South Asian Journal of Engineering and Technology Vol.2, No.15 (2016) 161–168

ISSN No: 2454-9614

Cache Invalidation Mechanisms for maintaining Cache Consistency in Wireless Mobile Environments

P.S Prakashkumar[a], R. Naraesh[a*], S. Arunprasath[a,b], R. Dhamodharan[a,b,c]

[a)] *Department Of Computer Science and Engineering, K.S.R Institute for Engineering and Technology, Tiruchengode, Tamilnadu, India.*
[b)] *Department Of Computer Science and Engineering, K.S.R Institute for Engineering and Technology, Tiruchengode, Tamilnadu, India.*
[c)] *Department Of Computer Science and Engineering, K.S.R Institute for Engineering and Technology, Tiruchengode, Tamilnadu, India.*
[d)] *Department Of Computer Science and Engineering, K.S.R Institute for Engineering and Technology, Tiruchengode, Tamilnadu, India.*

*Corresponding Author: P.S.Prakashkumar
E-mail: psprakashkumar39@gmail.com

Received: 15/11/2015, Revised: 27/12/2015 and Accepted: 05/03/2016

Abstract

On-demand data access in client-server wireless networks is an important support to many interesting mobile computing applications. Caching frequently accessed data by mobile clients can conserve wireless bandwidth and battery power, at the expense of some system resources to maintain cache consistency. Several mechanisms have been proposed in the literature to address the challenging problem of cache consistency in cellular wireless networks. This survey paper discuss about some of the mechanisms that have been proposed by researchers in the direction of data consistency maintenance between the clients and the data servers.

Keywords – cache, consistency, invalidation, MANET, Access point.

Reviewed by* **ICETSET'16 *organizing committee*

1. Introduction

Homes, Telecommunication networks and Enterprise installations avoid the costly process of introducing cables for building a connection between various equipment locations using wireless networking **[1]**. The advances in hardware and wireless technologies such as Wi-Fi and the proliferation in the usage of mobile devices have made wireless networks ubiquitous. The falling cost of both communication and mobile has made mobile computing commercially affordable to both business users and private consumers **[1]**. In the near future, people with battery powered mobile terminals (MTs) can access various kinds of services over wireless networks at any time or any place. Transmission over wireless **channel** is expensive due to limited bandwidth, transmission latency and large energy consumption by mobile clients **[1]**.

Caching frequently accessed data, is an important technique commonly used to reduce these costs, Cache invalidation strategy is used to ensure that the data items cached by a mobile client are consistent with those stored on the server **[9].** In a Wireless network, data caching is essential as it reduces contention in the network, increases the probability of nodes getting desired data, and improves system performance. The major issue that faces cache management is the maintenance of data consistency between the cache client and the data source **[11].** All messages sent between the server and the cache are subject to delays, thus, impeding consistency by download delays, that are considerably noticeable and more severe in wireless mobile devices.

Client caching means that each client caches some of its received data items in the local cache **[12].** If a new query arrives at a later time, requesting for the same data, the cached copy can be used without going through the server again. This reduces the amount of uplink traffic, possibly leading to reduce bandwidth requirement. In practice, data items are updated from time to time in an asynchronous manner **[9].** A client cannot rely on its cached copies to answer queries forever. Instead, the validity of the caches should be first ensured. All cache consistency algorithms are developed with the same goal in mind to increase the probality of serving data items from the cache that is identical to those on server **[11]**. However, achieving strong consistency, where cached items are identical to those on the server, requires costly communications with the server to validate (renew) cached items, considering the resource limited mobile devices and the wireless environments they operate in. Wireless networks are classified as Infrastructure based and ad hoc based **[1].** In Infrastructure based networks, Access points (AP) forwards messages that mobile hosts send or receive. Mobile devices cannot communicate directly in this type of networks. Routing of packets is done throughAP's **[1]**.

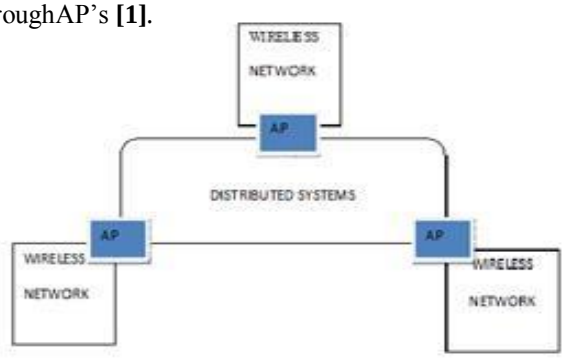

Fig. 1. Infrastructure based networks.

