Minimizing Network Contention with Updatable Cache for Consistency in Server Based Mobile Ad Hoc Networks

Sasikala.K, Reka.R, Suresh.T

***Abstract*— In a Mobile Ad hoc Network (MANET), data caching is essential as it reduces contention in the network increases the probability of nodes getting desired data, and improve system performance The major issue that faces cache management is the maintenance of data consistency between the client cache and the server. A cache consistency scheme for caching in MANET is implemented which enable the server to be aware of the cache distribution in the network, making the data items consistent with their version at the server. In cache replication the data which have more requests are distributed to more than one cache node in the network. While replicating , it is to consider whether the replicated cache cannot come under the same QD which stores the information about the present caching node for that data in order to avoid the network congestion in that QD. This approach of cache consistency improves the performance of the query response rate and average response time of node request is estimated. It is simulation to measure parameters, like the hit ratio and average data response time.**

***Index Terms*— Cache Replication, Cache Consistency, Smart server, Network Congestion**

I. INTRODUCTION

A Mobile Ad hoc Network (MANET) is a collection of wireless mobile nodes which forming a temporary network without any base station or any other preexisting network infrastructure. The Links between nodes may constantly change as nodes move around, enter, or leave the network. This can make storing and retrieving cached data particularly difficult and unreliable .the use of mobile devices adds even more complexity due to their relatively limited computing resources and limited battery life. However, frequent disconnections and mobility of the clients make cache consistency a challenging problem. Routing protocols are responsible for finding an efficient path between any two nodes in the network that wish to communicate, and for routing data messages along this path. The path must be chosen so that network throughput is maximized and message delays and other undesirable events are minimized. Data Caching is essential feature in MANET. It reduces contention in the network and increases the probability of nodes getting desired data, and improve system performance. The major issue that faces cache management is the maintenance of data consistency between the client cache and the server .In a MANET, all messages sent between server and the cache are subjected to network delays, thus impeding consistency by download delays.

All consistency algorithms are developed with the same goal in mind: to increase the Probability of serving data items from the cache that is identical to those on the server. A large number of such algorithms have been proposed in the system, and they fall into three groups: server invalidation, client polling, Time To Live (TTL). In server invalidation, the server sends a report upon each

update to the client. In client polling, validation request is sent to the server by client which may create network traffic and server overload.

In TTL Based Algorithm server assigned TTL value to each data item in the cache however, TTL-Based algorithms are popular due to simplicity, good performance and flexibility it is a weak consistency model whereas server invalidation is strong consistency approach. With strong consistency algorithms, users are served strictly fresh data items, while with weak algorithms, there is a possibility that users may get inconsistent copies of the data. The proposed work describes a server-based scheme implementation on the top of the COACS caching architecture. In COCAS, elected query directory (QD) Nodes cache submitted queries and use them as indexes to data stored in the nodes that initially requested them (CN nodes). Since COACS did not implement a consistency strategy, the system described in this proposed paper fills that void and adds several improvements: 1) enabling the server to be aware of cache distribution in the MANET, 2)making the cached data items consistent with their version at the server, 3) adapting the cache update process to the data update rate at the server relative to the request rate by the clients, and 4) minimizing the cache hit ratio by cache replication .With these changes, the overall design provides a complete caching system in which the server sends to the client's selective updates that adapt to their needs and reduces the average query response time.

The proposed system implements smart server update mechanism (SSUM). The SSUM is a server-based approach that avoids many issues associated with the push based cache consistency approaches. Push based mechanism are mostly server-based, where the server informs the caches about updates. This push based mechanism is a strong consistency model where cached items are identical to those on the server. Server sends a report upon each update to the client. In SSUM only one caching node is allowed to cache for a particular data. In this method if the request rate for the particular data is high then there is more number of hits in a particular QD&CN

Manuscript received December 22, 2014.
Sasikala.K, Associate Professor, Dept of IT, V.M.K.V. Engineering College, Salem, India
Reka.R, Assistant Professor, Dept of CSE, V.M.K.V. Engineering College, Salem, India
Suresh.T, M.Phil Scholar, Vinayaka Missions University, Salem, India

for that data. It also increases the traffic in the network. In order to avoid this , the concept of cache replication is implemented.

In cache replication the data which have more requests are distributed to more than one cache node in the network while distributing the data to the other cache we have to consider whether the replicated cache cannot come under the same QD which stores the information about the present caching node for that data in order to avoid the network congestion in that QD.

