ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL R-SIP CACHE CONSISTENCY MECHANISM FOR MOBILE WIRELESS NETWORKS

[1]LILLY SHEEBA S, [2]YOGESH P

[1]Assistant Professor, Department of Information Technology, Anna University, Jerusalem College of Engineering, Chennai,TamilNadu, India

[2]Associate Professor, Department of Information Science and Technology, Anna University, Chennai,TamilNadu, India

E-mail: slilly_sheeba@yahoo.com, yogesh@annauniv.edu

ABSTRACT

In Mobile Adhoc NETwork (MANETs), caching of data items among mobile nodes have become an inevitable solution for fast information retrieval. However most of the research works focus on ensuring routing, security and research on maintaining cache consistency within MANETS. Here we focus on providing cache consistency in a mobile environment so as to increase the probability of retrieving up to date data from nearby caching nodes instead of from the server. In order, to ensure cooperative cache consistency we are going for updating the cache nodes using Registration based Server Initiated Push(R-SIP) mechanism where server maintains a registration table to maintain the details of all registered clients and updates those registered, by pushing the updated data, as and when the server gets refreshed. Here only registered clients that have registered are updated frequently, there by not compelling unregistered clients to receive the recently updated data. This cache consistency mechanism for cooperative caching ensures reduced bandwidth utilization, less query latency, besides decreasing network traffic and the communication overhead at the data server.

Keywords: *Cache Consistency, Cache Invalidation, Mobile Ad Hoc Networks, Registration Table.*

1. INTRODUCTION

A computer network is one where the computer nodes that form the network are able to exchange data among them. A computer network can be established either using cable wires or wirelessly.

In a wired network, the computer nodes are connected to the network physically with the using wires. A wireless network as the name suggests is used to establish a computer network wirelessly. That is, it eliminates the need for using the cable wires by making use of the electromagnetic radio waves. Wireless network is therefore largely employed in areas where wired connection is highly impractical. Some of the most common wireless networks include Wireless Sensor Networks, MANETs, Wireless Fidelity (Wi-Fi), Cellular Data Service and Mobile satellite Communication, etc.

Our research takes into account a MANET environment. The term MANET stands for Mobile Ad hoc NETworks. Here the term mobile represents mobile nodes capable of movement. In a MANET, nodes are free to move in to or out of the network (i.e.) self-configuring nodes. Hence they lead to a dynamic network. MANETs have wide spread importance in areas where a fixed infrastructure based network is not possible. In short, it eliminates the need neither for installing base stations nor on wiring based network. MANET proves to be an apt solution to educational applications through the deployment of virtual classrooms and ad hoc communications during lectures, emergency services such as search and rescue operation, disaster recovery, sensor networks, coverage extensions, military communication and operation, automated battle fields, outdoor internet access etc. Nodes in MANET are unaware of the capabilities or services offered by other mobile nodes. Hence proper data accessing and retrieval methods are required to enhance data availability. Cooperative caching

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

allows sharing cached data contents among other mobile nodes of common interest. But suitable cache sharing schemes must be devised, since MANET suffers from constraints in bandwidth utilization, network traffic and limited battery power. Moreover the cached data must be identical to the data associated with the server. This is achieved by implementing cache consistency schemes within nodes.

In this paper we implement R-SIP, a push based server initiated cache consistency mechanism which ensures cache consistency in a infrastructure less and highly dynamic environment. Here the server is entitled to push recent updates only to the registered clients.

2. RELATED WORK

In CO-operative and Adaptive Caching Systems (COACS) [1], nodes that request data items are termed Requesting Node (RN) while a Query Directory (QD) is a node that maintains all details of Caching Nodes (CNs). H. Artail et al. have designed a system that uses multiple nodes to store and retrieve data. This not only eliminates the need of a service manager but also pushes the need for distributing the functions among multiple nodes. This in turn leads to reliable caching. Here a node can either be a CN or QD. A requesting node becomes a CN when it receives and caches the data item. The RN sends request to the QD, which in turn directs it to the CN which has the requested data item. The CN gives back the data to the RN. When the data reaches RN it becomes a CN. Supposing the data item is not found in local QD, the request is forwarded to other QDs using the Minimum Distance Packet Forwarding (MDPF) algorithm and once again data item if found sent and cached. Thereby becoming a CN, otherwise the request is forwarded to server which will always return the requested data item . The hit ratio of the system is increased because of using QDs but the increased hit ratio comes at the cost of higher utilization of bandwidth.

