provide the following advantages.

- Faster query response time: As now data request after missing from cache of vehicle can be resolved from cooperating GFAs instead of remote server. So, queries can be replied faster as:

$$\tau_{GFA} \leq \tau_{server}.$$

- Lesser congestion on server, as more request are fulfilled at lower level GFAs
- Higher hit rates: Hit rate of GFA (Similar to ECCI) + hit rate of neighboring GFAs
- More efficient utilization of GFAs.

4.3 Detailed Design

In this subsection, we describe the detailed operation of proposed CGCI scheme and cooperation of neighboring GFAs. The infrastructure components working is same as proposed in [27]. Stateful server is used to keep track of used list, i.e., which data d_x is used by which vehicle node, where $d_x \epsilon D$, and D is set of all data items on server. On updation of any data on server after a fixed time interval of L secs, an IR is generated and sent to HA. It contains records of all the vehicles which are roaming under its GFA. Then, IR is regenerated depending upon which data is cached under which vehicle and GFA. Depending upon number of matched GFAs, registry entries are created, Then, this newly generated registry entries are unicasted to corresponding GFAs and then to vehicles. When a vehicle connects to a server, then all its data cached and time stamp $< dx, ts, COA_V >$ are recorded on server. Whenever a server gets a request for data validation, then it also updates data residency in vehicle and its time stamp *ts* as t_{curr} in its own records. Also, when a vehicle handoffs to the coverage area which is under a different GFA, then the HA updates its registry and forwards the updated pages to the new GFA when a cached data item is updated in the vehicle. The same can also be forwarded to the appropriate APs to further reduce the query latency, but it may generate an extra overhead for the GFA, if a vehicle frequently moves among the coverage areas. So, handsoff are managed using location agents and GFAs.

According to proposed CGCI technique, let us describe the query response and cache invalidation mechanisms. When a vehicle generates a query for data item d_x, following procedure is executed. Firstly, when it is cached in vehicle and is validated after getting registry entry from HA or sending a DRP to AP, then acknowledgement packet is sent to vehicle directly to validate the data cached. Then, vehicle replies to queried data fastly with cached data. Secondly, when data is not cached in cache or not invalidated from AP or GFA, then DRP is appended with COA_{AP} and forwarded to GFA. DRP is checked with *Valid_Unified_list*. If requested data request is matched with this unified list of neighboring GFAs, then request is forwarded to neighboring GFA instead of sending it on server, and DRP is validated and requested data is sent to corresponding GFA which sends it further to desired vehicle via AP.

Algorithm for GFAs with cooperation is given below.

Algorithm 1 GFA Cooperative Algorithm

```
1:  Input w: Broadcast Time Window & $w = n * t_{bt}$. Server
    Keeps Track of Update of Data
2:  Output: Sending DRP to neighboring GFA or Requested Data Page to Vehicle.
3:  When GFA receives request from AP
    DRQ matches Cached Data
4:  Reply an Ack. to via AP
5:  Append to request packet requested data is present in
    neighbor GFA
6:  Send this requested page to ighbor GFA
7:  Send request packet to server
8:  When GFA Receives request packet from neighbor
    GFA then
9:  Reply requested page to GFA
10: When GFA receives a reply packet from GFA then
11: Go to step 19
12: When GFA receives a reply packet from server then
13: Update its registry with updated page and store it in its
    cache
14: Update list of valid pages.
15: When GFA receives an IR from HA then
16: Maintains list of updated pages for validation
17: After a fixed interval: Call Valid_Unified_List();
18:
19: Valid_Unified_List() {
20: Send list of stored pages to its neighbors
21: Receive list of stored pages from neighbors GFA
22: Create the unified list of stored pages of its own and
    neighbors.
23: }
```

To describe the functioning of cooperation of GFAs, a sequence diagram is given in Fig. 2 having following set of sequence of steps.

Step 1.1 As vehicle request for data via AP, then the DRP is forwarded to GFA via AP.

Step 1.2 AT GFA, validity of requested data is checked. If found valid then.

Step 1.3 Reply with an acknowledgment packet and vehicle can reply with cached data.

Step 1.4 If not validated on GFA then.

Step 1.5 Append COA_{GFA} to DRP as DRP, COA_{GFA}.

Step 1.6 Then, GFA's *Valid_Unified_List* is refereed to see if DRP is present on neighboring GFA. If presented on neighboring GFA, then.

Step 1.7 Send this DRP, COA_{GFA} to neighboring GFA.

Step 1.8 Neighboring GFA replies with data.

Step 1.9 Then, at parent GFA, registry is updated and page is saved to fulfill future requests.

Step 1.10 GFA replies with requested page and vehicle can reply with it.

