

Invalidation Method for Wireless Mobile Network to Maintain Consistency

Asmita R. Shelke
M B E Society's College of Engineering,
Ambajogai Maharashtra, India

B.M. Patil
M B E Society's College of Engineering,
Ambajogai Maharashtra, India

ABSTRACT

This paper is proposed for the distributed cache invalidation mechanism (DCIM), it is a client-based cache consistency scheme which is implemented on the top of a previously proposed architecture for the caching data items in mobile ad hoc networks (MANETs), namely COACS, where special nodes cache the queries and the addresses of the nodes that store the responses to these queries. Caching frequently accessed data items on the client side is an effective technique for improving performance in a mobile environment. The classical cache invalidation strategies are not suitable for the mobile environments due to frequent disconnections and mobility of the clients. The huge amount of demand has been created due to trend toward wireless communications and advances in mobile technologies are which increasing the consumer demands for ubiquitous access to Internet-based information and services. DCIM can be implemented using ns2, and compared against the client-based and server-based schemes to assess its performance experimentally. Hear introduce DCIM that is totally client-based. DCIM is a pull-based algorithm that implements adaptive time to live (TTL), piggybacking, and pre-fetching, and provides near strong consistency capabilities.

Keywords

MANET, data caching, pull-based, TTL, shared data.

1 INTRODUCTION

Because of the recent technological advances in high-speed wireless network, mobility support, and portability, wireless mobile device has become an essential tool to access the Internet services and information wirelessly. For example, smart cellphone is receiving a great attention and becoming popular. 42 million users 1 out of 239 million subscribers in the US already own a smart phone in 2010. 45% and 40% of all adult Americans are the wireless Internet users with laptop and cell phone, respectively in 2010 .Approximately six American adults out of ten access the Internet wirelessly anyhow. Typically, the explosive growth in applications running on the smart cell phone has fueled users to easily access the Internet anytime and anywhere. The Mobile ad-hoc network is one of the most common major areas for research in the recent years. It became one of the new emerging technologies this MANET helps the user to communicate with each other without any physical infrastructure, regardless of their geographical location.

Due to this reason MANETs are sometimes called as an "infrastructure less" network. The proliferation of more cheaper, small and more power full device make MANET a faster growing network. The mobile devices are the building blocks of MANET. They are typically characterized by limited resources, high mobility, transient availability and lack of direct access to the server (data source). One of

the most important and the effective technique for the MANET environment is data caching because it is used to improve the overall system performance and also increases the ability of the mobile devices to access the desired data. The caching architecture describes a caching technique, where as in this technique several mobile devices cache data that other devices frequently access or request. The data items that cache node stores can be anything like data base records, web pages, ftp files etc.

Caching the frequently accessed data on the client side is an effective technique for improving performance in a mobile environment. Average data access latency is reduced as several data access requests can be satisfied from the local cache, thereby obviating the need for data transmission over the scarce wireless links. However, frequent disconnections and the mobility of the clients make cache consistency a very challenging problem. The effective cache invalidation strategies are necessary to ensure the consistency between the cached data at the clients and the original data stored at the server. When the cache techniques are used, the 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. Problems related to cache consistency have been widely studied in many other systems such as multiprocessor architectures, the distributed file systems, distributed shared memory, and the client-server database systems. However, if the client moves to another cell, it has to notify the server. This implies some of the restrictions on the freedom of the clients. In the stateless server approach, the server is not aware of the state of the client's cache. The clients are needs to query the server to verify the validity of their caches before each use.

The different approaches apply the different techniques to construct the IR, these schemes to maintain the cache consistency by periodically broadcasting the IR. The IR-based solution is attractive because it can scale to any number of clients who listen to the IR. However, the IR based solution has some disadvantages. For example, this approach has along query latency since the client must listen to the next IR and use the report to conclude whether its cache is valid or not before answering a query. Hence, the average latency of answering a query is the sum of the actual query processing time and half of the IR interval. If the IR interval is long, the delay may not be able to satisfy the requirements of many clients. In most previous IR-based algorithms, when a client needs an invalid cache item, it requests the data from the server, and the server sends the data to the client. Although the approach works fine for some cold data items which are not cached by many clients; it is not effective for hot data. For example, suppose a data item is frequently accessed (cached) by 100 clients, updating the data item once may generate 100

uplink (from the client to the server) requests and 100 downlink (from the server to the client) broadcasts obviously, it wastes a large amount of wireless bandwidth and battery energy.