In [2] Yin. et al., have proposed different protocols to enhance the probability of finding data in the cache and to improve its presence within cache were taken into consideration. Three different schemes for caching were proposed here which includes Cache Path, Cache Data and Hybrid cache schemes. In Cache Path the location of the data is stored in all intermediate nodes. The nearest node does not need to store the path information of the data and hence the query latency can be avoided when compared to Cache Data where the entire data is cached. Here only the data is stored instead of the path information of the data, so the requesting nodes are served with data directly from the intermediate node rather than getting the data from the far away data server. In Hybrid Cache both the Cache Path and Cache Data approach can be alternatively employed based on some criteria.

The Smart Server Update Mechanism (SSUM) discussed in [4] is built on COACS architecture. This approach differs from the ordinary push based system by employing a server that maintains a directory to keep track of what data items are cached, when the node enters and leaves the network. Server can still say which CN has cached what data item. This method aims at reducing the network traffic by keeping a watch on update rate to request rate for cached data. Here K. Mershad and H. Artail, suggests solutions for handling node disconnections in MANET. It suggests ways to deal with disconnected nodes. When either the Query Directory (QD) or CN reenters or leaves the network, the data cached must be properly dealt with. They also estimated the cost incurred by this scheme based on average response time and bandwidth utilization.

In paper [5] Wenzhong Li et al. proposed many cache invalidation strategies. The proposed strategies include Pull on Demand (POD), Push-based Amnesic Terminal (PAT) and Modified Amnesic Terminal (MAT). POD is a polling method in which, when a query is initiated, a node approaches the Mobile Support Station (MSS) to verify the validity of the requested cached item before it responds to any query. If the data is valid and it sends a confirm message. In MAT, MSS broadcasts an Invalidation Report (IR) every second. Only the Mobile Terminals (MTs) which are found within its range receive the IR directly. By this it is ensured that IR packets are not flooded over the network. In PAT the node and the server maintains an IR queue. The server has an IR queue which contains the list of the last items updated.

In paper [6] proposed by K. Fawez and H. Artail, a cache consistency mechanism for a mobile ad hoc environment was proposed. They have implemented three methods including adaptive Time to Live (TTL), piggy backing and pre fetching to attain cache consistency. Here TTL values are involved. In case of piggy backing , if any node in a MANET needs to send requests to the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

server, if does so by riding at the back of packets passing through it. Pre-fetching is another technique under highlight. Here the client node sends the data request to the server with pre fetch bit set on .The server immediately responds as whether the data is valid or invalid .If valid it means the data in the cache and the data server is same and if invalid it implies that the data in the cache and server does not match. If the pre fetch bit is set on, then the server besides sending the invalid or valid reply responds again with the actual data.

J. Cao and G. Cao in [3], have proposed schemes for cooperative caching. Caching in general is used to share and access the data that are being cached by nodes and thereby increasing data availability within a network. Moreover, the data that is accessed must be an updated one, identical to that present in the server. This is achieved by going for appropriate consistency schemes. Here they propose a scheme that makes use of both push based and pull based techniques. Each node assumes any one of the three state transitions. The three states include Relay peer, Relay peer candidate and Cache node. Cache node is the node which contains the required data. The cache node becomes a relay peer candidate when the cache node satisfies some criteria like update rate by request rate, etc. Then the relay peer candidate will become a relay peer if the relay peer candidate can gets the invalidation message send form the source node. Here a relay peer candidate tends to function as an interface between the cache node and the relay peer.