The rest of the paper is organized as follows: Section II presents the related work on consistency approaches and caching system. Section III describes the SSUM mechanism Section IV we propose replication technique for reducing cache hit ratio. We devote Section V the performance evaluation of our scheme .Finally, we conclude the paper in Section VI.

II. RELATED STUDIES

A. *Cache invalidation*

Caching frequently accessed data on the client side is an effective technique for improving performance in a mobile environment .When cache techniques are used, data consistency issues must be addressed to ensure that clients see only valid states of the data, or at least do not unknowingly access data that is stale according to the rules of the consistency model.

Depending on whether or not the server maintains the state of the client's cache, two invalidation strategies are used: the stateful server approach and the stateless server approach. In the stateful server approach, the server maintains the information about which data are cached by which client. In the stateless server approach, the server is not aware of the state of the client's cache .The client need to query the server to verify the validity of their caches before each use. In this approach, the server periodically broadcasts an invalidation report (IR) in which the updated data items are indicated. Rather than querying the server directly, the clients can listen to these IRs over the wireless channel. One of the most important considerations in the design of cache invalidation scheme is to minimize the required bandwidth for broadcasting the invalidation reports as well as the size of the invalidation report.Various efficient methods have been proposed for preparing the cache invalidation reports such as the bit sequence and time-stamp methods.

B. *Cache Consistency Technique*

A Cooperative caching is an important technique to support efficient data dissemination and sharing in Mobile Ad hoc Networks (MANETs). Stateful algorithms can significantly reduce the consistency maintenance cost by maintaining status of the cached data and selectively propagating the data updates. Stateful algorithms are more effective in MANETs, mainly due to the bandwidth-constrained, unstable and multi-hop wireless communication. In this paper, it proposes a stateful cache consistency maintenance algorithm called Greedy Walk-based Selective Push (GWSP).

In GWSP, the data source node maintains the Time-to-Refresh value and the cache query rate associated with each cache copy. Thus, the data source node propagates the source data update only to caching nodes which are in great need of the update. After recipients of the source data update have been decided, GWSP employs a greedy but efficient strategy to propagate the update among the selected caching nodes.

In GWSP, the data source node maintains the Time to refresh value and the cache query rate associated with each cache copy. Based on the cache status, the data source node can hence make online decisions on which cache copies are in need of the data updates upon each update. When recipients of the data update have been decided, GWSP employs the greedy walk mechanism to propagate the data update among the selected caching nodes. Cooperation among the data source node and caching nodes amortizes the cost for data update propagation

C. *Dynamic Replica Allocation Scheme*

The CLEAR (Context and Location-based Efficient Allocation of Replicas), a dynamic replica allocation scheme for improving data availability in mobile ad-hoc peer-to-peer (M-P2P) networks. CLEAR exploits user mobility patterns and deploys super-peer architecture to manage replica allocation efficiently. CLEAR avoids broadcast storm during replica allocation and eliminates the need for broadcast-based querying. CLEAR considers different levels of replica consistency and load as replica allocation criteria. In performance study indicates CLEAR's effectiveness in improving data availability in M-P2P networks.

D. *Cooperative Cache Consistency*

A Cooperative caching is an important technique to support efficient data dissemination and sharing in Mobile Ad hoc Networks (MANETs). Stateful algorithms can significantly reduce the consistency maintenance cost by maintaining status of the cached data and selectively propagating the data updates. Stateful algorithms are more effective in MANETs, mainly due to the bandwidth-constrained, unstable and multi-hop wireless communication. In this paper, it proposes a stateful cache consistency maintenance algorithm called Greedy Walk-based Selective Push (GWSP).

In GWSP, the data source node maintains the Time-to-Refresh value and the cache query rate associated with each cache copy. Thus, the data source node propagates the source data update only to caching nodes which are in great need of the update. After recipients of the source data update have been decided, GWSP employs a greedy but efficient strategy to propagate the update among the selected caching nodes.