In ad hoc networks, there is no fixed infrastructure. Mobile devices cannot communicate directly in this type of networks. Routing of packets is done through intermediate nodes. It is an independent system of mobile nodes connected by wireless links forming a short, live, on-the-fly networkeven when access to the Internet is unavailable **[1]**. Nodes in ad-hoc network generally operate on low power battery devices. These nodes can function both as hosts and as routers. As a host, nodes function as a source and destination in the network and as a router, nodes act as intermediate bridges between the source and the destination giving store-and-forward services to all the neighbouring nodes in the network. Easy deployments, speed of development, and decreased dependency on the infrastructure are the main reasons to use ad-hoc network. Because of the different characteristics of the two

types of networks, caching mechanisms and cache invalidation schemes will not be same. Cache consistency algorithms should develop according to the different types of networks. Client caching is widely recommended technique to conserve wireless bandwidth.

Fig. 2. Ad-hoc network

2. Cache Invalidation Methods for Ad-hoc Networks

All Mobile nodes in infrastructure based networks directly contact with the Access points. So, the cache consistency is easier in Infrastructure based networks. But in Ad-hoc networks some mobile nodes cannot directly contact with the Access points. The Cache consistency is difficult in Ad-hoc networks.

2.1 COACS (A Cooperative and Adaptive caching system)

COACS is a distributed caching scheme. Cache consistency is difficult in Ad-hoc networks compared to the Infrastructure based networks **[12]**. COACS introduces new architecture for cache consistency. COACS consists three types of nodes: Requesting node (RN), Query directory (QD), Caching node (CN). A QD's task is to cache queries submitted by the requesting mobile nodes, while the CN's task is to cache data items (responses to queries).

As shown in Fig. 3, RN sends request, QD forwards that request to the CN. CN checks whether the requested dataavailable or not. If the requested data is available then it forwards reply to the RN. If the CN does not have the requested data then request is forwarded to the Server

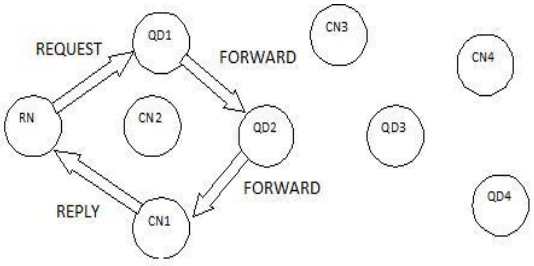

Fig. 3. COACS Design

2.2 DCIM (Distributed cache invalidation mechanism)

DCIM is a client based cache consistency mechanism that implements adaptive time to live (TTL), piggybacking, and prefetching, and provides near strong consistency capabilities **[11].** DCIM follows the COACS Architecture. As shown in Fig. 4, Requesting node (RN) sends request using Data request packet (DRP) packet. Query directory (QD) checks whether the requested data is available in caching node (CN). Then, CN returns reply

using Cache update request (CURP) **[11]**. 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. This validation requests send using CURP.

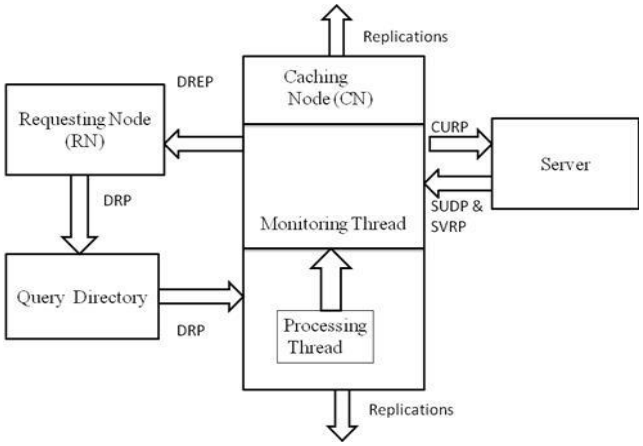

Fig.4. DCIM Architecture

Server gives validation reply using Server update data (SUDP) and Server validation reply (SVRP) **[11].**whereas un expired ones but with high request rates are prefetched from the server. Since COACS did not implement a consistency strategy, the system DCIM (Distributed Cache Invalidation Method) provides the several improvements: Enabling the server to be aware of the cache distribution in the MANET, Making the cached data items consistent with their version at the server, Adapting the cache update process to the data update rate at the sever relative to the request rate by the clients, With these changes, the overall design provides a complete caching system in which the server sends to the clients selective updates that adapt to their needs and reduces the average query response time **[11].**

2.3 SSUM (Smart server update mechanism)

SSUM is a server-based approach that avoids many issues associated with push-based cache consistency approaches. In SSUM, the server autonomously sends data updates to the CNs, meaning that it has to keep track of which CNs cache which data items **[15]**. This can be done using a simple table in which an entry consists of the id of a data item (or query) and the address of the CN that caches the data.