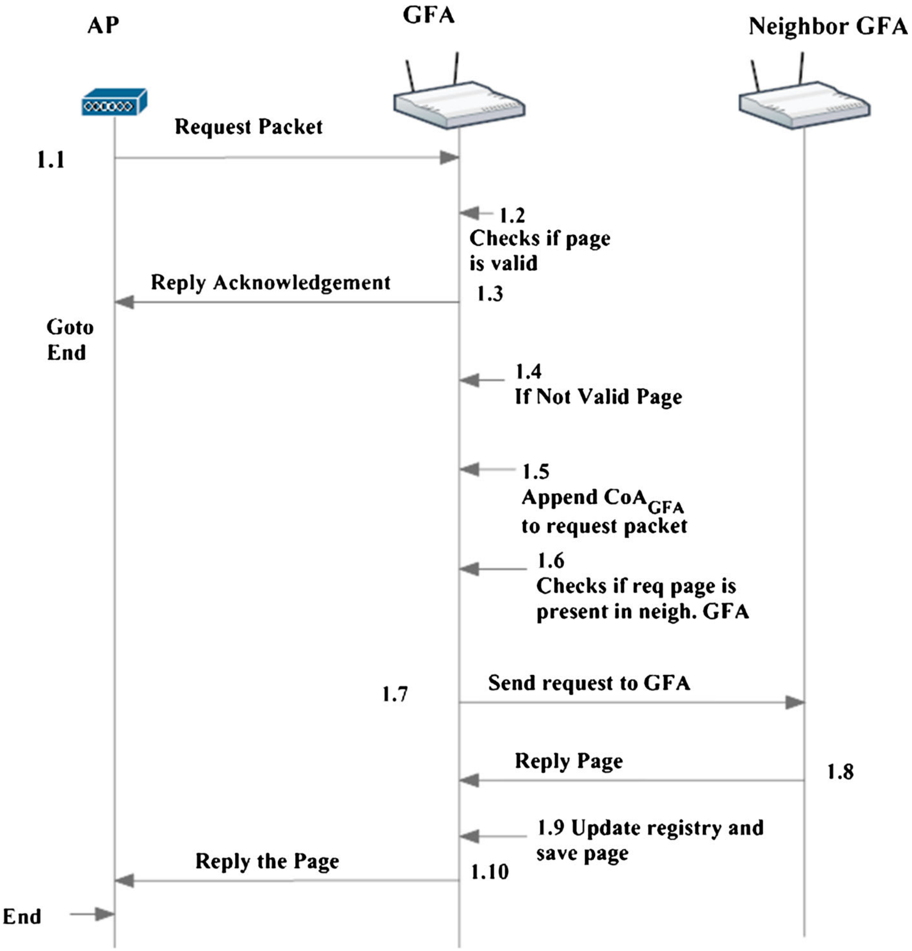


Fig. 2 Sequence diagram used for CGCI

5 Analytical Model

In this section, we presented an analytical model for computing the cost of the proposed cooperative gateway cache invalidation technique. In this model, we have computed the cost for query reply using cooperation among neighboring GFAs thus providing mathematical formulation for query response time, where arrival of queries and updates are considered as a Poisson distribution.

5.1 Query Response Time and Uplink Requests

All the symbols used are listed in Table 2.

$$QRT = P_{hit} * (TimeSpent1) + P_{miss} * (TimeSpent2) \tag{2}$$

where, P_{hit} is probability of getting a hit of queried data on GFA, and P_{miss} is probability of getting a miss from GFA. $TimeSpent1$ is time spent when hit occurs in local cache of vehicle for queried data and $TimeSpent2$ is Time spent when miss occurs in cache of vehicle, when the data is looked from GFA, neighboring GFA, and server.

The $TimeSpent1$ is considered as a hit case, when queried data is in cache and is validated from GFA via AP. So, query processing occurs twice on AP, i.e., to-and-fro of data packets. Once query processing occurs on GFA, then twice query processing delay incurred for Vehicle and AP to GFA has a factor $2 * k * \left(N_{gfa}^{ap} + w_c\right)$ and the combined value is written as follows.

$$TimeSpent1 = 2 * QP_{ap} + QP_{gfa} + 2 * k * \left(N_{gfa}^{ap} + w_c\right) \tag{3}$$

Moreover, $TimeSpent2$ is the time spent during miss occurrence. It consists of cost of looking data from neighboring GFA due to cooperation and cost of looking data from server. So, in CGCI technique P_{miss}, we can write as follows.

$$P_{miss} = P_{GFA} + P_S \tag{4}$$

where, $P_{GFA}\&P_S$ are probability of finding data from neighbor GFA and probability of finding such missed data on Server. So, $TimeSpent2$ can be given as below.

$$TimeSpent2 = C_{query}^{miss}(GFA) + C_{query}^{miss}(S) \tag{5}$$

$$(P_{miss} * TimeSpent2) = \left\{P_{GFA} * C_{query}^{miss}(GFA) + P_S * C_{query}^{miss}(S)\right\} \tag{6}$$

Here, $C_{query}^{miss}(GFA)$ is total time spent for query data validation on miss occurrence from neighboring GFAs, and $C_{query}^{miss}(S)$ is total time spent for query data validation on miss occurrence from server which is similar to ECCI [27, 38] scheme.

Moreover, $C_{query}^{miss}(S)$ and $C_{query}^{miss}(GFA)$ can be given in terms of delays of infrastructure components as given below in Eqs. 7 and 8.

$$\begin{aligned} C^{miss}_{query}(S) = {} & \left\{2 * QP_{ap} + 2 * QP_{gfa} + QP_{s}\right\} \\ & + \left\{N^{ap}_{gfa} * k + N^{s}_{gfa} + w_c * k\right\} \\ & + \left\{N^{ap}_{gfa} * k * n + N^{s}_{gfa} * k * n \right. \\ & \left. + k * n * w_c\right\} + \left\{w_c * k + N^{ap}_{gfa} * k\right\} \\ & + \left\{k * n * w_c + N^{ap}_{gfa} * k * n\right\} \\ & + 2 * QP_{gfa} \end{aligned} \tag{7}$$

$$\begin{aligned} C^{miss}_{query}(GFA) = {} & \left\{2 * QP_{ap} + 2 * QP_{gfa} + QP_{s}\right\} \\ & + \left\{w_c * k + N^{ap}_{gfa} * k\right\} \\ & + \left\{k * n * w_c + N^{ap}_{gfa} * k * n\right\} \\ & + 2 * QP_{gfa} \end{aligned} \tag{8}$$

So, Eq. 2 can be rewritten using Eqs. 3, 6, 7 and 8 in terms of delay constants of infrastructure components and delay of neighboring GFA.

$$\begin{aligned} QRT = {} & P_{hit} * \left(2 * QP_{ap} + QP_{gfa} \right. \\ & \left. + 2 * k * \left(N^{ap}_{gfa} + w_c\right)\right) + \left\{P_{gfa} * C^{miss}_{query}(GFA) \right. \\ & \left. + P_S * C^{miss}_{query}(S)\right\} \end{aligned} \tag{9}$$