In GWSP, the data source node maintains the Time to refresh value and the cache query rate associated with each cache copy. Based on the cache status, the data source node can hence make online decisions on which cache copies are in need of the data updates upon each update. When recipients of the data update have been decided, GWSP employs the greedy walk mechanism to propagate the data update among the selected caching nodes. Cooperation among the data source node and caching nodes amortizes the cost for data update propagation

III. PROPOSED APPROACHES

A. . Problem Statement

The Cooperative Caching architecture for mobile environments minimizes delay and maximizes the likelihood of finding data that is cached in the ad hoc network, but there is no consistency between the cached data item and the server data. To maintain the consistency, smart server is implemented on the top of the COACS system ,in which the server maintains the information about the cache node and the data item .The data item in the CN is consistent with the version in the server by sending an invalidation report to CN by using Push algorithm. The server autonomously sends data updates to the CNs, meaning that it has to keep track of which CNs cache which data item

In SSUM the RN sends request to the nearby QD, The QD forwards the request to the CN containing desired data item, if the request is not in any of the QD it is forwarded to the server and the server reply to the CN. If there is more number of request in the Particular QD & CN , then there is more number of hit in the QD & CN . Average response time increases and lead to more traffic . The maximum Hit in the CN & QD leads to the overhead in that particular CN & QD . SSUM reduces wireless traffic by tuning the cache update rate to the request rate for the cached data.

In this paper, cache replication mechanism is used for minimizing the hit occur in the CN & QD. Frequently accessed data item is replicated to two or more nodes so that the average response time decreases and the minimize the cache hit ratio. The replication scheme is implemented in the SSUM system.

B. Packet forwarding algorithm

Packet forwarding algorithm is used for traversing the QD system. The idea behind PFA is to use routing table information for visiting nodes in the order of shortest distance. According to PFA , the client uses the information in the routing tables to send its request to the nearest QD. If an QD does not have the requested data ,it uses PFA and forwards the request to the nearest unvisited QD . Nodes request database data, which may be cached in any of the caching nodes (CNs). The QD cache previously submitted request (queries), and for each such query, an QD maintains a reference to the result that resides on a CN.

In fig 2. The first scenario, the client submits its request to the nearest QD (QD3), which does not have a matching query. the request is then forwarded in accordance with PFA through QD1 and QD4 before it arrives to QD2, where a match is found. Using the reference that is stored along with the cached query, the request of the client is forwarded to the CN that stores the result. This CN sends the result to the client whose address is found in the forwarded packet. In the second scenario, no match is found in the QDs, and so, the last visited QD (QD5) forwards the request to the data server via the access point. The server retrieves the result and sends it directly to the client, which, in turn, asks QD3 to cache the query. it is noted that he nod at which the client requested the data item that was retrieved from the outside dataserver becomes a CN for this particular item.

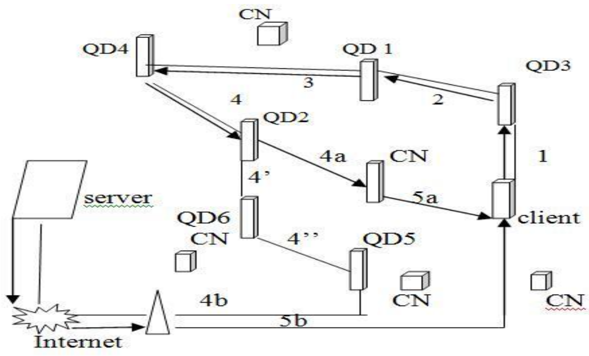

Fig 1. Two scenarios for request forwarding: scenario 1corresponding to a hit and includes steps 1-4, 4a and 5a. Scenario 2describes a miss and includes steps 1-4,4',4'',4b,and 5b.

C. Handling Node Disconnection

Mobile nodes are dynamic in nature, when a CN leaves the network, the QD, which first tries to forward it a request and fails, will set the addresses of all queries whose items are cached by this unreachable CN in its cache to -1,and informs the server about the mobility of the cache node.

The server, in turn, changes the address of that CN in its cache to _1 and stops sending updates for these items. If the cache node returns to the network after disconnecting ,it informs the QD about its entry. By this approach the network traffic is reduced. The packet forwarding algorithm handles the node disconnection, it forwards the packets to the nearby node by using AODV routing protocol which avoids network traffic.

IV. PROPOSED CACHE REPLICATION SCHEME

A. Network Model

The network consists of mobile hosts that form a group. Mobile host takes two possible roles such as QD and CN . The network connectivity is maintained using a QD. Routing protocols are responsible for finding an efficient path between any two nodes in the network that wish to communicate, and for routing data messages along this path. RN send a request to nearby QD, and it finds the data in the CN. Reply for the desired data is sent from the corresponding CN.

B. System Model

The system model is assumed to be an ad hoc network where MN(mobile node) access data item held as originals by other MNs. A MN that holds the original value of a data item is called data source/server. A data request initiated by a MN is called RN(request node). The frequently accessed data item is stored in a MN called CN(cache node). The Requested Query and the address of the cache node is stored in a MN called QD (query directory).