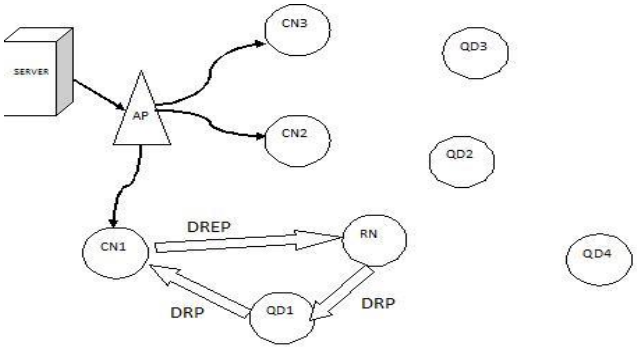

Fig. 5 SSUM Design

As shown in Fig. 5, a node that desires a data item sends its request to its nearest QD. If this QD finds the query in its cache, it forwards the request to the CN caching the Item, which, in turn, sends the item to the requesting node (RN)**[15]**. A Server sends update reports frequently to the CN's for indicating updation in the cached data items **[15].**

2.4 Zone based Cooperative Caching Scheme

One hop neighbour of a mobile client form a cooperative cache zone.Mobile client share cache with its neighbours lying in the zone **[10].**During cache discovery process when a data request is initiated it first looks for the data item in its own cache **[10].** If there is a local cache miss, the mobile client checks whether the data item is cached in other mobile hosts within its home zone. When a mobile host receives the request and has the data item in its local cache, it will send a reply to the requester to acknowledge that it has the data item. In case of a zone cache miss, the request is forwarded to neighbour along the routing path **[10].**

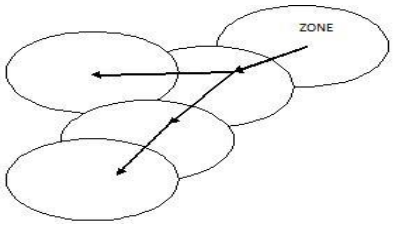

Fig. 6 Cache Discovery Process

2.5 COOP (A cooperative caching service in MANETs)

COOP is a novel cooperative caching scheme for on-demand data access applications in MANETs. The objective is to improve data availability and access efficiency by collaborating local resources of mobile nodes **[6]**. The cooperation of caching nodes is twofold. First, a caching node can answer the data requests from other nodes. Second, a caching node stores the data not only on behalf of its own needs, but also based on other nodes' needs **[6].** COOP addresses two basic problems for cooperative caching in MANETs **[6]:**

Cache resolution – how does a mobile device decide where to fetch a data item requested by the user?

Cache management – how does a mobile device decide which data item to place/purge in its local cache?

For cache resolution, COOP tries to discover a data source which induces less communication cost by utilizing historical profiles and forwarding nodes **[6]**. For cache management, COOP minimizes caching duplications between neighbour nodes and allows cooperative caches to store more distinctive data items to improve the overall performance **[6].**

The server has partial knowledge about the mobile node caches, and flag bits are used both at the server and the mobile nodes to indicate data updates. Such mechanisms necessitate server side modifications and overhead processing. More crucially, they require the server to maintain some state information about the MANET, which is costly in terms of bandwidth consumption especially in highly dynamic environments. So, The client based cache consistency schemes are most probably preferred. Table. 1 shows comparison between different cache invalidation schemes in ad-hoc networks.

Table 1. Comparison charts for cache invalidation schemes in ad-hoc networks.