Thus, cache inconsistency poses a serious threat to the availability of Internet services. This is simply because during the cache inconsistency period, the clients served with out-of-date DN2IP mappings cannot reach the appropriate Internet servers or end-hosts. Once it happens, the clients have no idea of what is the cause of service unavailability: is it due to server shutdown, network failure, or something else? An aggressively small TTL (on the order of seconds) can lower the chance of cache inconsistency, but at the expense of significant increase of the DNS traffic, name resolution latency, and the workload of domain name servers, which seriously degrades the scalability and performance of DNS.

In this paper, propose a proactive DNS cache update protocol, called DNS cup, working as middleware to maintain strong cache consistency among DNS name servers and improve the responsiveness of DNS-based service redirection. The core of DNS cup uses a dynamic lease technique to keep track of the local DNS name servers whose clients are tightly coupled with an Internet server. Upon a DN2IP mapping change of the corresponding Internet server, its authoritative DNS name server proactively notifies these local DNS name servers still holding valid leases. While the notification messages are carried by UDP, dynamic lease also minimizes storage overhead and communication overhead, making DNS cup a Light-weight and scalable solution. Based on client query rates (or service importance to their clients), it is the local DNS name servers themselves that decide on whether or not applying for leases (or renewal) for an Internet service. On the other-side, the authoritative DNS name server grants and maintains the leases for the DNS resource records of the Internet service. The duration of a lease is dependent on the DN2IP mapping change frequency of the specific DNS resource record.

The distribution of the Web content is governed by HTTP (Hyper-Text Transfer Protocol). Each object has a URL specifies its "location" and its authoritative server. The object is requested by sending an HTTP request and the content is sent back on the respective HTTP response. The response includes a header with important information on the object, including cache directives. The directives specify if the object can be cached, and may provide explicit expiration time or information that can be used to compute one. When an object is requested from the cache then if the cache has a fresh (non-expired) copy, the request is processed locally. If the cached copy is stale (expired), it must be validated by contacting a server with a fresh copy. To this end, HTTP provides conditional GET requests. Similarly, if there is no cached copy, the cache must obtain a fresh copy. Requests for which the cache does not have a fresh copy and thus must contact an external server are termed cache misses.

2 LITERATURE SURVEY

Wensheng Zhang et al Cooperative caching published in 'Secure Cooperative Cache Based Data Access in Ad Hoc Networks' which allows the sharing and the coordination of the cached data among multiple nodes can be used to improve the performance of data access in ad hoc networks. When caching is used, data from the server is replicated on the caching nodes. Since a mobile node may return to the cached data, or modify the route and forward a request to a caching node, it is very important that the mobile nodes do not

maliciously modify data, drop, or forward the request to the wrong destination. In this paper, identify the possible security attacks on cache consistency and propose a randomized grouping based schemes for intrusion detection, damage recovery and intruder identification.

Guhongcao and Lianzong Yin, as mentioned in Cooperative Cache Based Data Access in Ad Hoc Networks In ad hoc networks, mobile nodes communicate with each other using multi-hop wireless links. Due to lack of the infrastructure support, each node acts as a router, forwarding data packets for other nodes. Most previous research in ad hoc networks focused on the development of dynamic routing protocols that can efficiently find the routes between the two communicating nodes. Although routing is an important issue, the ultimate goal of ad hoc networks is to provide the mobile nodes with access to information.

3 PROPOSED APPROACH FRAMEWORK AND DESIGN

3.1 Problem Definition

Due to tremendous growth of cloud computing services in real time applications allows end users of cloud to share their data with each other easily. Multi user data sharing should be secure and integrity should be achieved on cloud. To achieve data integrity on cloud there are different methods introduced for public integrity auditing of users data by considering the different practical features. Those features are like less storage overhead, less auditing cost for communication, supporting dynamic data etc. Such methods considering only data owner can modify the shared data on cloud, this becomes the limitation, and applicable only in applications those are client read only. Therefore multi user modification of shared data is recently proposed in few methods with assurance of integrity. But practically such methods failed to achieve the said objectives. To achieve public integrity auditing dynamic data sharing over cloud by multiple users' modification, recent efficient method proposed. Practically this method achieve better performance, however still their scope of improvement as some of methods used are poor for integrity with this method.

