Journal of Computer Science 10 (4): 614-622, 2014
ISSN: 1549-3636
© 2014 Science Publications
doi:10.3844/jcssp.2014.614.622 Published Online 10 (4) 2014 (http://www.thescipub.com/jcs.toc)

CLUSTER BASED CACHE CONSISTENCY USING AGENT TECHNIQUE IN MOBILE ENVIRONMENT

[1]Shanmugarathinam Ganesarathinam and [2]Vivekanandan Kanniappan

[1]Department of Computer Science, Bharathiar University, India
[2]School of Management, Bharathiar University, India

Received 2013-09-22; Revised 2013-10-09; Accepted 2013-12-19

ABSTRACT

Caching is an essential mechanism in mobile environment. Mobile node makes access dynamic changing data object to their server and to get the interested data to keep their own as local copies in the cache. If the server database is updated, the data cached in mobile client is invalidated and it becomes inconsistent between client and server. In this study we design the cluster based caching technique in mobile node, this technique improves data accessibility, reduces the query latency and easy to maintains cache consistency in mobile environment. Designed a Cluster head node in between the server and the client using agent technique. Cluster head is selected nearest to the center of the grid and full battery power among other nodes in the mobile environment. Simulation is done on NS2, result show that reduces update delay, reduces Query delay, high throughput, high energy level when compared with existing approaches Distributed Cache Invalidation Method (DCIM).

Keywords: Mobile Database, Wireless Networks Database Cache, Threads Agent, Low Bandwidth Networks

1. INTRODUCTION

Mobile background uses manageable computing devices eg laptops, PDAs and wearable computers. Example of applications are sales force automation, order entry, e-mail, financial and news services, insurance companies, emergency services (police, medicals), traffic control, taxi dispatch. A mobile background, when data updated at the server, the client hosts must get the latest updated information otherwise it becomes a invalidate their local cache data, if not the host would continue to answer queries with the cached incorrect data. Mobile environment has rigorous limitations in network resources, such as bandwidth and connectivity. In this issue, Mobile applicastion data are kept in cached at clients to increase performance, data availability and reliability. Although a number of studies have been made in this subject, few researchers focused on mobile data access. We design a node as cluster head. It is between the server and client. Whenever the server data is updated Agent will starts the synchronization with cluster head and the client using agent technique. Some of the clients wake up from sleep mode and immediately request the cluster Head for the updated data and need not request the server. So it reduces the work load in the main server.

1.1. Related Work

Many attempts have been made to solve the problem of cache consistency maintenance in mobile environment and it has its own drawback. In this study (Artail *et al*., 2008) cooperation based database caching system. In this method query delay and bandwidth utilization more. In this approach (Shanmugarathinam and Vivekandan, 2011) middle server betwee main server and client. But more workload on server. In this study Shanmugarathinam and Vivekandan (2013) queing model apparoach raduce the traffic but still more work load on server. In this method (Madhukar and Alhajj, 2006) is used to reduce the bandwidth requirement, the server transmits in one of the three modes slow, fast and super-fast. Drawback of this method, if the mode changes to slow, so the client has to

Corresponding Author: Shanmugarathinam Ganesarathinam, Department of Computer Science, Bharathiar University, India