As per Eq. 8, P_{GFA} times the query miss is validated from neighboring GFAs instead of getting valid data from server. So, it reduces the Uplink requests of validation of data leaving upper side network less congested. □

6 Performance Evaluation

6.1 Simulation Settings

We have customized discrete event network simulator to evaluate the performance of the proposed scheme. We have used simulator NS2.35 [44, 45] for simulation of the designed scheme. We have considered a scenario consisting of wired and wireless connection with hierarchical addressing, comprising of domains and clusters in network topology. We have taken 10 Vehicles, 3GFAs,2 APs, 1 HA, and 1 server with all data items. We have deployed 2 APs to cover area under a GFA and considered AP situated centrally. Then, ten vehicles are deployed on a straight track with velocity of 30 km/hr which is suitable for local city driving. We have considered 100 Mbps bandwidth of infrastructure network considered in the proposed scheme, i.e., between Server and HA. Both update and query arrival times are exponential distributed with query arrival and update as Poisson's distributed. We have taken 10 % of total data D as hot data. The simulation parameters are summarized in Table 3 as given below.

Table 3 Parameters used and their values

Parameter	Default value
Number of vehicles	10
Database size (D)	1000 data items
Cache size (C_{size})	500 to 2500
Broadcast window size (w)	10
Broadcast time (L)	20 s
UIR times (m)	4
Data id (d_x)	32 bits
Data size	2048 bits
Time stamp size	64 bits
Query arrival rate (Poisson's Dist.), (Q_{rate})	2 queries/s
Object update rate (Poisson's Dist.), (U_{rate})	5 objects/s
Distribution of query arrival times	Exponential
Distribution of update arrival times	Exponential
%age of hot data items (H)	10 % of D
%age of cold data items (C)	90 % of D (remainder)
Topography	670*670 m
Simulation time	1000 s
Routing protocol	DSDV
No of GFA	3
No of AP	2
No of server	1

6.2 Results and Discussion

To evaluate the performance of proposed CGCI technique, we have modified earlier existing techniques to work in IVANET environment. We have considered the existing benchmarked techniques for comparison as-No cache (NC), UIR, SAS, CCI, and ECCI [27–29, 37] as described in Sect. 2. In NC scheme, all cache requests are satisfied from server itself.

6.2.1 Effect of Object Update Rate

When object update rate increases on server, then cache miss ratio also increases on vehicle side because most of the requested data items stored in the cache of vehicle are invalid. Also, the number of Uplink requests sent to server increases, which has increased query response time as queries are answered from server. This occurs in No Cache, UIR, SAS,CCI and ECCI techniques but, in the proposed scheme, we have cooperation among neighbor GFAs and they contain list of valid hot data. So, queried data can be validated from GFAs instead of server. Moreover, round trip time of vehicle to GFAs is lesser than vehicle to server for getting data from server. So, we have more chances of getting valid data either from GFA or from neighboring GFAs in the proposed scheme.

$$\tau_{GFA} \leq \tau_{server}.$$

So, majority of queried data requests are satisfied or validated from GFAs or through neighboring GFAs. The Uplink requests are sent only for cold data query requests with reduced frequency in the proposed scheme which results a reduction in Uplink requests. For reference, we have taken the base case of No Cache Scheme where query delay is very large and higher number of Uplink requests exists.

- *Impact on query response time-(QRT)* Fig. 3 shows the variation of query response time with object update rate of NC, UIR, SAS, CCI, ECCI, and CGCI. All the schemes have lower AQRT for smaller values of object update rate. But, it increases with an increase in Object Update rate in all the schemes. QRT is very large in NC case as it has no lacal cache to respond even to frequently accessed pages. It increases sharply in case of UIR because, with such larger update rates, the data in cache becomes invalid and more Uplink requests are sent. In SAS, with higher update rates, the QRT becomes larger due to longer IRs and historical IR data. In CCI and ECCI, it has reduced and not varied sharply as it has considerations of hot data in cache and cooperation of location agent and HA. But, in CGCI, the variation of QRT is not much as it has cooperation of GFAs, queries can be replied instantly with updated page as data is sent along with IRs otherwise can be resolved from cooperative GFAs
- *Impact on Uplink requests* In Fig. 4, with higher update rates, data in cache becomes invalid in peer test case techniques. Once any miss occurs for queried data then in these techniques data is brought only from the server, there is no intermediate component or any cooperation to invalidate the data at lower ends to reduce the up link requests. Only in ECCI technique, it has some lesser Uplink request as it's GFA has an enhancement of storing the popular hot data in it and it can respond directly to queried data in future. So, in other cases, more Uplink requests are sent on server to validate a cache data. But, in case of CGCI, as update rate becomes higher, but hot data is considered which has more chances of being cached and due to cooperation among infrastructure components updated data reaches to GFAs and cooperating GFAs validate the data of vehicles. So, lesser Uplink requests are sent to the server in hierarchy.
- *Impact on hit ratio* As shown in Fig. 5 with higher update rates the hit ratio of all the schemes is decreased. As in UIR, it is falling sharply with an increase in update rate. In SAS, there is a sharp decrease of hit ratio. But in CCI, the hit ratio is not decreased sharply as it has considered hot data cached in the cache of vehicles, whose updates are