	Message Traffic	Network Utilization	Load on Server/MSS	Good Performance during disconnections	Scalability of mobile clients	Latency	Serves the purpose of mobility
COACS	Yes	Yes	No	No	Yes	No	No
DCIM	Yes	Yes	No	No	Yes	No	No
SSUM	No	No	Yes	No	Yes	No	No
ZONE BASED	Yes	No	No	Yes	Yes	Yes	Yes
COOP	No	No	No	Yes	Yes	Yes	Yes

3. Caching Consistency for Infrastructure Based Networks

3.1 Invalidation Reports

This is a push-based data delivery carried out by data server. In this approach, the server broadcasts invalidation report in which the changed data items are indicated. Rather than querying the server directly regarding the validation of cached copies, the clients can listen to these IRs over the wireless channel, and use them to validate their local cache. The server can broadcast IRs periodically **[2]** or on-demand **[3]**. As shown in Fig. 2, A Cache in a client node contains copy of the Server data. A Server sends an invalidation report when the updation occurs in the server.

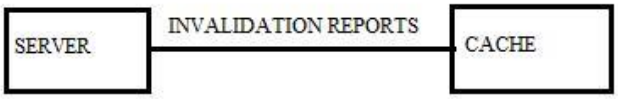

Fig. 7 Invalidation Reports.

3.2 Periodic Broadcast of Invalidation Reports

This technique **[2]** was proposed to ascertain that every mobile device receives the updated value of all data items that have been changed at the server. Since, IRs arrive periodically, clients can go to sleep any time and can retrieve the data when they wake up. The drawback of this technique was latency time that was incurred due to long IR from the server. Further studies were performed by **[5]** in which the IR size was optimized to decrease the latency time of cache validations in the mobile cache.

3.3 Sleepers and Workaholics [2]

This paper was proposed to maintain data consistency between servers and the MUs by periodically sending Invalidation Reports (IR) from the data server to MUs for a data item. The server agrees to the obligation of notifying about the data items that have changed during last *w* seconds. Every data item is sent with a timestamp, which specifies the latest timestamp of that item in the data server. If the data item is present in MUs cache with an

old timestamp, it is updated with this new timestamp that came along in the IR. This algorithm was quite effective in providing the data consistency between the data server and MU's cache. But the only drawback was that periodic transfers of IRs produced heavy message traffic between the servers and MUs. Also, every MU would get the IRs for data items that they don't even require.

3.4 Asynchronous Stateful (AS) **[3]**

This paper proposed an algorithm that used asynchronous call-backs for maintaining cache consistency. It uses call-back model and associate a Home Location Cache (HLC), with each Mobile unit (MU) to deal with the problem of disconnection. Each MSS maintains a distinct HLC for each MU in its cell and this home location cache is used to store additional information specific to each MU in the cell. HLC can be used to cache data even after prolonged period of disconnection of its MU from the network. Thus, no data is lost when the MU is sleeping or entering into a new cell. Whenever an item is update or invalidated at the data server, information is sent to MSS which stores the values in the HLC of the MU which is disconnected. When the MU wakes up, the data item can be retrieved from its HLC. Extra overhead on MSS to maintain HLC.

3.5 Scalable Asynchronous Cache Consistency Scheme (SACCS) **[4]**

This paper proposes an algorithm to maintain consistency between the data server and the MUs by using a flag bit for every data item. At any time, an MU can ask for a data item from the MSS or check whether data item *x* in its cache is consistent with the data item in the data server. The MSS can answer the query or confirm the validity of *x* by comparing the timestamp of *x* in MU's cache and the timestamp provided by the data server. Whenever a query is issued for *x* by any MU, the flag bit for *x* is set to 1 in the MSS. If *x* has been invalidated by the data server, the IR reports are sent by the MSS to all MUs if flag bit for *x* is set to one. Thus, only the information about data items queried by any of the MUs is sent by the MSS. MU sets all its data items into uncertain state every time it wakes up or enters into a new cell and asks the MSS for validity of data items present in its cache. If the items have been invalidated, MSS sends the IRs to MUs, otherwise sends the valid data with new timestamp provided by the data server. Thus, no data is lost.

3.6 Bit-Sequences **[5]**

This paper was proposed for the database servers to optimize the size of the IRs which were sent periodically to all the mobile clients. In this algorithm, all cache entries in mobile cache are deleted only when half or more of data entries in the cache have been invalidated. The server sends a set of bit-sequences and each bit in the bit-sequence represents the data item. The position of the bit decides the indexes of the numbered data items. For example, the nth bit in size N of sequence represents the data item d_n. To indicate the update status of the data item, each data item is associated with a timestamp. A bit ―1 in sequence means that item represented by the bit has been updated since the time specified by the timestamp. A bit ―0means that data item has not been updated since the specified time. All the mobile devices in Infrastructure based networks have direct communication with the Access points. So, the cache consistency in infrastructure based networks is easier. Server sends invalidation reports directly to the mobile devices. It shows comparison between different cache invalidation schemes in Infrastructure networks.