wait for long time to utilize cached data. This study (Huang *et al*., 2010) Flexiable combination of push and pull algorithm. drawback of this method is latency more. In this study (Kuppusamy and Thirunavukkaru, 2012) cluster based approach for cache consistency drawback of this method long query latency. More network traffic. SSUM (Artail and Artail, 2010) the server sends data updates to the Cache Node (CN) request Mobile host that desires a data item send its request to its nearest Query Directory (QD). If this QD finds the query in its cache, it forward the request to the CN caching the items, which in turn sends the item to the requesting mobile host. Otherwise it forward, it to its nearest QD, If the request traverses all QDs without being found, a miss occurs and it gets forwarded to the server which sends the data item to the request mobile host. The drawback in this method is latency more, bandwidth is wasted. This study proposes DCIM (Fawaz and Artail, 2012) a client-based cache consistency scheme that is implemented for caching data items in MANETs, namely COACS, where special nodes cache the queries and the addresses of the nodes that store the responses to these queries. IT is a pull-based algorithm that implements adaptive TTL, piggybacking and prefetching, Cached data items are assigned adaptive TTL values that correspond to their update rates at the data source, where items with expired TTL values are grouped in validation requests to the data source to refresh them, whereas, unexpired ones but with high request rates are perfected from the server. The shortcoming in this approach the client wants wait for the server replay for the Invalidation report, Query delay and more latency.

2. MATERIALS AND METHODS

2.1. Cluster Active Node Selection

One of the nodes in the cell is selected as cluster head, this node will be in the midpoint of the cell and it have highest battery power among other nodes in the cluster node. Cluster Head (node's coordinator) is assumed by location method. The distance between each node's coordinates to the midpoint of grid cell is calculated. Among all other nodes in the cell, the node which is centre of the cell is chosen to be Cluster head. Cluster head maintains a list which maintains the information of all other nodes in the group. In a cluster, the number of hops between any two nodes is not more than two. Cluster member is just like a mobile node with data cache. The node which is common to two cluster heads is elected as a gateway. Gateway is used for providing the communication between two cluster heads. Whenever a node requests for the data, first it has to be checked in the cluster head list. If it is not available in the list of cluster head then the cluster head forwards the requested data item to the other cluster via gateway shown in **Fig. 1**. The cluster head is election re-election is done in at periodic interval. The re-election process will place among the cluster members only. The node which is elected as a cluster head should inform to other server and other cluster node in the network.

2.2. CBCCAT Process

Information search operation, mainly deals with locating and fetching the data item requested by the client from the cache using Agent technique. This Information search includes 4 cases as shown in **Fig. 2**.

Case 1:

Local hit: When copy of the requested data item are avilable in local cache of the requester, the data item is retrieved to serve the query and no need request to server or other node is necessary using agent technique.

Case 2:

Cluster hit: When the requested data item is not avialble in local node cache and the requester sends a request to the Cluster head and check with local database returns the data to the node, If data not available check with other cluster head node through gateway using agent technique.

Case 3:

Remote hit: When the data is found in other of cluster head than home cluster of the requester, the data retrieve and store in cluster head for future use and return the data to the node using agent.

Case 4:

Global hit: Data item is retrieved from the server. When the client data request comes to the mobile node, first it checks in the local hard disk of mobile node i.e., local cache of mobile node. If it is available in the local cache it sends back the reply to the client. Otherwise the request is forwarded to the cluster head and check with database. If the cluster head has the requested data, cluster head gives back to the requester by giving the cluster member id. If it is not available within the cluster then the request is forwarded to the other cluster through gateway. The request is processed the same way and sends back the reply to the requester. Otherwise the request is reached to the data center, the data center processes the data request and sends backs the requested information to the client via multi hop communication.