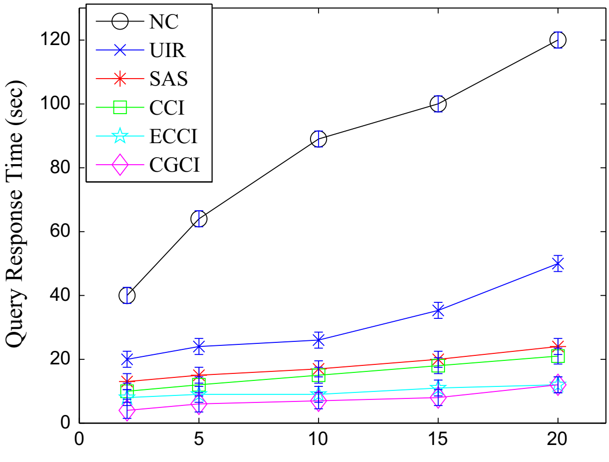


Fig. 3 Variation of query delay with object update rate

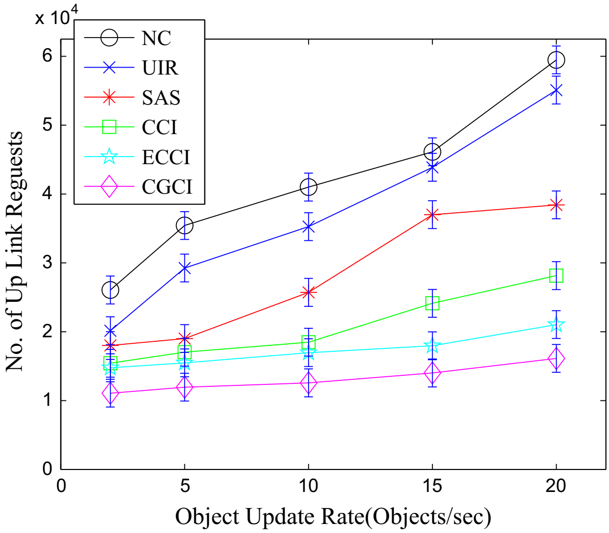


Fig. 4 Variation of no of Uplink requests with object update rate

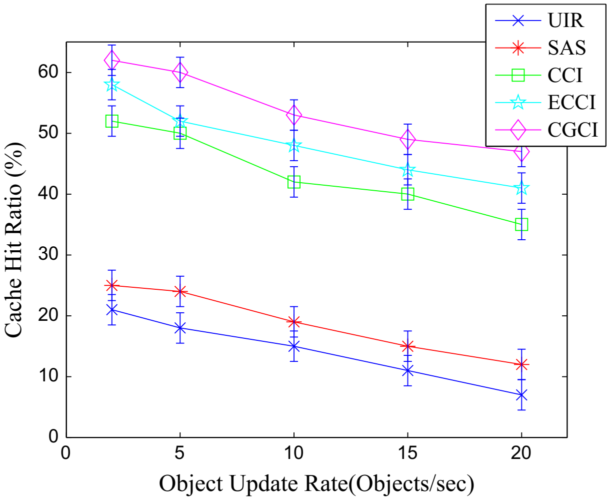


Fig. 5 Hit ratio (%) with object update rate

stored at corresponding GFAs by HA. In ECCI, the decrease in hit ratio is gradual as it has GFA, which has stored hot data pages in it and respond on time. But in CGCI technique, hit ratio is good and is not decreasing sharply as neighboring GFAs has cooperation among themselves and data is validated from corresponding GFA and ite neighboring GFAs after checking the meta data of Valid_Unified_List of GFAs.

6.2.2 Impact of Cooperative Approach

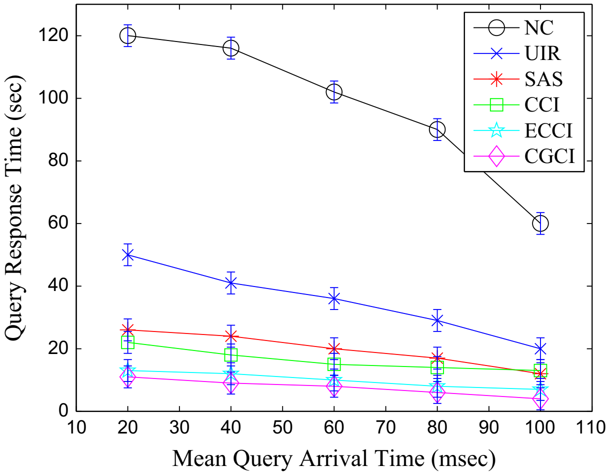
Techniques such as NC, UIR, SAS, CCI have no cooperation among infrastructure nodes so they have larger query delays, more Uplink requests, more congestions and lesser hit ratio. But, we have exploited the cooperation of neighboring GFAs in CGCI. So, cooperation has influenced effectively for hit ratio of data, i.e., data is validated and queries are replied from local components now. As P_{GFA} is the probability of finding data from neighboring GFA so, there is an increase in the hit ratio. As queries are replied from lower side of hierarchy, so upper side of the network is very less congested so it can scale the workload. Hence, cooperation effect can be seen directly on query response time, hit ratio, and Uplink requests.

6.2.3 Effect of Mean Query Arrival Time (MQAT)

When MQAT is smaller on server, i.e., more number of queries are generated per unit time, then cache miss ratio increases on vehicle side because most of the requested data items stored in the cache of vehicle are invalid due to limited size of cache and replacements. Also, the number of Uplink requests sent to server increases, which increases the query response time as queries are replied from server. This occurs in No Cache, UIR, SAS,CCI and ECCI techniques but, in the proposed scheme, due to neighbor GFAs and their cooperation, queried data can be validated from GFAs instead of server. So, majority of queried data requests are satisfied or validated from GFAs or through neighboring GFAs. The Uplink requests are sent only for non cached data query requests with reduced frequency in the proposed scheme which results a reduction in Uplink requests.