3.2 Proposed system

In our proposed System distributed cache invalidation mechanism (DCIM), a client-based cache consistency scheme that is implemented on top of a previously proposed architecture for caching data items in mobile ad hoc networks (MANETs), namely COACS, where special nodes cache the queries and the addresses of the nodes that store the responses to these queries. Previously proposed a server-based consistency scheme, named SSUM, whereas in this paper DCIM that is totally client-based. DCIM is a pull-based algorithm that implements adaptive time to live (TTL), piggybacking, and perfecting, and provides near strong consistency capabilities. Cached 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, where as unexpired ones but with high request rates are perfected from the server. In this paper, DCIM is analyzed to assess the delay and bandwidth gains (or costs) when compared to polling every time and push-based schemes. DCIM was also implemented using ns2, and compared against client-based and server-based schemes to assess its performance experimentally. The consistency ratio, delay, and overhead traffic are reported versus several variables, where

DCIM showed to be superior when compared to the other systems.

Below figure 1 showing the proposed system block diagram and details of algorithm proposed.

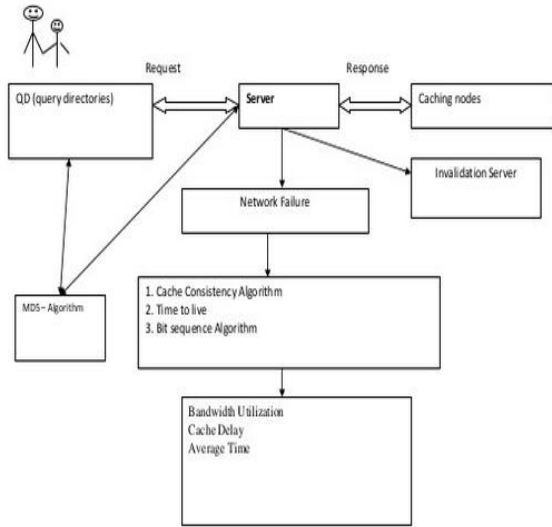


Fig 1: proposed system

3.3 Mathematical Module

$$P_R(t) = \lambda_R e^{-\lambda_R t}, P_U(t) = \lambda_U e^{-\lambda_U t}.$$

To find the response time and traffic gains, took concepts from our early study related to the average no.of hops required in the many situations in the calculations.

- HC is the average number of hops between the corner of the topology and a randomly selected node. When packet is sent between the server and a node in the network then it's used.
- HR is the expected number of hops between any two randomly selected nodes.
- HD is the expected number of hops to reach the QD containing the reference to the requested data, in the case of a hit.
- T_{in} is the transmission delay between two neighboring nodes (i.e., one hop delay), while T_{out} is the round trip time between the MANET and the server.
- SD is the size of the data packet and SR is the size of the request.

In what follows, discuss the time response and bandwidth gains for DCIM when compared with the PET and PBS schemes. show that although a push-based stateful scheme could offer slightly smaller response times, it generates considerably more traffic.

3.4 Response Time Gain

In this derive the response time gain of DCIM over PET and over PBS, and show them in (2) and (3)

$$GT_{PET} = T_{RIT} - P_{SC1} \times T_{RIT} - (1 - P_{SC1}) \times T_{MAN}, \quad (2)$$

$$GT_{PBS} = -P_{SC1} \times T_{RIT} + P_{SC1} \times T_{MAN}, \quad (3)$$

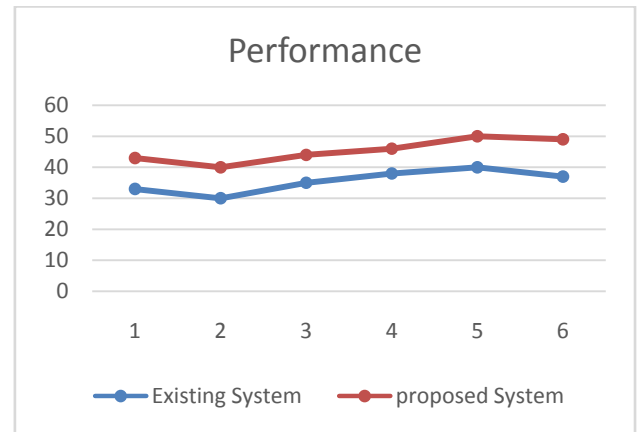
$$\text{where } T_{RIT} = T_{out} + T_{in}(2H_C + H_D), \quad (4)$$

$$T_{MAN} = T_{in}(2H_R + H_D), \quad (5)$$

$$P_{SC1} = \lambda_U \times T_{RIT} \times e^{-\lambda_U \times T_{RIT} - 1}. \quad (6)$$

As implied from (2), the gain mainly depends on the update rate, which causes it to decrease slightly when it increases. Note that for PBS, consider the best case scenario, where items are always up-to-date and answered directly from the CNs. In the graph, the lower curve confirms that in DCIM the majority of the requests are answered from the MANET, this is why the time response difference is less than 10 ms.