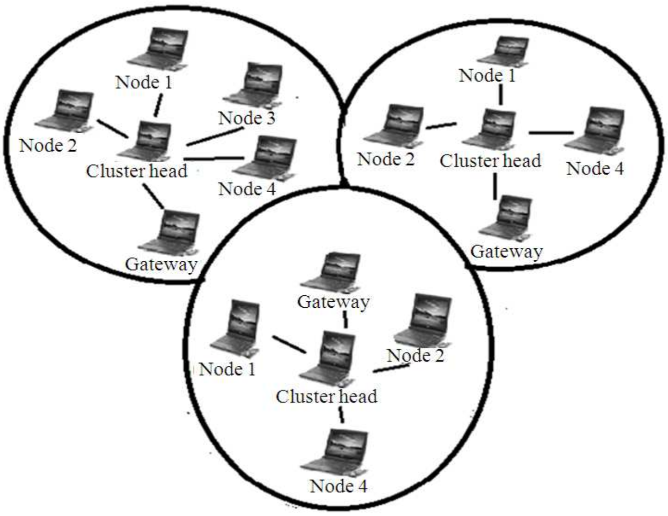

Fig. 1. Cluster communication

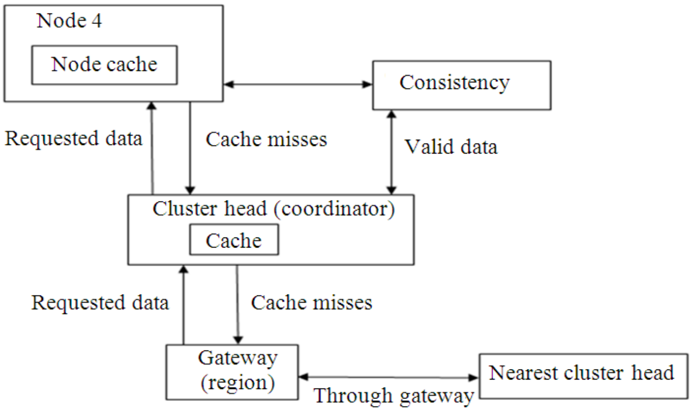

Fig. 2. CBCCAT process

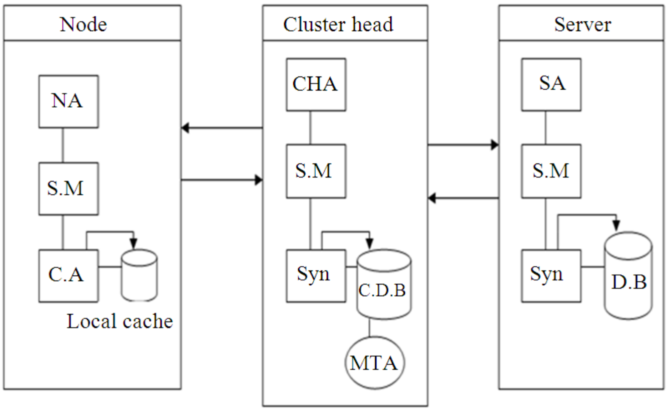

Fig. 3. CBCCAT architecture

Table 1. Summary of cluster based cache consistency

Components	Node	Cluster node
NA-Node Agent	To listen for response, to synchronize and prompt with Active node	To listen for request, to synchronize and prompt users for login authentication
SM-Security Module	To authenticate user identity and to control user access based on granted privileges	To verify user ID and to keep track of the client node
CA-Cache Agent	To check the data available in local cache	To check the valid data in the active node
MTA-Migration Threads Agent	To update the data in the local cache	When it becomes invalid data in mobile cache, it starts updating data with cluster head
Synch module	To work together with the synchronization of active node agent.	To work together with the synchronization module at the node's agent to propagate changes to and from the server
DB-Data Base	To store the data in local cache	To store database

2.3. System Architecture Fig. 3 and Table 1

- Key Terms
- NA-Node agent
- SM-security module
- CA-Cache Agent
- MTA-Migration threads Agent
- ANA-Active Node Agent
- DB-Data base

2.4. Cluster Based Using Agent Based Technique

Some of the node are grouped as cluster, mobile node data cached is placed in cluster head (coordinator). The cluster head are near to the other node. So the communication cost, energy consumption are very less, easy to update the cache data and easily maintain consistency. If Data requested is not available in local cache, the node agent send the broadcast request to the cluster head agent. cluster head agent receive the packet, search in the cache and send the acknowledge data to the node agent. The Cluster head satisfy the nearest node request Advantage is low cost for communication and reduces the network traffic in mobile networks.