- *Impact on query response time-(QRT)* Fig. 6 shows the variation of Query Response Time with MQAT of NC, UIR, SAS, CCI, ECCI, and CGCI. All the schemes have higher AQRT for smaller values of MQAT. But, it decreases with an increase in MQAT in all the schemes. QRT is very large in NC case as it has no local cache to respond even to frequently accessed pages. It decreases sharply in case of UIR because, with such larger query rates, the data in cache becomes invalid and more Uplink requests are sent. In SAS, with higher query rates, the QRT becomes larger due to longer IRs and historical IR data. In CCI and ECCI, it has reduced and not varied sharply as it has considerations of hot data in cache and cooperation of location agent

Fig. 6 Variation of query delay with mean query arrival time



and HA. But, in CGCI, the variation of QRT is not much as it has cooperation of GFAs, queries can be replied instantly with updated page as data is sent along with IRs otherwise can be resolved from cooperative GFAs

- *Impact on Uplink requests* In Fig. 7, with higher MQAT, data in cache becomes invalid in peer test case techniques. Once any miss occurs for queried data then in these techniques data is brought only from the server. There is no any intermediate component or any cooperation to invalidate the data at lower ends to reduce the Uplink requests. Only in ECCI technique has lesser Uplink request as it's GFA with an enhancement of storing the popular hot data in it so no more Uplink requests are sent on server to validate a cache data. But, in case of CGCI, as query rate becomes higher, and hot data is considered which has more chances of being cached and due to cooperation among infrastructure components, data is validated from local devices So, lesser Uplink requests are sent to the server in hierarchy.
- *Impact on hit ratio* As shown in Fig. 8 with smaller MQAT the hit ratio of all the schemes is smaller. As in UIR, it is rising gradually with an increase in MQAT from 20 to 100 ms. In SAS, there is also gradual rise of hit ratio. But in CCI and ECCI, the hit ratio has raised sharply as it has considered hot data cached in the cache of vehicles, whose updates are stored at corresponding GFAs by HA. But in CGCI technique, hit ratio is good and is not varying sharply as neighboring GFAs has cooperation among themselves and data can be validated from corresponding GFA and after checking the meta data of Valid_Unified_List of GFAs.

6.2.4 *Impact of Cache Size on Average Query Delay, Uplink Requests, and Hit Ratio*

Figure 9 shows the average query delay, Uplink requests, and hit ratio by varying the cache size. As shown in the figure, with an increase in the cache size, there is a decrease in the average query delay in all the schemes. But, the decrease in the average query delay is more in the propose scheme as compared to the other schemes. This is due to the fact that proposed scheme used a cooperative approach from the nearest locations so that in case of

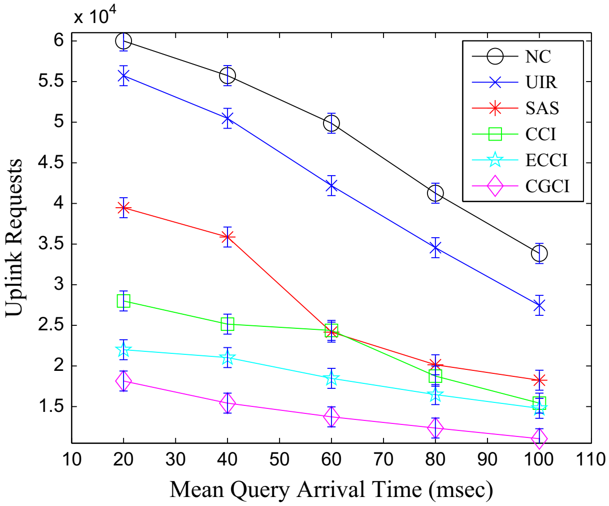


Fig. 7 Variation of no of Uplink requests with mean query arrival time

Fig. 8 Hit ratio (%) with mean query arrival time

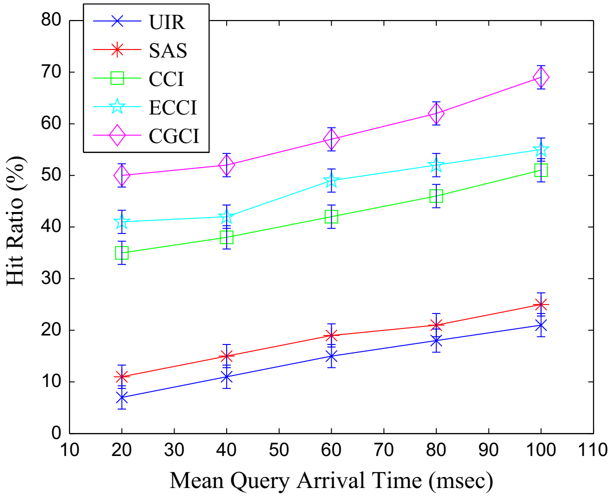
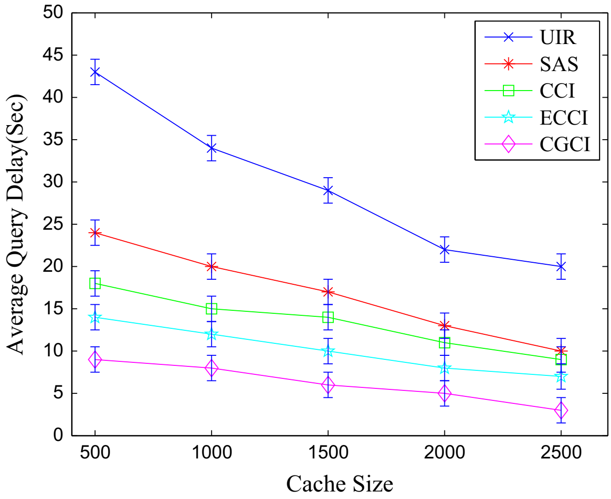


Fig. 9 Average query delay with variation in cache size



emergency situations, contents can be retrieved from the nearest locations as per the demand. This reduces the average query delay in the proposed scheme. None of the other existing schemes have used the cooperation from the nearest locations which causes a longer delay in accessing the contents from a particular location.

