

DISTRIBUTED CACHE INVALIDATION METHOD FOR PACKET ROUTING MECHANISM IN WIRELESS MANET

Mr. Kiran Deshmukh

Information Technology, PVPPCOE, Mumbai University, India

kirandesh23@gmail.com, Tel: +97661 77553

Abstract : This paper proposes distributed cache invalidation mechanism (DCIM), a client-based cache that is implemented on top of a previously proposed architecture for caching data items in mobile ad hoc networks (MANETs). This project deals with an efficient data routing in wireless mobile networks that overcomes the problem of data loss during a very low connectivity range. DCIM introduces that is totally client-based architecture. In this paper, DCIM is analyzed to assess the delay analysis and Packet Loss Rate when compared. The Packet Delivery Ratio, Throughput, Energy-consumption, Packet LOSS RATE, DELAY ANALYSIS are reported versus several variables, where DCIM showed to be superior when compared to the other systems.

IndexTerms— *TTL, Piggybacking, MANET, DCIM.*

I. INTRODUCTION

MOBILE devices are the blocks of mobile ad-hoc networks (MANET). MANET are characterized by less resources, high mobility, low availability and lack of access to the data source (server). In MANET environments, data caching is essential because it increases the availability of mobile devices to access desired data, and improves overall performance. In a typical caching architecture, several mobile devices cache data that other devices frequently access query.

This is the first complete client side approach achieving superior availability, delay, and traffic performance.[3][6]

Objectives:

The main aim of the project is to reduce the delay and data loss due to low connectivity range and hence reduce the burden of the server to know about their caches.

II. PROPOSED SYSTEM

We propose a pull-based algorithm that implements TTL, piggybacking, and perfecting. Cached data items are assigned TTL values that update rates at the data source.[3][6]. The data source, which in turn sends the cache devices the actual items that have changed, or invalidates them, based on request. This approach, which we call distributed cache invalidation mechanism (DCIM), works on top of the previously proposed architecture COACS cooperative caching architecture.

TTL algorithms are popular due to their sufficiently good performance, and flexibility to assign TTL values to individual data items.[3][6] This is the first complete client side approach employing TTL and achieving availability, delay, and traffic performance.[3][6]

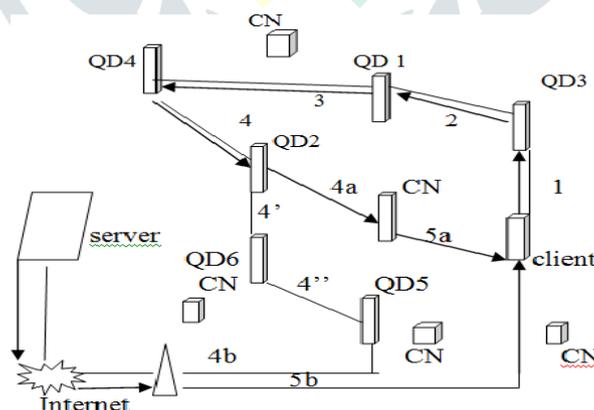


Fig.1. Overview of DCIM basic design

Description:

In the above figure it is shown that one of mobile nodes has a low range [11][9]. Hence, to overcome the delay and data loss, energy consumption will be overall reduced. It forwards the requests through the intermediate node query directory (QD) and cache node (CN). Cache nodes use data for a temporary memory while the query directory is used to maintain the information about those caches and efficiently directs them to the requesting node [1][2].

Request Node (RN) and Query Directories (QD) :

A node data item sends its request to its nearest QD. If QD finds out the query in its cache data then it forwards the request to the Cache node (CN) caching the item and sends the item to the requesting node (RN). Otherwise, it forwards it to its nearest QD, which has not received the request node. If the request sends all Query Directories (QD's) without being found, it gets forwarded to the server which sends the data item to the Request Node (RN).

III. RESULT OF SIMULATION

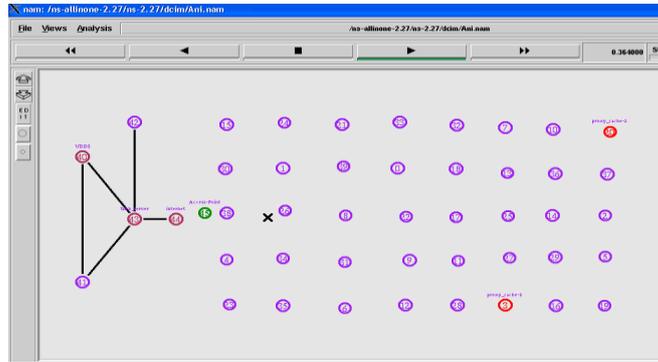


Fig. 2 select two different node in mobile network

Two Proxy cache node select in mobile network for the Requesting Node sends the Request for the particular Data Items. The Request flow through Query Directory(QD) to Cache Node(CN). And

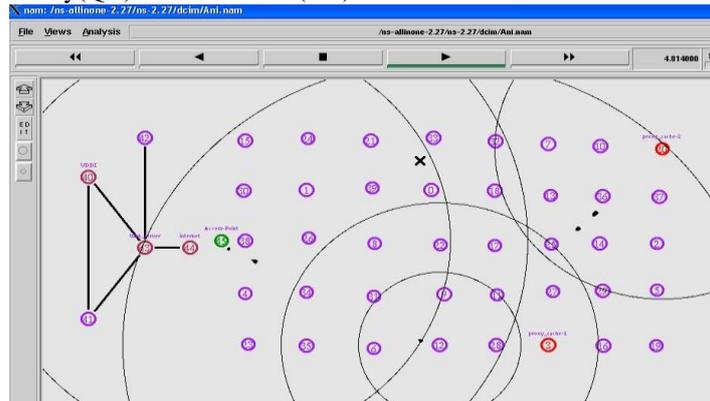


Fig 3. Request Handling

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 . Otherwise, it forwards it to its nearest QD, which has not received the request yet. If the request traverses all QD's without being found, a miss occurs and it gets forwarded to the server via access point which sends the data item to the RN. Performance analysis of DCIM are given below.