3. PROPOSED SYSTEM

In general caching of data in MANET adopts either push or pull based cache consistency schemes. In push based cache consistency schemes, server initiates the process of pushing data to the clients where as in a pull based scheme the client initiates the request query for attaining desired data item from the server. In a push based scheme the server compels all the caching clients to receive the up to date data, even against the wish of the client. It is quite unlikely that all clients are expecting recent updates on data at all times. This might lead to unnecessary band width utilization as the requestor node is not interested in receiving regular frequent updates of data.

In order to overcome this, we go in for a Registration based Server Initiated Push(R-SIP) mechanism which aims at maintaining consistency between caching nodes in MANET and the server by implementing a registration based data push algorithm.

3.1 Cooperative Caching

Instead of every time requesting the server for a data, the data if already cached by any pre requested nodes can be shared to another requesting node. This leads to all nodes in a MANET cooperating for caching. Cooperative caching ensures sharing of the cached data among cooperating clients. Initially a client may initiate a request for a data item to the server. The server immediately responds by sending the data of interest to the requesting client via intermediate nodes. In case of cooperative caching, the intermediate nodes can cache the data item and broadcast this cache information to all its nearest nodes. These neighboring nodes can then cache information about this cache event. At a later stage, if these neighboring nodes receive requests for the same data, it easily diverts the request to the caching node that has the data.

As a result, all intermediate nodes are capable of delivering data to any future requestors of this data without the request approaching the server.

3.2 R-SIP

In R-SIP, the server maintains a registration table. Here requestors that require frequent updates of any particular data of interest should register with the server, for otherwise, the requestors will not be compelled to receive recent updates of a data. Any requestor that wants to register sends a request to the server by setting the registration bit on. If the registration bit is set on, the server makes an entry in the registration table and responds by sending the requested data along with a time stamp equivalent to the refresh rate of the server. After registration, it becomes the prime job of the server to update the registered clients with frequent updates.

Here we assume that the refresh rate of the data in a server is quite regular and the server is aware of the next refresh time. This time stamp indicates the time at which the server is going to update or refresh the data in future. After this time the data present with the requestor node might become obsolete or outdated if not updated. Hence the data can be removed from the cache or should be updated to provide for cache consistency.

Journal of Theoretical and Applied Information Technology
20th July 2014. Vol. 65 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

As our protocol is implemented in a co-operative caching mobile environment, the intermediate nodes are all aware of the data and the time stamp. This information is again broadcasted to the nearest nodes and updating of the cached information takes place. Hence any future requests can easily attain the data from the intermediate nodes rather than from the server before the speculated time expires.

Requestors sending requests with registration bit set off are not compelled to receive the updates, as pushing of all updates of the desired data is initiated by server only for the registered clients.

3.3 Registration Table

Each entry in the registration table has the following attributes:

a) data item
b) first time of update
c) next refresh time
d) list of registered requestors

Keeping track of each entry in the table, the server correctly updates data in all the registered nodes based on the next refresh time.

3.4 Automatic Replacement

R-SIP employs automatic replacement algorithm to delete old outdated data from the cache of the caching nodes, to ensure that any new requesting nodes get only recent updates of data identical to that present with the server. This in turn ensures cache consistency.

As mentioned before, timestamp is included to denote the validity of the data item. To better utilize cache memory which happens to be small and fast, the cache has to be refreshed as and when the timestamp expires. This refreshment involves removing the expired data item and replacing it with the new data item. Thus the cache memory is effectively used in this system.

4. SYSTEM MODEL

Fig.1 represents a simple MANET. Here the mobile environment includes a server from which data items can be fetched. Here the mobile nodes include Laptops, PDAs, Mobile Phones, etc. In a MANET, nodes are free to enter and exit the network anytime. A data server is a fixed node that can store and manage data related to any specified application.