Figure 10 shows the variation of Uplink request with variation in the cache size. With an increase in the cache size, the Uplink requests are decreased but the proposed gateway cooperative scheme has the least Uplink requests as compared to the other schemes of its category. This is mainly due to the reason that there is a cooperation among the nodes while sharing the content for caching in the proposed scheme which is missed from most of

Fig. 10 Uplink requests with variation in cache size

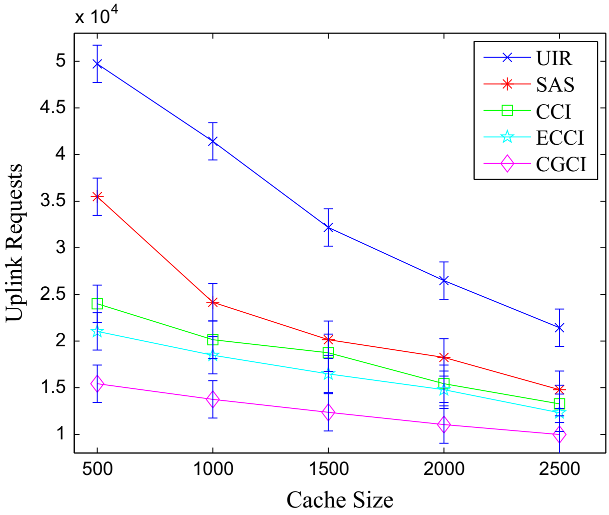
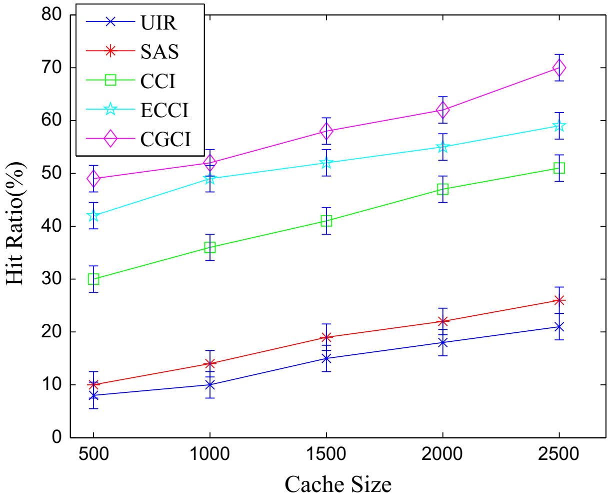


Fig. 11 Hit ratio with variation in cache size



the existing schemes in the literature which results an improved performance of the proposed scheme as compared to the other existing schemes.

Figure 11 shows the variation in hit ratio with varying cache size. As shown by the results obtained, with an increase in the cache size, cache hit ratio is highest in the proposed scheme as compared to the other schemes of its category. This is mainly due to the reason that the proposed scheme is cooperative with support from the nearest gateways which results an increase in the cache hit ratio with a decrease in broadcast time and Uplink requests. None of the other existing schemes in literature have used the cooperation from the gateway for sharing the contents among themselves. Moreover, in the proposed scheme using the cooperation from the gateway, cache invalidation is done periodically and the notification of the same is brodcasted which makes the proposed scheme effective as compared to the other existing schemes. Hence, there is an improvement in the cache hit ratio in the proposed scheme as compared to the the other existing schemes in literature.

Table 4 shows the percentage improvement of proposed scheme in comparison to CCI and ECCI schemes with respect to parameters Query Response Time, Up link requests, and Hit Ratio. We have considered approaches which are much closer to the proposed CGCI scheme, i.e., the behavior of the selected schemes is almost similar to the proposed scheme. Table 4 clearly shows that the proposed scheme performs better than the other schemes of its category with respect to metrics such as QRT, Uplink requests, and hit ratio.

We have considered all test case approaches with our CGCI scheme. Then, we have shown improvements in various parameters due to which the proposed technique has an upper edge from other peer techniques. Table 5 clearly shows that the proposed scheme has incorporated more features in it for performance improvement, better than the other schemes such as cooperation of infrastructure components, cooperated GFA, faster query response time and higher hit ratio due to consideration of hot data.

7 Conclusion

In this paper, we proposed a new cooperative GFA Cache Invalidation scheme called *Cooperative Gateway Cache Invalidation Technique* (*CGCI*) *for IVANET Environment* that overcomes the higher query response time and more number of Uplink requests problems found in the existing state-of-the-art schemes in literature. The proposed scheme includes stateful server, thats keeps track of updates of data and generates IR which is broadcasted to HA which in turn, unicast the updated data to GFA and to the vehicles. The proposed scheme has taken benefits of cooperation of neighboring GFAs which store the hot data fetched from server on any query reply. The proposed scheme works faster with cooperation of GFA, so queries are replied faster from cache or GFA which reduces the number of Uplink requests. To evaluate the performance of CGCI, exhaustive simulations are performed by varying the parameters such as-update rate query response time, Uplink

Table 4 Percentage improvement of CGCI over other existing schemes

Metrics	In comparison with CCI [27]	In comparison with ECCI [38]
QRT	1.12 times faster	043 times faster
Uplink requests	55.1 % lesser	31.2 % lesser
Hit ratio	24.4 % higher	11.7 % higher

Table 5 Features comparison table of various schemes

Features	CGCI [proposed]	UIR [28]	SAS [29, 30]	CCI [27]	ECCI [38]
Cooperation	✔	✗	✗	✔	✔
GFA cooperation	✔	✗	✗	✗	✗
Faster query response	✔	✔	✗	✗	✔
Higher hit ratio	✔	✗	✗	✗	✗

✔ Represents scheme has incorporated this feature

✗ Represents scheme has not incorporated this feature requests hit ratio, and broadcast time. Results obtained show the effectiveness of proposed CGCI scheme over other state-of-the-art schemes such as-UIR, SAS,CCI and ECCI.

In the future, we will test the performance of the proposed scheme in environments where mobility rate is higher than 30 km/h for urban areas. Moreover, P2P cooperation of the proposed scheme would also be evaluated with respect to different network parameters.