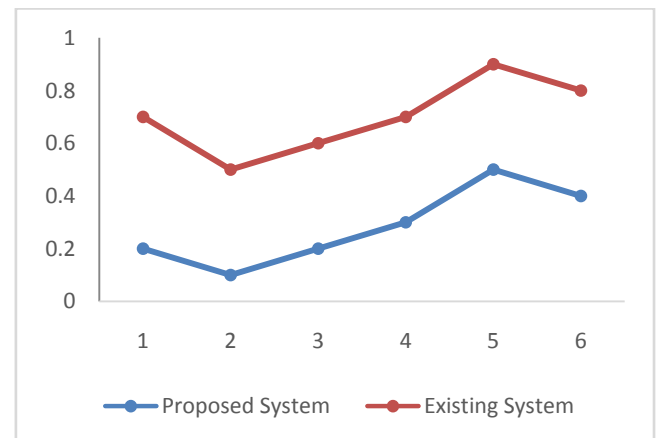
4 EXPERIMENTAL RESULTS



X axis-Hit rate.

Y axis- Inter Request Interval

Fig 2: Performance



X axis-Time response (ms).

Y axis- (item/seconds)

Fig 3: Time

Fluctuating the Request Rate

In this trial, the inter-request interval was varied between 5 s and 120 s, and the results are plotted in the graph. It is evident that poll every time delivers the highest consistency ratio, since the requested items are always validated, and the items are always fresh, except in certain cases when they modified just after being authenticated. But, when using fixed TTL, the caches might serve stale items as in the case of TTL = 500 s, but this likelihood decreases when the TTL is less than the update interval as in the case of TTL = 100 s.

Effectiveness of Priority Based Scheduling

In the Internet environment, both access interval and service time distribution are different from widely used synthetic workloads. Therefore, we have used real traces for performance evaluation. The high variance of the inter-arrival time and service rate deteriorates system performance. As see from Figures 2 and 3, with the increase in server utilization, response time increases much faster under high utilization. Curve (c) in Figure 2 is the slowdown curve of requests of our model without priority distinction. Curve (d) is the slowdown curve of a general M/M/1 queuing system. The disparity between curve (c) and (d) indicates the impact of self-similar traffic on the system utilization. The utilization of 0.5 in the simulator leads to about the same level of delays as utilization of 0.8 in a Poisson distribution.

Performance degradation of high priority tasks with priority-based scheduling happens at a much higher utilization compared to the non-priority-based model. Curves (a) and (b) in Figure 2 are the slowdown curves of low priority requests and high priority requests, respectively. The ratio of high priority is 0.5 and is uniformly distributed in the whole arrival sequence. The introduction of priority queuing causes a steep rise of low priority requests response time at a system utilization of 0.4. High priority requests incur low delay even when the system approaches full utilization.

5 SYSTEM MODEL

The system consists of a MANET of wireless mobile nodes interested in data generated at an external data source connected to the MANET using a wired network (e.g. Internet) via Wi-Fi Access Points (APs). Nodes that have direct wireless connectivity to an AP act as gateways, enabling other nodes to communicate with the data source using multi-hop communication. For example, Node CN4 in Fig. 1 is accessing the server through N4 and then CN1, which in turn acts as a gateway by connecting to the Internet via the AP. The proposed system builds on top of COACS, which introduced in and did not include provisions for consistency. Briefly, the system has three types of nodes: caching nodes (CNs) that cache previously requested items, query directories (QDs) that index the cached items by holding the queries along with the addresses of the corresponding CNs, and requesting nodes (Rns) that are ordinary nodes. Any node, including a QD or a CN, can be a requesting node, and hence, an RN is not actually a special role, as it is only used in the context of describing the system. One, therefore, might view the employed caching system as a two layered distributed database. The first layer contains the QDs which map the queries to the caching nodes which hold the actual items that are responses to these queries, while the second layer is formed by the CNs.

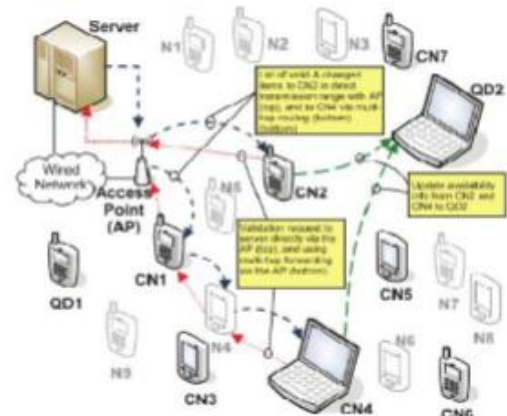


Fig 4: Overview of the basic design [10]

Fig. 4 shows a scenario for illustration purposes where two CNs are sending cache validation requests to the server (dotted arrows) via gateway nodes and through the Access Point. The server replies back with lists of valid and changed data items (short-dashed arrows) to the CNs, which in turn update the corresponding QDs asynchronously about the items they cache (long-dashed arrows).