Cluster head have a log and Agent Model maintains consistency between the Server data and mobile node cache data. When a data d_x is retrieved by a node log is created to indicate data is valid or not. When the Server updated data object d_x server agent starts broadcasts and synchronizes with cluster head Agent to make cache data object reliable. During this process a log maintained in server is compared with recent log of cluster head. If so there is a need of Updation, it processes to perform update function(s). In mobile environments a node Cache is one of two states. (i) Awake or (ii) Sleep. If a node is awake an internal request is shared between node Agent at cluster head agent and node Agent at client to ensure that data object is updated. If there is an Updation the SynchM of server synchronizes withSynchM of cluster head and mobile node in order to make as valid data object.

The data objects of a node in the sleep state are unaffected until it wakes up. When a node wakes up a new agent upon is created which holds last accessed log, this log passes to the cluster head, on receiving upon the log it compares with previous log maintained by it. If it is invalid data cache the agent of cluster head cache starts the synchronizes with node for the updating data.

2.5. Performance Evaluation

Simulation was done using Ns2 for the mobile environment concept. The simulation time is 2000 sec network size 400×400 m, wireless bandwidth 2 Mb/s. Node speed 2(m/s), size of data items 10 kb and channel capacity of each mobile host has 1 Mbps. The MAC protocol uses 802.11. We take a number of nodes 10-100, query request rate 5-50, Data update rate 5-50.

2.6. Update Delay

Table 2 and 3, simulation value shows that the DCIM method is more update delay compare with proposal model CBCCAT in varies with number of nodes and query request rate.

2.7. Query Delay

Table 4 simulation value shows that the DCIM method is more Query delay compare with proposal model CBCCAT in varies with Data update rate.

3. RESULTS

The simulation result show in **Fig. 4, 6 and 8** explain that DCIM model update delay is more when compared with proposal CBCCAT model which varies with number of nodes, data update rate and query request rate. **Figure 5, 7 and 11** shows that DCIM model have more query delay compare with CBCCAT model, which varies with number of nodes, data update rate and query request rate. **Figure 9 and 10** result shows that average energy and throughput increase in value in the proposal CBCCAT model, when compared with DCIM model.