Acknowledgments We are thankful to all the anonymous reviewers for their valuable comments which helped us to improve the quality, content, and presentation of this paper.

References

1. Kumar, N., & Lee, J.-H. (2014). Peer-to-peer cooperative caching for data dissemination in urban vehicular communications. *IEEE Systems Journal*, *8*(4), 1136–1144.
2. Dua, A., Kumar, N., & Bawa, S. (2014). A systematic review on routing protocols for vehicular Ad Hoc Networks. *Vehicular Communications*, *1*(1), 33–52.
3. Kumar, N., Misra, S., & Obaidat, M. S. (2015). Collaborative learning automata-based routing for rescue operations in dense urban regions using vehicular sensor networks. *IEEE Systems Journal*. *9*(3), 1081–1090.
4. Kumar, N., Lee, J. -H., & Rodrigues, J. P. C. (2015). Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: Learning automata approach. *IEEE Transactions on Intelligent Transportation Systems*. *16*(3), 1148–1161.
5. Kumar, N., Chilamkurti, N., & Misra, S. C. (2015). Bayesian coalition game for internet of things: An ambient intelligence approach. *IEEE Communication Magazine*, *53*(1), 48–55.
6. Kumar, N., Chilamkurti, N., & Rodrigues, J. P. C. (2014). Learning automata-based opportunistic data aggregation and forwarding scheme for alert generation in vehicular ad hoc networks. *Computer Communications*, *39*(2), 22–32.
7. Kumar, N., Misra, S., Rodrigues, J. P. C., & Obaidat, M. S. (2014). Networks of learning automata: A performance analysis study. *IEEE Wireless Communication Magazine*, *21*(6), 41–47.
8. Kumar, N., & Lin, C. C. Reliable multicast as a Bayesian coalition game for a non-stationary environment in vehicular Ad Hoc Networks: A learning automata based approach. *International Journal of Ad Hoc and Ubiquitous Computing* (in press).
9. Kumar, N., Iqbal, R., Misra, S., & Rodrigues, J. J. P. C. (2015). Bayesian coalition game for contention-aware reliable data forwarding in vehicular mobile cloud. *Future Generation Computer Systems*, *48*(7), 60–72.
10. Kumar, N., Tyagi, S., & Deng, D. J. (2014). LA-EEHSC: Learning automata-based energy efficient heterogeneous selective clustering for wireless sensor networks. *Journal of Networks and Computer Applications*, *46*(11), 264–279.
11. Kumar, N., Misra, S., & Rodrigues, J. J. P. C. (2015). An intelligent approach for building a secure decentralized public key infrastructure in VANET. *Journal of Computer and System Sciences*, *81*(6), 1042–1058.
12. Kumar, N., Misra, S., Rodrigues, J. J. P. C., & Obaidat, M. S. (2015). Coalition games for spatio-temporal big data in internet of vehicles environment: A comparative analysis, *IEEE Internet of Things Journal*, p. 99.
13. Kumar, N., Rodrigues, J. J. P. C., & Chilamkurti, N. (2014). Bayesian coalition game as-a-service for content distribution in internet of vehicles. *IEEE Internet of Things Journal*, *1*(6), 544–555.
14. Bali, R., Kumar, N., & Rodrigues, J. J. P. C. (2014). Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. *Vehicular Communications*, *1*(3), 134–152.
15. Dua, A., Kumar, N., & Bawa, S. (2014). QoS-aware data dissemination for dense urban regions in vehicular Ad Hoc Networks. *Mobile Networks and Applications* (in press).
16. Kumar, N., & Kim, J. (2013). Probabilistic trust aware data replica placement strategy for online video streaming applications in vehicular delay tolerant networks. *Mathematial and Computer Modelling*, *58*(1/2), 3–14.
17. Dua, A., Kumar, N., Bawa, S., & Rodrigues, J. J. P. C. (2015). An intelligent context-aware congestion resolution protocol for data dissemination in vehicular Ad Hoc Networks. *Mobile Networks and Applications*, *20*(2), 181–200.