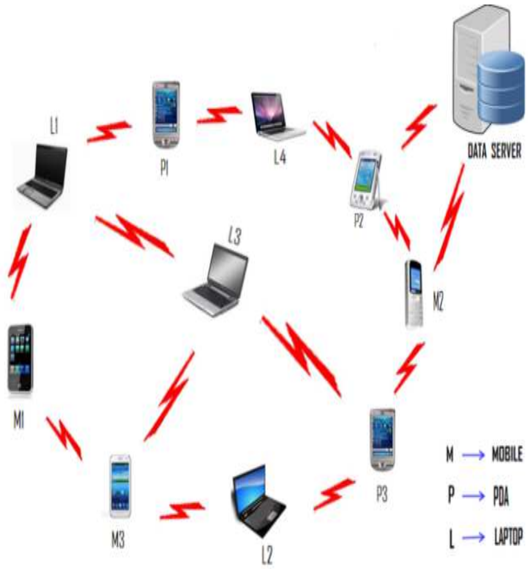

Fig.1 A Simple MANET

Here L1, L2, L3, L4 are laptops, P1, P2, P3 are personal digital assistants and M1, M2, M3 are mobile phones which communicate with each other through wireless technologies like Wi-Fi, Blue tooth, etc. to form a dynamic infrastructure less topology.

Fig. 2 explains the information retrieval procedure adopted by R-SIP. Any request will end either with a cache hit or a cache miss. In case of cache hits the data is acquired and immediately directed to the requestor node. If the desired data is not present in the local cache, or in the intermediate node ultimate caches miss occurs. When an ultimate cache miss is incurred then there is no other way to fetch the data item other than to forward the request to the data server which will return the data.

Journal of Theoretical and Applied Information Technology
20th July 2014. Vol. 65 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

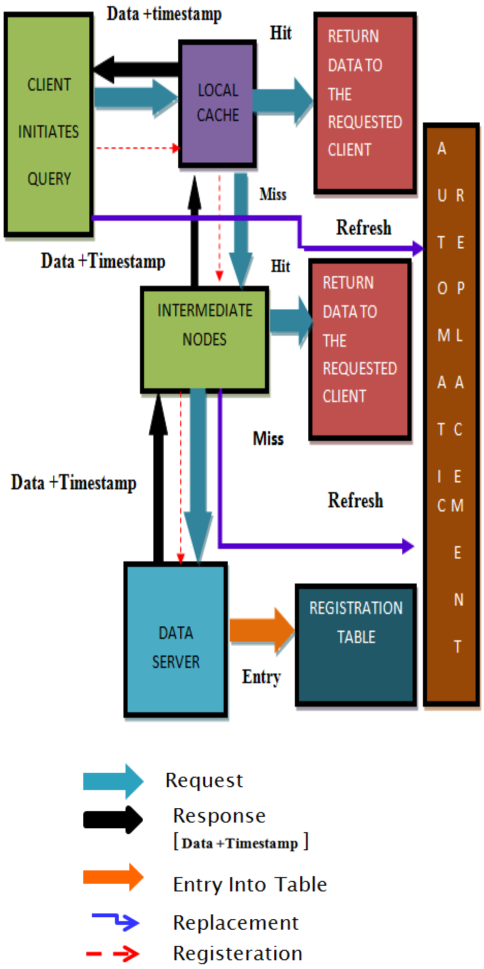

Fig.2 Information Retrieval

5. PERFORMANCE EVALUATION

5.1 Simulation Parameters

Simulation of the R-SIP is done using ns2 simulator. Ns2 is a network simulator that is employed for creating layer level simulation environment for mobile wireless networks.Table.1 specifies the simulation parameters used in the implementation R-SIP. Here the mobile environment is spread across an area of 1000 m * 1000 m. All nodes employ a random way point model. Performance metrics taken for analysis include cache hit ratio, query latency and communication overhead. These metrics assist in evaluating the effectiveness of the proposed scheme when included in a wireless mobile network.