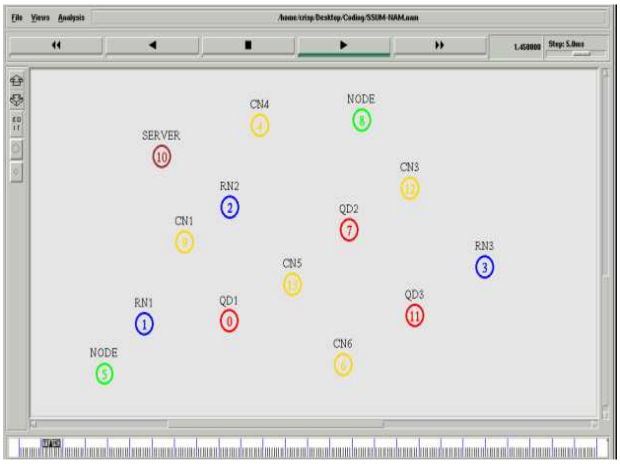

Fig 2. Roles of MN

Nodes takes different roles such as QD,CN and RN for request handling and Network Monitoring in the Mobile ad hoc network

C. *Request Handling*

The RN sends request to the nearby QD. The QD maintains a table consists of id of the data item and the address of the cache node. When the RN sends the request to the QD, it first checks its table ,if the match is found in QD, it forwards the request to the related CN and the particular CN replies directly to the RN. If any miss occurs in the QD, it forwards the request to the nearby QD ,when a miss occurs in all of the QD the request is forwarded to the server and the server replies to the RN, and act as a cache node for the particular data.

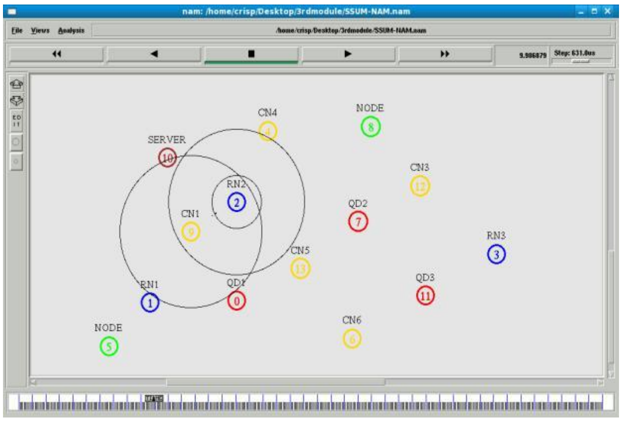

Fig 3. Request Handling

D. *Network Monitoring*

As mobile nodes are dynamic in nature ,they can move freely from one network to the other network. In Smart server update mechanism when a cache node moves to other location due to mobility it informs the QD. The QD marks the address of the cache node as -1. If again a new node enters into the network it sends the request to the nearby QD, and the address of the Cache node is stored in the nearby QD. The server maintains the information about all the cache node in the network. server autonomously sends updates to the all cache node in the network.

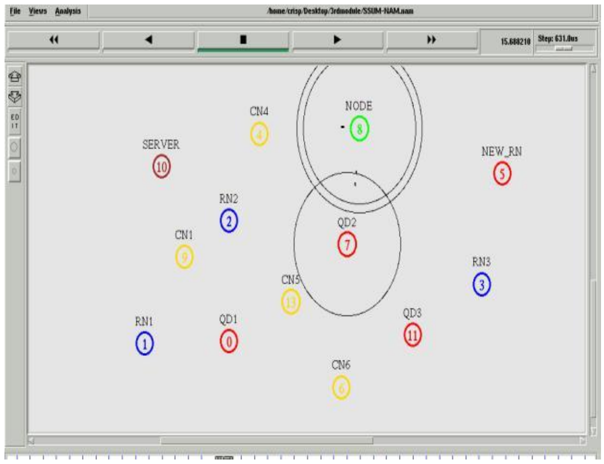

Fig 4. Mobility of Nodes

E. *Data Replication*

In SSUM, when there is more request in a particular QD and CN ,then there will be more hit in that particular QD & CN. The Average response time also decreases and network traffic arises. In order to minimize the average response time Data is replicated to more than one node in the network. The Replicated Cache node should not come under one QD. The Data Replication minimizes the network traffic and minimizes the maximum hit occur in a particular QD & CN.

The replication cache cannot come under the same QD which stores the information about the present caching node for that data in order to avoid the network congestion in that QD.

V. SIMULATION SETUP

In this section, we are going to reduce the response delay by replicating the data to one or more nodes in the network. we use the ns2 software to implement the replication scheme in the SSUM System .

