ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Wireless Ad hoc Network using Counter Based Cache Replacement Algorithm

A.RadhaVasan[1], N.Vinoth[2], M.Abdul Rahuman[3], S.Silambarasan[4]

Senior Assistant Professor, Christ College of Engineering and Technology, Puducherry[1]

UG Student, Christ College of Engineering and Technology, Puducherry[2]

UG Student, Christ College of Engineering and Technology, Puducherry[3]

UG Student, Christ College of Engineering and Technology, Puducherry[4]

Abstract: Wireless adhoc network is used to access network for user anywhere at anytime. Data access will take more time in Wireless adhoc network and so data caching is used to reduce the data access. Cache algorithm is used to improve the caching performance and the network become more efficient. In this algorithm, we are using the cache counter so that each time the request is given which the counter value is increased. If the counter value exceed means it should be removed and the space in the counter will be free so that if too many request arrive means the crash will not occur and reduce the capacity.

Index Terms-Adhoc Network, Caching, AD, Multihop path.

1. INTRODUCTION

Adhoc network is used to form the temporary network and it does not need the centralized administration. The nodes are formed in the adhoc network dynamically. The nodes in the network is used to transfer the data from one node to another. In this network, the mobile node will act as both the host and the router which is used to forward the packets in the network. The node in the adhoc network will analyze the other nodes and then it will act as the router to transfer the data. This paper proposes a new approach, counter-based cache replacementthat significantly improves highly-associativecache replacement performance.

By using the cache server in the adhoc network the data access delay will be reduced and the request and response will be increased. The counter-based cache replacement algorithm is used to improve the efficiency of the adhoc network.

2. EXISTING SYSTEM

When data are delivered through multihop paths, caching the data at intermediate nodes can significantly reduce the message cost and consequently save various resources, from network bandwidth to battery power. Accessing data at cache node can also help reduce data access delay (AD).Existing works on cache placementmainly focus on how to make use of the data accessfrequency information and network topology informationin selecting cache nodes.The main drawback is quite a lot of work has been conducted for data caching in wireless ad hoc networks, including cache placement cache discovery and cache consistency. Data caching has been widelyused to reduce data access cost in traditional computer networks.When data aredelivered through multihop paths, caching the data atintermediate nodes can significantly reduce the messagecost and consequently save various resources, from networkbandwidth to battery power. Accessing data at cache nodecan also help reduce data access delay (AD).The cache placement problem in ad hoc networks hasbeen proved to be NP-hard, even if only one data item isconsidered. Existing works on cache placementmainly focus on how to make use of the data accessfrequency information and network topology informationin selecting cache nodes. Forcache discovery, recent research has been focused oncombining passive and active query approaches.Overhearing-aided data caching algorithm consists oftwo parts: cache placement and cache discovery. The cacheplacement part is used for a node to choose data items tocache based on our proposed data access cost function. Thecache discovery part realizes the overhearing-aided dataaccess by providing a node with the mechanism to servedata requests by overhearing.Due to the openness of wireless links, a network nodetransmits data in a broadcast way by nature, so a packet canbe received by any node within the transmission range evenif the node is not the intended target of this transmission.

2.1 Drawbacks

- Quite a lot of work has been conducted for data caching in wireless adhoc networks, including cache placement cache discovery and cache consistency.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

3. PROPOSED SYSTEM

In this approach, each line in the cache is augmented with an event counter that is incremented when an event of interest, such as a cache access to the same set, occurs.We propose two counter-based replacement algorithms, whichdiffer by the type of intervals during which the events arecounted. When the counter exceeds a threshold, the line .expires., and becomes evictable.This paper proposes a new approach, counter-based cachereplacementthat signi_cantly improves highly-associativecache replacement performance.When the counter exceedsa *threshold*, the line .expires., and immediately becomesevictable. We observe that data blocks in applications showdifferent reuse patterns.The advantage of this the cache stored in the counter and it stores the request so the data access can be accessed easily and data cost will be reduced frequently.

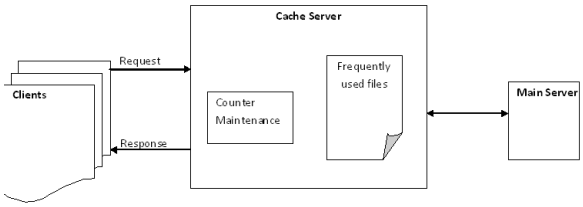

Fig 1 Counter Based Cache Replacement

3.1 Advantages

✓ The data access delay will be reduced and the request will be sent immediately.
✓ Using Cache server, the request will be stored and the request will be retrieved easily.