Table1. Simulation Parameters

Simulation Parameter	Default Value
Simulation time	1000 s
Network time	1000*1000 m^2
Bandwidth	3Mb/s
Transmission range	200 m
Number of nodes	50
Speed	2 m/s
Pass time	20 s
Data items in the server	3000
Size of data item	10 Kb
Cache size	200 Kb
Updates/ second	200 updates/s
Delay	30 ms
Request rate	5 requests/minute
Access pattern	Zipf distribution

6.2 Result Analysis

Here in Fig.3, the density of the nodes present within the network are varied compared with cache hit ratio. Cache Hit Ratio is the ratio of the number of query requests to the number of the requests that has turned up with ultimate responses successfully. The number of nodes within the network are varied like 50, 75, 100, 125, 150, 175 and 200 nodes. It can be incurred from the Fig.3, that the cache hit ratio in R-SIP increases considerably when compared to the SSUM approach, because unlike SSUM where the datas are retained only in query directories, here information are scattered among multiple intermediate nodes.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

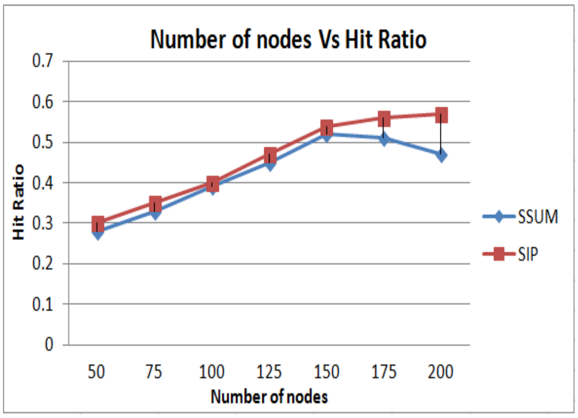

Fig. 3 Number of nodes Vs Hit Ratio

In Fig.4 we compare Query Request Rate against Query Latency. Here query latency is the delay incurred between a request for desired information and that of subsequent response. The default query rate here is 5 requests/minute. It is varied as 5, 10, 15, 20, 25, 30 requests/minute. From Fig.4 it can be incurred that the query latency in R-SIP gets reduced when compared to SSUM where every request goes via query directories.

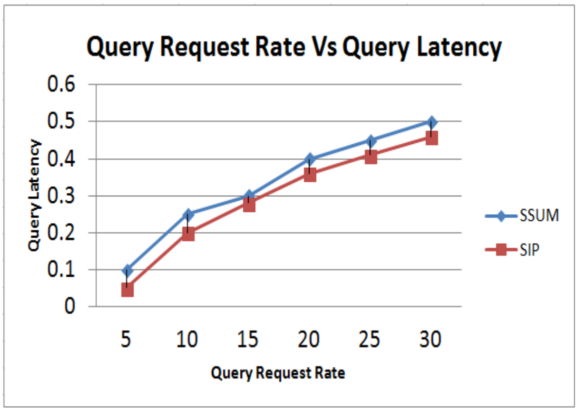

Fig. 4 Query Request Rate Vs Query Latency

Fig. 5, compares Data Update Rate Versus Communication Overhead with respect to SSUM and R-SIP. Here communication overhead deals with message exchanges involved for information retrieval and maintaining high consistency levels. The default rate is the rate at which data items are updated per second. Here the data update rates are kept as 10, 20, 30, 40 and 50 updates per second. From figure 5, we can conclude that when compared to SSUM, R-SIP induces less communication overhead as, no additional messages are exchanged for information retrieval here, owing to the removal of query directories.