Table 2. Comparison Charts for consistency in mobile cache model

	Message Traffic	Network Utilization	Load on Server/MSS	Good Performance during disconnections	Duplication of data in the client cache	Scalability of mobile clients	Latency	Serves the purpose of mobility
IR Push based	Yes	Yes	Yes	No	Yes	Yes	No	No
Periodical Push based	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Bit Sequences	Yes	No	No	Yes	No	Yes	No	Yes
Timestamp (TS)	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Asynchronos Stateful (AS)	Yes	Yes	Yes	Yes	No	No	No	Yes
SACCS	Yes	Yes	No	Yes	No	Yes	No	yes

4. Conclusion

Mobile computing has proven a fertile area of work for researchers in the areas of client cache management and database management at servers. The inherent limitations of mobile computing systems present a challenge to the fixed problems like consistency, concurrency and currency of data to client systems. The amount of research in this area in the last few years has been staggering. However, some problems remain open for research. There is a need for better protocols in the area of data sharing, message delays, duplication of data at client caches and transaction management,. Better interfaces, clever algorithms that exploit locality to shape the answers to queries.

References

[1] Jochenschiller, ―Wireless communication‖, Addison-Wesly, 2003.
[2] Daniel Barbara, Imielifiski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments" , Proc. OfACM SIGMOD, May 1995
[3] Gupta S.K.S, San Jose, Srimani P.K, "A strategy to manage cache consistency in a disconnected distributed environment", Parallel and Distributed Systems, IEEE Transactions Vol 12, 2001."
[4] Wang Z, Arlington TX, Das S.K, HaoChe, Kumar M, "A Scalable Asynchronous Cache Consistency Scheme (SACCS) for mobile environments", Parallel and Distributed Systems, IEEE Transactions, 2004.
[5] Jin Jing, Ahmed Elmagarmid, Abdelsalam (Sumi) Helal, Rafael Alonso, "Bit-Sequences: An adaptive cache invalidation method in mobile client / server environments", Springer, Mobile Networks and Applications, Vol 2, 2006.
[6] Yu Du and S.Gupta, ―COOP-A Cooperative Caching service in MANETs‖, Proceedings of the IEEE ICAS/ICNS (2005), pp.58-63, 2006.
[7] Seetha A. J, Kannan A, "Maintaining Data Consistency in Mobile Database Broadcasts", Annual International conference and Exhibition on GIS GPS and Remote sensing; 271-278, 2002.
[8] Kam-Yiu Lam, Edward Chan, Hei-Wing Leung and Mei-Wai Au ―Broadcasting Consistent Data to Mobile Clients with Local Cache‖.
[9] Jiannong Cao, Yang Zhang and GuohongCao,LiXie, ―Data Consistency for Cooperative Caching in Mobile Environments‖
[10] N.Chand, R.C.Joshi and M.Misra.―Cooperative caching strategy in mobile Ad hoc networks based on clusters,‖ Wireless Personal Communications,43(1):41– 63, 2007.
[11] KassemFawaz and Hassan Artail, ―DCIM: Distributed Cache Invalidation Method for maintaining Cache Consistency in Wireless Mobile Networks‖, IEEE Trans. Mobile Computing, Vol. 12, No. 4, April 2013.
[12] H. Artail, H. Safa, K. Mershad, Z. Abou-Atme, and N. Sulieman, ―COACS: A Cooperative and Adaptive Caching System for MANETS,‖ IEEE Trans. Mobile Computing, vol. 7, no. 8, pp. 961- 977, Aug. 2008.
[13] T. Andrel and A. Yasinsac, ―On Credibility of MANET Simulations,‖ IEEE Computer, vol. 39, no. 7, pp. 48-54, July 2006.
[14] J. Cao, Y. Zhang, G. Cao, X. Li, ―DataConsistency for Cooperative Cache In Mobile Environ, Computer, vol. 40, no. 4, pp. 60-66, 2007.
[15] Mershad K, Artail H,―SSUM: Smart Server Update Mechanism for Maintaining Cache Consistency in Mobile Environments‖, IEEE Transactions on mobile computing, 2010.