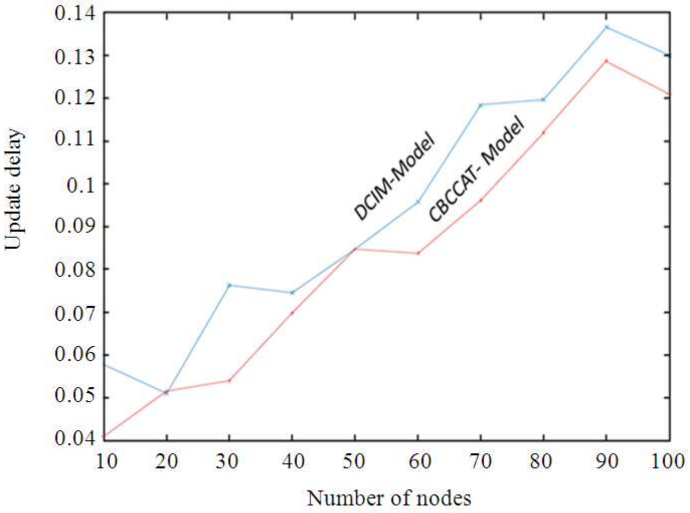

Fig. 4. Update delay varies with number of nodes

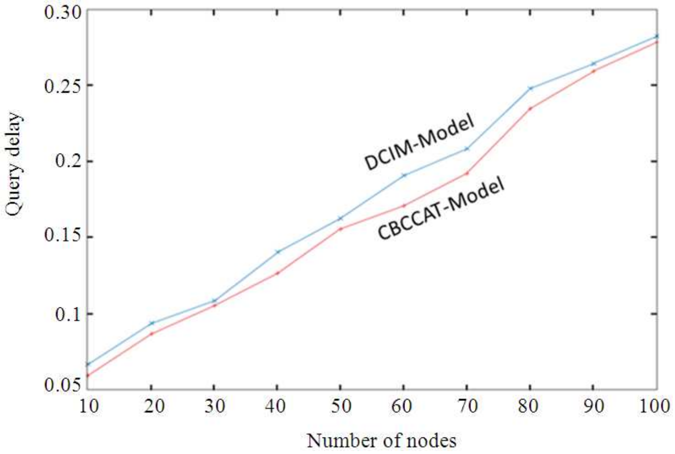

Fig. 5. Query Delay varies with number of nodes

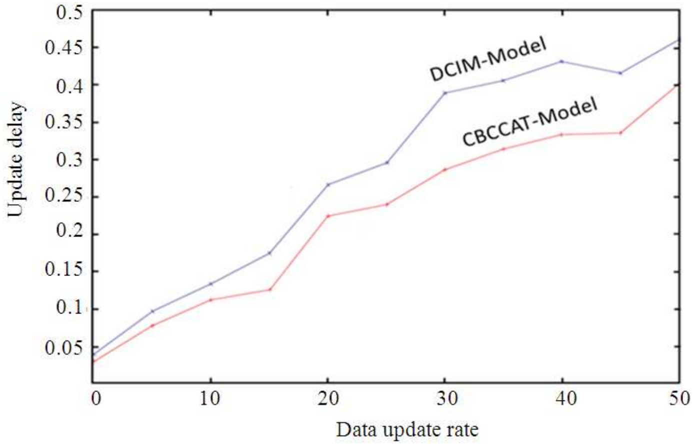

Fig. 6. Update delay varies with Data update rate

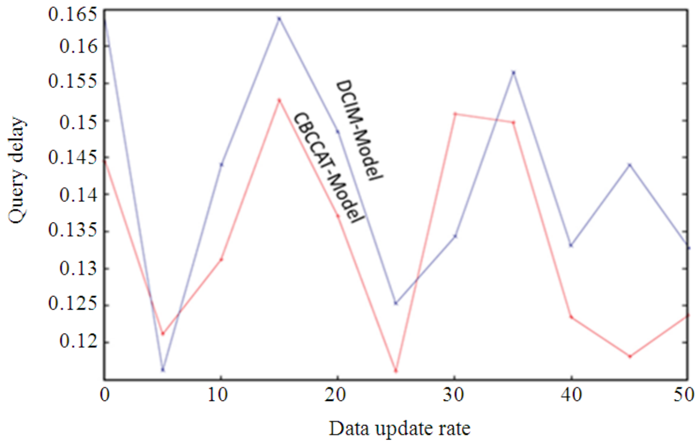

Fig. 7. Query delay varies with data update rate

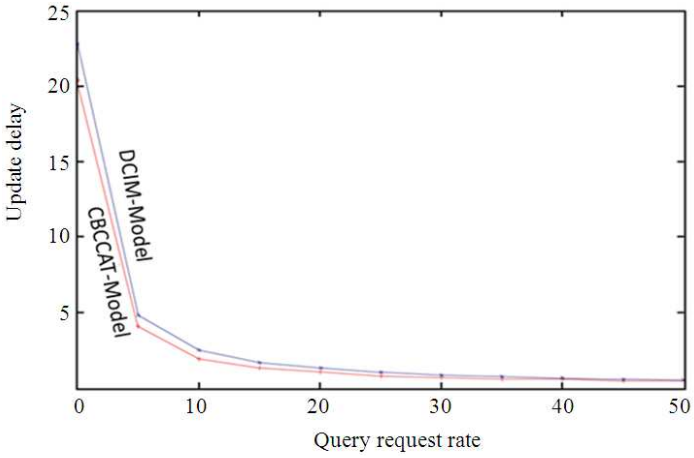

Fig. 8. Update delay varies with query request rate

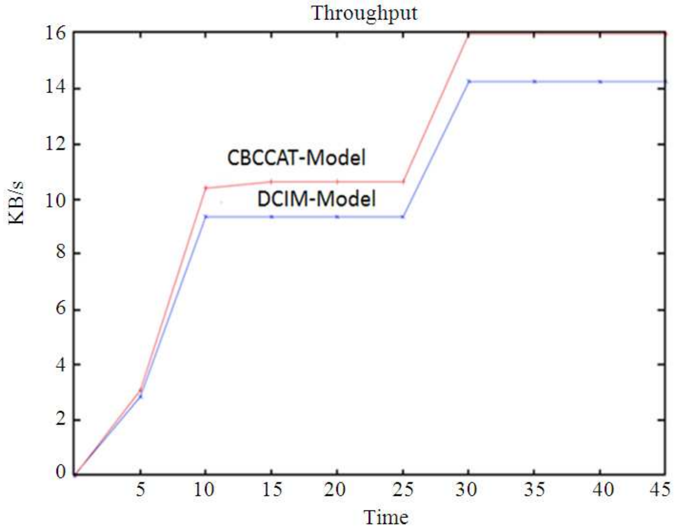

Fig. 9. Throughput

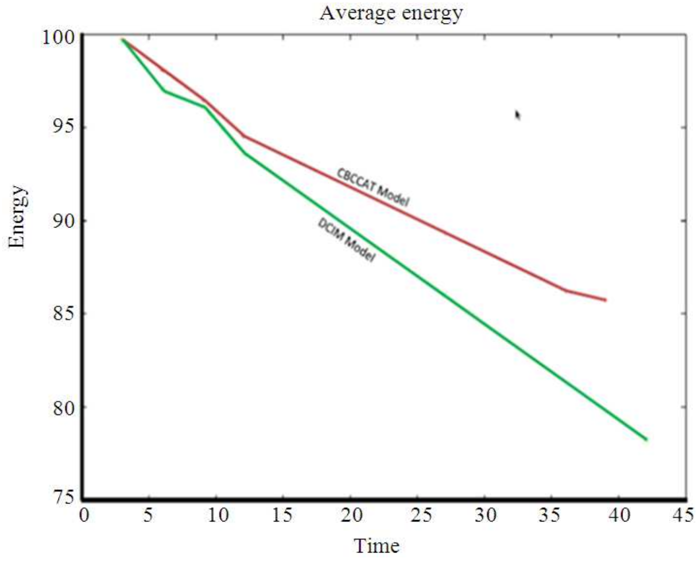

Fig. 10. Average energy

Table 2. Update delay varies with number of nodes

Number of Nodes	10	20	30	40	50	60	70	80	90	100
CBCC-AT	0.0419	0.0429	0.0655	0.0614	0.0735	0.0874	0.0982	0.1164	0.1295	0.1379
DCIM	0.0578	0.0511	0.0763	0.0747	0.0849	0.0958	0.1184	0.1196	0.1365	0.1300

Table 3. Update delay varies with query request rate

Query Request Rate	5	10	15	20	25	30	35	40	45	50
CBCC-AT	3.9225	2.1330	1.3223	0.9958	0.8320	0.7097	0.6327	0.5136	0.4952	0.4547
DCIM	4.7621	2.5171	1.6890	1.2959	1.0354	0.8579	0.7635	0.6164	0.5593	0.5035

Table 4. Query delay varies data update rate

Data update rate	5	10	15	20	25	30	35	40	45	50
CBCC-AT	0.0624	0.1198	0.1405	0.1812	0.2664	0.2182	0.2511	0.3550	0.4646	0.3565
DCIM	0.0974	0.1341	0.1751	0.2671	0.2962	0.3891	0.4069	0.4319	0.4164	0.4617