Fig.4. Effect of Packet data rate for different node



Fig. 5 Effect of energy consumption for different node



Fig. 6 Effect of Throughput in mobile network

In mobile network, throughput is the amount of data moved successfully from one place to another in a given time period

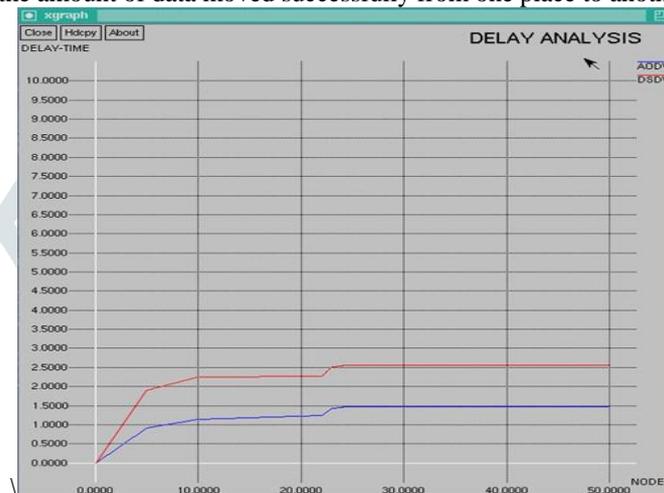


Fig. 7 Effect of Delay for different node

The cache node is data's initially, the random data from the server and then maintained for a calculated TTL based on the demand for that data i.e. Number of requests for that particular data that is maintained by that cache node. Whenever a data is requested by the request node, hit rate is increased. He created and since it does not contain any guest . The request containing the information about the data is forwarded by the query directory to the list of all the cache nodes in its vicinity. The data is received through the query directory; the response is indicated through the textbox in the response. The cache node maintains the hit rates based on the demand request for that particular data which in this case is the java file.

III. CONCLUSION & FUTURE SCOPE

The results have demonstrated that, in comparison with the existing methods, our proposed scheme is more effective and efficient in accessing the data, reducing response time and improving the Performance of the system. The project has reduced the delivery failures in the scenario of low connectivity ranges.

With the use of Distributed Caches the fetching of data has been lot easier and less time consuming. The cache node is made to burden of the server to maintain the state information about the cache nodes is strictly reduced compared to other mechanisms.

In future research this will show, how securely the data transfer can be made with in the nodes involved in the scenario. The process that is taking place between the nodes involved can be made to act as background process without the knowledge of the user of the mobile node..

REFERENCES

- [1] J. Cao; Y. Zhang, G. Cao, X. Li, "Data Consistency for Cooperative Caching in Mobile Environments," Computer , v.40, n.4, pp.60-66, 2007
- [2] P. Cao, C. Liu, "Maintaining strong cache consistency in the World-Wide Web," IEEE Trans. Computers, v. 47, pp. 445–457, 1998.
- [3] J. Jung, A.W. Berger, H. Balakrishnan, "Modeling TTL-based internet caches," IEEE INFOCOM 2003, San Francisco, CA, March 2003
- [4] H. Artail, H. Safa, K. Merhad, Z. Abou-Atme, N. Sulieman, "COACS: A Cooperative and adaptive caching system for MANETS", IEEE TMC, v.7, n.8, pp. 961-977, 2008
- [5] K. Merhad, H. Artail, "SSUM: Smart Server Update Mechanism for Maintaining Cache Consistency in Mobile Environments," IEEE TMC, v. 9, n. 6, pp.778-795, 2010.
- [6] Y. Fang, Z. Haas, B. Liang, Y. B. Lin, "TTL prediction schemes and the effects of inter-update time distribution on wireless data access," Wireless Networks,v. 10, pp. 607-619, 2004.
- [7] J. Jing, A. Elmagarmid, A. Helal, R. Alonso, "Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," Mobile Networks and Applications, pp. 115-127, 1997.
- [8] H. Maalouf, M. Gurcan, "Minimisation of the update response time in a distributed database system," Performance Evaluation, v. 50, n. 4, pp. 245-66,2002.
- [9] Z. Wang, S. Das, H. Che, M. Kumar, "A scalable asynchronous cache consistency Scheme (SACCS) for mobile environments," IEEE TPDS, v. 15, n.11, pp. 983- 995, 2004.66
- [10] "Cache Invalidation Scheme for Mobile Computing Systems with Real-time Data" Joe Chun-Hung Yuen, Edward Chan, Kam-Yiu Lam and H. W. Leung
- [11] [HL97] Hu, Q. and Lee, D.L., "Adaptive Cache Invalidation Methods in Mobile Environments", Proc. 6th IEEE Intl. Symp. On High Performance Distributed Computing Environments,1997