4. Modules

- Working of Adhoc Network
- Cache Server Working
- Data Transfer between Nodes

4.1 WORKING OF ADHOC NETWORK

The Adhoc network does not contain no master and slave nodes. When the node form the network means the nodes are automatically connected and the data transfer takes place from source node to destination node.Accessing data at cache node can also help reduce data access delay (AD). The main drawback is quite a lot of work has been conducted for data caching in wireless ad hoc networks, including cache placement cache discovery and cache consistency. Existing works on cache placementmainly focus on how to make use of the data accessfrequency information and network topology informationin selecting cache nodes. When data aredelivered through multihop paths, caching the data atintermediate nodes can significantly reduce the messagecost and consequently save various resources, from networkbandwidth to battery power. Accessing data at cache nodecan also help reduce data access delay (AD).

4.2 Cache Server Working

A cache server is a dedicated network server or service acting as a server that saves content locally. By placing previously requested information in temporary storage, or cache, a cache server both speeds up access to data and reduces demand on an enterprise's bandwidth. Cache servers also allow users to access content, including rich media files or other documents. A cache server is sometimes called a "cache engine."

A cache server will act as an intermediate and it will get the incoming requests so that if too many request arrive means the server will not crash. The requests will be stored in the cache server and the cache server in the network will act as the proxy server.

4.3 Data Transfer between Nodes

Node A's request to cache server or directly to another Node B. Suppose, the file is not available in cache server then Node A's request reaches to Node B directly. The cache server acts as an interface between Node A and Node B. First, Node A's request reaches to cache server if that file is available in cache. The copy of requested file is stored in cache server. Node A is directed to Node B, if that file is not available. Cache server maintains counter value for file. The counter value is incremented when each Node A requested the particular file. If the counter reaches the threshold value, the file is replaced.

5. FRAMEWORK

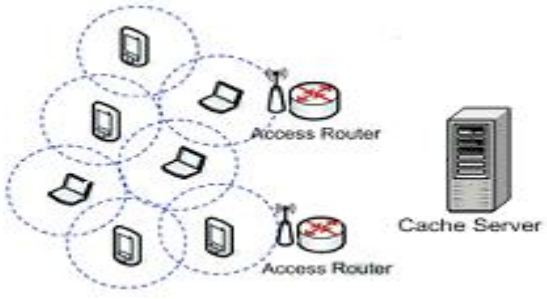

Fig 2 Block Diagram

6. CONCLUSION

We have compared our approach with two existing approaches: sequence-based approach and time-based approach. Compared to the sequence-based approach, our counter-based algorithms achieve better prediction coverage

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

and accuracy, lead to better performance improvement, and are more space efficient. Compared to the time based approach, our counter-based algorithms perform better and are easier to implement because they are more architecture independent. The result show that, compared with one representative algorithm , our proposed algorithm can significantly reduce both message cost and access delay.

REFERENCES

[1] I. Baev and R. Rajaraman, "Approximation Algorithms for Data Placement in Arbitrary Networks," Proc. Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '01), 2001.
[2] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-Like Distributions: Evidence and Implications," Proc. IEEE INFOCOM '99, 1999.
[3] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks," Proc. IEEE INFOCOM, 2002.
[4] P. Cao and C. Liu, "Maintaining Strong Cache Consistency in the World Wide Web," IEEE Trans. Computers, vol. 47, no. 4, pp. 445-457, Apr. 1998.
[5] L. A. Belady. A study of replacement algorithms for virtualstorage computers. *IBM Systems Journal*, 1966.
[6] K. Beyls and E. D'Hollander. Compile-Time Cache Hint Generation for EPIC architectures. In *2nd Workshop on ExplicitlyParallel Instruction Computing Architecture and Compilers*, 2002.
[7] G. Chen, N. Vijaykrishnan, M. Kandemir, M. Irwin, and M. Wolczko. Tracking Object Life Cycle for Leakage Energy Optimization. In *Proc. of the ISSS/CODES joint conference*, 2003.
[8] A.-C. Lai, C. Fide, and B. Falsa_. Dead block prediction and dead block correlating prefetchers. In *28th Intl. Symp. on ComputerArchitecture*, 2001.
[9] J. Lee, Y. Solihin, and J. Torrellas. Automatically Mapping Code on an Intelligent Memory Architecture. In *7th Intl. Symp.on High Performance Computer Architecture*, 2001.
[10] W.-F. Lin and S. K. Reinhardt. Predicting last-touch references under optimal replacement. *University of Michigan Tech. Rep.CSE-TR-447-02*, 2002. [11] R. L. Mattson, J. Gecsei, D. Slutz, and I. Traiger. Evaluation Techniques for Storage Hierarchies. *IBM Systems Journal*, 9(2), 1970. 8
[12] M. Takagi and K. Hiraki. Inter-Reference Gap Distribution Replacement: an Improved Replacement Algorithm for Set-Associative Caches. *Proc. of the 18th Intl. Conf. on Supercomputing*, 2004.