18. Kumar, N., Rodrigues, J. J. P. C., Lloret, J., & Dua, A. (2015). Replication-aware data dissemination for vehicular Ad Hoc Networks using location determination. *Mobile Networks and Applications*, *20*(2), 251–267.
19. Kumar, N., Kaur, K., Misra, S. C., & Iqbal, R. (2015). An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud. Peer-to-Peer Networking and Applications, 1–17.
20. Yang, Q., Lim, A., Li, S., Fang, J., & Agrawal, P. (2010). ACAR adaptive connectivity aware routingfor vehicular Ad Hoc Networks in city scenarios. *Mobile Networks and Applications*, *15*(1), 36–60.
21. Diacui, C., & Berkenbroc, M. (2008). Supporting cache coherence in mobile cooperative system. In: Seventh IEEE international symposium on network computing and applications, pp. 240–243.
22. Chuang, J. P., & Chiu, Y. (2008). Constructing efficient cache invalidation schemes for mobile environments. In: Third international IEEE conference on signal-image technology and internet based system, pp. 281–288.
23. Li, G. S., Wang, W. L., & Yao, X. W. (2012). An adaptive and opportunistic broadcast protocol for vehicular Ad Hoc Networks. *International Journal of Automation and Computing*, *9*(4), 378–387.
24. Kumar, N., Misra, S., Chilamkurti, N., Lee, J., & Rodrigues, J. J. P. C. (2015). Bayesian coalition negotiation game as a utility for secure energy management in a vehicles-to-grid environment. *IEEE Transactions on Dependable and Secure Computing*. doi:10.1109/TDSC.2015.2415489 (in press).
25. Kumar, N., Zeadally, S., Rodrigues, J. J. P. C. QoS-aware hierarchical web caching scheme for online video streaming applications in internet-based vehicular Ad Hoc Networks. *IEEE Transactions on Industrial Electronics*. doi:10.1109/TIE.2015.2425364 (in press).
26. Barbara, D., & Imielinksi, T. (1994). Sleepers and workaholics: Caching strategies for mobile environments. In: Proceedings of ACM, SIGMOD, International conference on Management of Data, pp. 1–12
27. Sunho, L., Chansu, Yu., & Das, Chita R. (2012). Cache invalidation strategies for internet-based vehicular Ad Hoc Networks. *Computer Communications*, *35*(6), 380–391.
28. Cao, G. (2000). A scalable low-latency cache invalidation strategy for mobile environments. In: Proceedings of ACM, 6th International conference for Mobile Computing and Networking, pp. 200–209.
29. Safa, H., Aartail, H., & Nahhas, M. (2010). A cache invalidation strategy for mobile networks. *Journal of Network and Computer Applications*, *33*(2), 168–182.
30. Safa, H., & Aartail, H. (2008). COACS: A cooperative and adaptive caching system for MANETs. *IEEE Transactions on Mobile Computing*, *7*(8), 951–977.
31. Tiwari, R., & Kumar, N. (2012). A novel hybrid approach for web caching. In: proceedings of 6th IEEE International Conference on Innovation Mobile and Internet Services in Ubiquitous Computing (IMIS), (pp. 512–517). Palermo, Italy.
32. Cai, T., & Ooi, H. (2001). An evaluation of cache invalidation strategies in wireless environments. *IEEE Transactions on Parallel and Distributed Systems*, *12*(8), 789–807.
33. Yang, K., Ou, S., Chen, H.-H., & He, J. (2007). A multihop peer-communication protocol with fairness guarantee for IEEE 802.16-based vehicular networks. *IEEE Transaction on Vehicular Technology*, *56*(6), 3358–3370.
34. Wischhof, L., Ebner, A., Rohling, H., Lott, M., & Halfmann, R. (2003). Adaptive broadcast for travel and traffic information distribution based on intervehicle communication. In: Proceedings of IEEE Intelligent Vehicle Symposium (IV'03), pp. 6–11.
35. Korkmaz, G., Ekici, E., & Ozguner, F. (2004). Urban multi-hop broadcast protocols for inter-vehicle communication systems. In: Proceedings of ACM VANET, p. 765.
36. Shafiee, K., & Leung, V. C. M. (2009). A reliable robust fully ad hoc data dissemination mechanism for vehicular network. *International Journal of Advanced Science and Technology*, *2*(3), 3–62.
37. Cao, G., Yin, L., & Das, C. (2004). Cooperative cache based data access in Ad Hoc Networks. *Computer*, *37*(2), 32–39.
38. Shen, H., Joseph, M. S., Kumar, M., & Das, S. (2005). A scheme for cooperative caching in mobile peer to peer network. In: IEEE international parllel and distributed processing symposium, pp. 57–64.
39. Chan, E., & Liw, Lus. (2012). Movement prediction based cooperative caching for location dependent information service in mobile Ad hoc network. *The Journal of Supercomputing*, *59*(1), 297–322.
40. Yin, L., & Cao, G. (2006). Supporting cooperative caching in Ad Hoc Networks. *IEEE Transaction on Mobile Computing*, *5*(1), 77–89.
41. Kumar, P., Chauhan, N., Awasthi, L., & Chand, N. (2010). Proactive approach for cooperative caching in mobile Ad hoc networks. *International Journal of Computer Science*, *7*(8), 21–27.
42. Mrmol, F., & Prez, G. (2012). TRIP: A trust and reputation infrastructure-based proposal for vehicular Ad Hoc Networks. *Journal of Network and Computer Application*, *35*(3), 934–941.

43. Dubey, A., & Sharma, S. (2011). A cache invalidation scheme through data classification in IVANET. *International Journal of Computer Application*, *25*(7), 54–57.
44. NS2 simulator. (2008). http://www.insi.edu/nsnam/ns
45. Issariyakul T., & Hossain E. (1995). Introduction to network simulator NS2, Springer, London.

Rajeev Tiwari is working as an Assistant Professor in UPES, Dehradun (India). He is pursuing Ph.D. from Thapar Univeristy, Patiala (Punjab). He has more than 10 years of experience in teaching and research. He has served many institutes of repute in teaching and research during last 10 years. His broad area of research are MANET, VANET, QoS in wireless networks and adaptive, dynamic and cooperative cache invalidation techniques. He is also an expert in simulation and design of scenarios on NS2, SUMO and MOVE. He is a member of IEEE,IACSIT and IAENG. He is a reviewer in SCI indexed journal like AHWSN. He is TPC member of many international conferences from Springer, IEEE and Elsevier.

Neeraj Kumar is working as Associate Professor in Department of Computer Science and Engineering, Thapar University, Patiala (Punjab), India. He received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra (India) and PDF from Coventry University, Coventry, UK. He has more than 100 scholarly research publications in peer reviewed journals and conferences including IEEE, Elsevier, Springer, and John Wiley. Some of his research findings can be found in top-cited journal such as- IEEE TIE, IEEE TDSC, IEEE TITS, IEEE TCE, IEEE Netw., IEEE Wireless Comm., IEEE Netw., IEEE SJ, FGCS, JNCA, TJ, and ComCom. His research is focused on mobile computing, parallel/distributed computing, multiagent systems, service oriented computing, routing and security issues in wireless adhoc, sensor and mesh networks. He is leading the Mobile Computing and Distributed System Research Group. Prior to joining Thapar University, Patiala, he has worked in SMVDU, Katra, HEC Jagadhri and MMEC Mullana, Ambala, Haryana, India. He has delivered invited talks and lectures in various IEEE international conferences in India and abroad. He has organized various special sessions in international conferences in his area of expertise in India and abroad. He is TPC of various IEEE sponsored conferences in India and abroad. He is reviewer/ editorial board of various international journals. He is guest editor of special issue of 6 international journals. He is senior member of ACEEE and IACSIT.