A. *Simulation Parameter*

A single database server is connected to the wireless network through a fixed access point, while the mobile nodes are randomly distributed. The client cache size was fixed to 300 Kb, meaning that a CN can cache between 20 and 300 items ,while the QD cache size about 700 items. The SSUM system was implemented as a new C++ agent in ns2 that get attached to the node class in the tcl code at simulation runtime PFA is used for traversing the QD system

Table-1: Simulation Parameters

Simulation Parameter	**Default value**
Network size	750x750m^2
Node transmission range	100m
Number of Nodes	80
Node Speed	2.5(m/s)
Node request period	22sec
Size of data item	1-20kb
Total number of data items	20,000

B. Results

Fig. 5 clearly demonstrates the decrease in hit ratio by replicating the data item to one or more nodes in the network. By data replication the network traffic can be reduced significantly. The delay in the response time reduces and minimizes the maximum hit occur in the CN. The Hit rate is calculated by simply count the number of queries and the number that are cache hits. The hit rate is defined to be the number of cache hits divided by the number of cache queries in the simulation run. The cache hit ratio increases while the cache size increases. when the data is replicated for more than one node, the query delay decreases.

The smart server update Mechanism updates the data in the cache node with the current version in the server so the reply of stale document decreases in the MANET

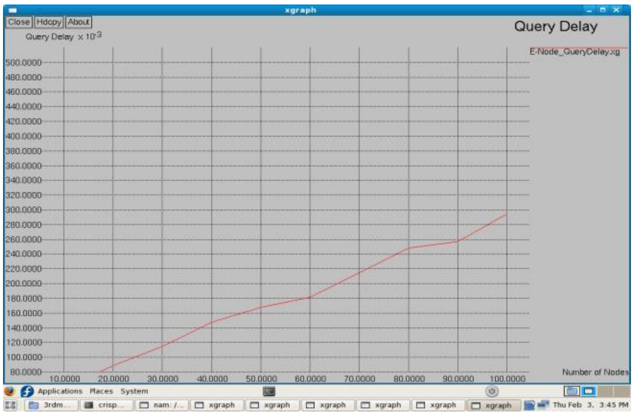

Fig: 5. Query Delay

Query delay is the important issue faced by Smart server update mechanism . By data replication the query delay can be reduced.

VI. CONCLUSION

The proposed scheme can minimize the query delay in the Smart server update mechanism and reduces the network traffic. In addition, we have adopted the Data replication model for frequently accessing the data, thus improving the performance of the system as compared to the SSUM, and minimize the contention in the network. The extensive results have demonstrated that, in comparison with the existing methods, our proposed Replication scheme is more effective and efficient in accessing the data, reducing response time, and improving the Performance of the system.

REFERENCES

[1]H.Artail,H.Safa,K.Mershad, Z.AbouAtme, N.Sulieman, "COACS: A Cooperative and adaptive Caching System forMANETS,"IEEETrans.MobileComputing,vol.7,no.8,pp.961-977,Aug.2008

[2] H.Artail and K.Mershad,"MDPF: Mnimum Distance Packet Forwarding for Search Applications in Mobile AdHocNetwork,"IEEETrans.MobileComputing,vol.8,no.10,pp.1412-1426,oct.2009

[3] K. G.Cao,"A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments, "IEEE Trans. Knowledge and Data Eng.,vol 15, no.5, pp.1251-1265, sept.2003

[4] P.Cao and C.Liu,"Maintaining Strong Cache ConsistencyintheWorldWideWeb,"IEEETrans.Computers,vol.47, no.4,pp.445-457,apr.1998

[5] K.Shanmugavadivu and Dr M.Madheswaran,"caching Technique for Improving data retrival performance in mobile adhoc networks .

[6] J.Yuen,E.Chan,K.Lain,andH.Lueng,"CacheInvalidation scheme for Mobile computing with real-time Data,vol.29,no.4,pp.34- 39,dec.2000

[7] H.Maalouf and M.Gurcan,"Minimisation of the Update Response Time in Distributed DatabaseSystem,2002

[8] H.Artail,and,K.Mershad, "SSUM:Smart Server Update Mechanism for Maintaining Cache Consistency in mobile environments,"IEEE TMC,v.9,n.6,pp.778-795,2010

[9] J.Jung,A.W.Berger,andH.Balakrishnan,Modelling TTL- Based Internet Caches,"Proc.IEEE INFOCOM,Mar.2003