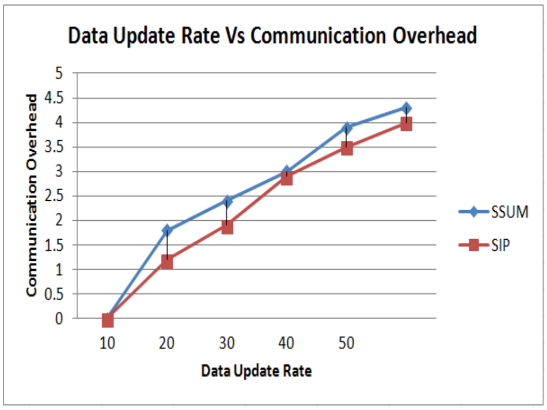

Fig.5 Update Rate Vs Communication Overhead

6. CONCLUSION

Caching of data in MANET has always been an expected solution for enhancing data availability in any network. But proper data consistency mechanism should be adopted to improve the overall performance of the network. Our R-SIP, is an effective mechanism specially designed for maintaining cache consistency in a dynamic mobile environment. This work makes use of a registration based push mechanism. This protocol employs a registration table at the server to keeps track of registered requestor nodes and the time of subsequent data update. Nodes which are interested in receiving subsequent updates are expected to register themselves with the server to receive periodic updates. Moreover unregistered nodes are not compelled to receive frequent updates of the requested data. It also provides for automatic replacement where in the outdated data are regularly out casted. Thus R-SIP ensures reduced bandwidth utilization, query latency besides decreasing network traffic and communication overhead at the data server.

REFERENCES:

[1] 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, Aug. 2008pp.961-977.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad hoc Networks", IEEE Trans. Mobile Computing, vol.5, no.1, 2006, pp.77-89.

[3] J. Cao, Y. Zhang, G. Cao, and X. Li, "Data Consistency for Cooperative Caching in Mobile Environments", Computer, vol.40, no.4, 2007. pp.60-66.

[4] K. Mershad and H. Artail, "SSUM: Smart Server Update Mechanism for Maintaining Cache Consistency in Mobile Environment," IEEE Trans. Mobile Computing, vol.9, no.6, June 2010.pp.778-796.

[5] Wenzhong Li, Edward Chan, Y. Wang and D. Chen, "Cache Invalidation Strategies for Mobile Ad hoc Networks.", International conference on Parallel Processing, , 2007, 0-7695-2933-x/07.

[6] Kassem Fawaz and Hassan Artail, "DCIM: Distributed Method for Maintaining Cache Consistency in Wireless Mobile Networks", IEEE Trans. Mobile Computing, vol.12, no.4, April 2013, pp.680-693.

[7] X. Tang, J. Xu, and W-C. Lee, "Analysis of TTL-Based Consistency In Unstructured Peer-to-Peer Networks," IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 12, Dec. 2008, pp. 1683-1694.

[8] Y. Fang, Z. Haas, B. Liang, and Y.B. Lin, "TTL Prediction Schemes And the Effects of Inter-Update Time Distribution on Wireless Data Access," Wireless Networks, vol. 10, 2004pp. 607-619.

[9] X. Tang, J. Xu, and W.C. Lee, "Analysis of TTL-Based Consistency in Unstructured Peer-to-Peer Networks," IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 12, Dec. 2008 pp. 1683-1694.

[10] Naveen, Chauhan, and L.K. Awasthi, "Prefetching based Cooperative Caching in Mobile Ad Hoc Networks", International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE'2012) March 24-25, 2012 Dubai.

[11] Z. Jiang and L. Kleinrock, "An Adaptive Network Prefetch Scheme," IEEE Journal on Selected Areas in Communications, Vol. 16, No. 3, 1993, pp 1-11.

[12] F. Sailhan and V. Issarny, "Cooperative Caching in Ad Hoc Networks," Proceedings of the International Conference on Mobile Data Management (MDM), 2003, pp. 13-28.

[13] M. Denko and Juan Tian, "Cooperative Caching with Adaptive Pre fetching in Mobile Ad Hoc Networks", IEEE International Conference on wireless and Mobile computing and communication, 2007, pp. 38-44.

[14] Mobile Communications –Jochen H. Schiller Addison-Wesley, 2003 - Computers - 492 pages.

[15]_Ad Hoc wireless networks: architectures and protocol C. Siva Ram Murthy, B. S. Manoj Prentice Hall PTR, 2004 - Computers - 857 pages