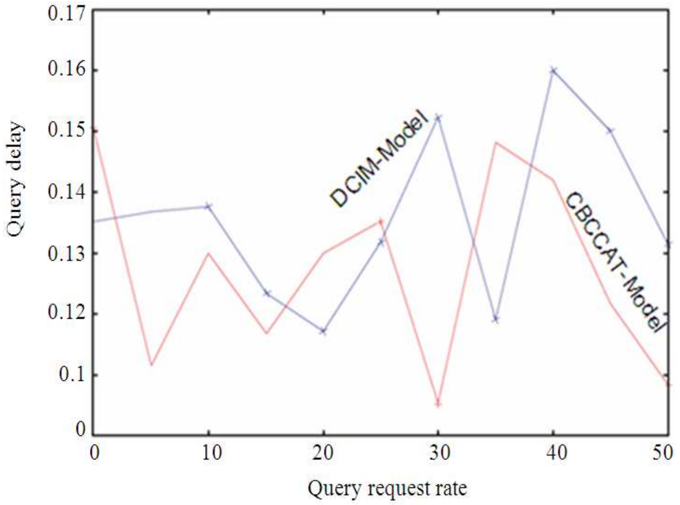

Fig. 11. Query delay varies with query request rate

4. DISCUSSION

Simulation has been conducted for the nodes varying from 10 to 100 for the update delay and data update rate 0 to 50 for the query delay. From the simulation result, it is inferred that there is a decrease in update delay of 10.52%, decrease in query delay is 9.51%, increase in throughput is 10.1% varies in time 5 to 45, increase in average energy is 2.1% varies in time 5 to 45 compared with DCIM model. The result exposed that proposed CBCCAT model proved better in performance when compared with the existing approach of DCIM model.

5. CONCLUSION

In this study, we proposed a cluster Based cache consistency maintenance in mobile environment using Agent Technique (CBCCAT), key features as stated earlier are low cost for communication, energy consumption is very low and easy to maintain the cache consistency,increase the local cache hit ratio,avoiding every request send to server so the network traffic reduces. Simulation results show that the proposed algorithm has significantly better performance than earlier approaches.

6. REFERENCES

Artail, H. and H. Artail, 2010. SSUM: Smart server update mechanism for maintaining cache consistency in mobile environments. IEEE Trans. Mobile Comput., 9: 778-795. DOI: 10.1109/TMC.2010.18

Artail, H., H. Safa, K. Mershad, Z. Abou-Atme and N. Sulieman, 2008. COACS: A cooperative and adaptive caching system for MANETs. IEEE Trans. Mobile Comput., 7: 961-977. DOI: 10.1109/TMC.2008.18

Fawaz, K. and H. Artail, 2012. DCIM: Distributed cache invalidation method for maintaining cache consistency in wireless mobile networks. IEEE Trans. Mobile Comput., 12: 680-693. DOI: 10.1109/TMC.2012.37

Huang, Y., J. Cao, B. Jin and X. Tao, 2010. Flexible cache consistency maintence over wireless adhoc networks. IEEE Trans. Parallel Distributed Syst., 21: 1150-1161. DOI: 10.1109/TPDS.2009.168

Kuppusamy, P. and K. Thirunavukkaru, 2012. Cluster based cooperative caching technique in mobile adhoc network. Eur. J. Sci. Res., 69: 337-349.

Madhukar, A. and R. Alhajj, 2006. An adaptive energy efficient cache invalidation scheme for mobile databases. Proceeding of the ACM Symposium on Applied Computing, Apr. 23-27, ACM Press, New York, USA., pp: 1122-1126. DOI: 10.1145/1141277.1141545ss

Shanmugarathinam, G. and K. Vivekandan, 2011. Recent research issues and methodology to improve the performance in mobile computing. Int. J. Res. Rev. Comput. Sci.

Shanmugarathinam, G. and K. Vivekandan, 2013. Multiple servers-queue model for agent based technology in cache consistence maintenance of mobile environment. Int. J. Comput. Sci., 10: 507-511.