5.1 Basic Design

Use three messages which are already introduced in COACS. Its description can be shown in the table

Table. 1. Packets used in DCIM

Packet	Function	Description
CURP	Cache Update Request	Sent from CN to server to validate certain data items
SVRP	Server Validation Reply	Sent from server to CN to indicate which items are valid
SUDP	Server Update Data	Sent from server to CN. It includes updated data items and timestamps

Fig. 5 shows the basic interactions of distributed cache invalidation mechanism through a scenario in which more than one RN is submitting a data request packet (DRP) for a query indexed in the QD. QD maintain two FIFO queues for store the DRP. One queue consist the DRP messages that become forward request to CN in the case of a hit.

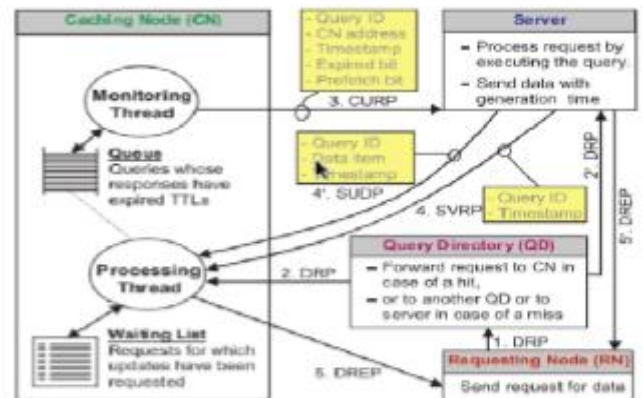


Fig 5: Packets used in DCIM Interaction between 2 nodes in the system [10]

6 CONCLUSION AND FUTURE WORK

This paper presented a client based cache consistency scheme based on distributed cache invalidation method in wireless mobile networks. This scheme is based on pull-based approach. It is to increase the accuracy of its consistency ratio, traffic, and query delay.

In this presented pull-based cache consistency scheme for MANET's that relies on estimating the inter update intervals of data items to set their expiry time. The proposed system considered the network intrusion and extended the security features of DCIM. The results show the comparison of DCIM with normal caching protocol. Thus DCIM gives better performance as compared to normal network.

7 REFERENCES

- [1] L. Bright, A. Gal, and L. Raschid, "Adaptive Pull-Based Policies for Wide Area Data Delivery," *ACM Trans. Database Systems*, vol. 31, no. 2, pp. 631-671, 2006.
- [2] C.P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell, "A Hierarchical Internet Object Cache," *Proc. Ann. Conf. USENIX Ann. Technical Conf.*, p. 13, 1996.
- [3] V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM Computer Comm. Rev.*, vol. 25, p. 187, 1995.
- [4] X. Chen and P. Mohapatra, "Lifetime Behavior and its Impact on Web Caching," *Proc. IEEE Workshop Internet Applications*, pp. 5461, 1999.
- [5] Y. Sit, F. Lau, and C-L. Wang, "On the Cooperation of Web Clients and Proxy Caches," *Proc. 11th Int'l Conf. Parallel and Distributed Systems*, pp. 264- 270, July 2005.
- [6] P.Papadimitratos and Z. Haas, "Secure Data Transmission in Mobile Ad Hoc Networks," *Proc. ACM Workshop Wireless Security*, pp. 41-50, 2003.
- [7] W. Zhang and G. Cao, "Defending Against Cache Consistency Attacks in Wireless Ad Hoc Networks," *Ad Hoc Networks*, vol. 6, pp. 363-379, 2008.
- [8] H. Maalouf and M. Gurcan, "Minimisation of the Update Response Time in a Distributed Database System," *Performance Evaluation*, vol. 50, no. 4, pp. 245-66, 2002.
- [9] T. Hara and S. Madria, "Dynamic Data Replication using Aperiodic Updates in Mobile Ad Hoc Networks," *Proc. Database Systems for Advanced Applications*, pp. 111-136, 2004.
- [10] K. Fawaz and H. Artail, "A Two-Layer Cache Replication Scheme for Dense Mobile Ad Hoc Networks," *Proc. IEEE Global Comm. Conf. (GlobeCom)*, Dec. 2012.