

# PEER TO PEER CACHE RESOLUTION MECHANISM FOR MOBILE AD HOC NETWORKS

Preetha Theresa Joy<sup>1</sup> and K. Poullose Jacob<sup>2</sup>

<sup>1,2</sup>Department of Computer Science, Cochin University of Science and Technology,  
Kochi, Kerala, India.

preetha@mec.ac.in, kpj@cusat.ac.in

## **ABSTRACT**

*In this paper we investigate the problem of cache resolution in a mobile peer to peer ad hoc network. In our vision cache resolution should satisfy the following requirements: (i) it should result in low message overhead and (ii) the information should be retrieved with minimum delay. In this paper, we show that these goals can be achieved by splitting the one hop neighbours in to two sets based on the transmission range. The proposed approach reduces the number of messages flooded in to the network to find the requested data. This scheme is fully distributed and comes at very low cost in terms of cache overhead. The experimental results gives a promising result based on the metrics of studies.*

## **Keywords**

*Cache Resolution, MANET, Cooperative caching, Wireless Networks.*

## **1. INTRODUCTION**

Recent wireless communication growth and the continuous improvement of mobile terminal performance have enabled the use of these technologies in different fields and to consider new applications. There are two categories of wireless mobile networks: networks with infrastructure using access points to connect to a wide range of network and networks without infrastructure which do not presuppose the presence of a wired infrastructure. An ad hoc network is a wireless self configurable network made up of mobile free moving terminals interconnected by wireless connections. Ad hoc networks provide a way of communicating which can be quickly and easily deployed. This makes the ad hoc networks a good choice for applications in which various civil and military fields.

The goal of a computer network is to offer a way for users with different terminals to communicate in order to share data and access services provided by different servers. In mobile ad hoc networks, devices generally have limited energy reserves and processing capabilities. Bandwidth is also a scarce resource, limited by the nature of the wireless medium. In a data-management point of view, these restrictions introduce several issues that need to be addressed. Data transfers must be reduced and mechanisms must be deployed to confront the frequent disconnections and low bandwidth constraints. Therefore it is a challenging task to present the data efficiently by reducing the delay or waiting time to the end user.

Data caching is widely used in various domains to improve data access efficiency, by reducing the latency experienced by the end users. In wireless mobile network, holding frequently accessed

data items in a mobile node's local storage can reduce network traffic, response time and server load. Caching in ad hoc networks is effective because a few resources are requested often by many users, or repeatedly by a specific user, which is known as the locality of reference. To have the full benefits of caching, the neighbor nodes can cooperate and serve each other's misses, thus further reducing the wireless traffic. This process is called cooperative caching. Since the mobile nodes can make use of the data stored in another node's cache the effective cache size is increased.

The implementation of a good cooperative caching technique essentially involves four major design considerations: cache placement and resolution, cache admission control, cache replacement and cache consistency maintenance [1]. The cache admission control module decides whether a received data item is cacheable or not, and a cache resolution module decides how to fetch the data from the neighboring nodes when there is a local miss. Since the network node has limited memory, it can cache multiple data items subject to its memory capacity. The objective of cache admission control module is to store more distinct data items in the given cache space. This reduces the data traffic across the network since more number of requests can be served from the neighboring nodes. The cache consistency module is to maintain cache consistency, ie, ensuring that each node caching the data item is aware of the data update at the source. Due to the limited cache space, replacement policies decide which items should be evicted from the existing cache to make way for new ones. In this paper we focus on cache placement and resolution.

Towards the goal of improving the performance in cooperative cache, we propose a novel cache resolution scheme which uses a split table approach to resolve the cached data request which in turn reduces latency and message overhead. More precisely, our algorithm splits the neighboring nodes in to two sets based on communication range. The mobile devices keep the information about the neighboring nodes in two tables, top level nearest node cache table and next level nearest node cache table. The searching time and the request messages flooded in to the network is reduced by this method. Moreover, due to simplicity, this approach does not have any additional overhead in implementation. From the evaluation results we state that our proposal yields significant improvements compared to the previous solutions.

The remainder of the paper is structured as follows. Section 2 investigates the problem of cache resolution, Section 3 reviews the related works in cache resolution, Section 4 presents the system model and assumptions made and detail the proposed cache resolution, Section 5 describes simulation environment and metrics, section 6 gives the detailed analysis of results and section 7 concludes the paper.

## **2. CACHE RESOLUTION**

Cache resolution in ad hoc networks constitutes an issue that several researchers are focusing on. There are a few interesting studies on this subject. Two basic mechanisms used to discover cache in ad hoc networks are broadcasting and cluster based approach. The first method is based on request/response message exchanges in which a user interested in discovering a cache broadcasts a discovery request containing information on the service requested and the node offering the data sends a reply. In the second method, a node known as cluster head is used to temporarily register the information about the cached data in the neighboring nodes. A node searching for data will first look in to the cluster head node to locate the data and then directly send a request to the corresponding node.

Although first method is the simpler version, it relays on flooding to broadcast the data request. Flooding increases network contention and overhead when the network density is high. In

contrast, cluster based approach has been best owed with the features of reduced overhead by having a coordinator node which manages the cache resolution. The neighboring node which possesses the data can be easily found out checking the lookup table maintained by the cluster head. The disadvantage of this approach is that group maintenance is difficult due to the mobility of nodes. The control node may get disconnected which causes excessive overhead [4].The number of entries in the look up table increases when the network density is high. To maintain the correct status of the network, these tables must be frequently updated. This involves information exchange between the nodes which in turn increases the traffic overload in a dense network. To circumvent these drawbacks we designed a cache resolution protocol among the mobile nodes. Our approach is based on a distributed cache discovery algorithm and reduced cache table entries to find the location of the requested data. An advantage of using this technique is that it reduces network traffic as flooding is not used. This technique also avoids the drawback of group maintenance by having a distributed approach. To reduce the number of entries in the look up table we split the tables in two based on transmission range.

### 3. RELATED WORK

The main challenge in cooperative cache is to instantly find a cache containing the desired data. To facilitate data discovery in cooperative cache, some schemes [2][3][5][6][7] based on broadcasting and cluster based approach [9][12] is proposed in literature. These approaches differ in how the data request is resolved to find the location of the data stored in the neighboring nodes. In broadcast based approach the mobile nodes broadcast the request to find a node with the required data. On the other hand in cluster based approach, some mobile nodes are selected as information coordinators which deal with the task of finding the location of the required information. We can also find some discovery mechanism [8] that doesn't comes under this category. Below, we describe some representative cache resolution strategies for cooperative caching from the different groups mentioned above.

Aggregate caching scheme proposed in [5] tries to increase the data accessibility in an Internet based mobile network. A broadcast based information search algorithm called simple search is used to locate the required data item. Whenever a mobile nodes needs some data the request is broadcasted to its adjacent nodes .Upon receiving the broadcast request, the adjacent nodes replies to the request if it has already cached the data, otherwise the request is forwarded to its neighbors until it is acknowledged by an access point or some other nodes which have the requested data. Flooding is the technique used for broadcasting. This algorithm sets a hop limit for the request packet to reduce the traffic in the network. A caching technique which uses cluster based approach can be seen in [9], in which a coordinator node maintains the cluster cache state information of different nodes within its cluster domain. If there is a local cache miss, the coordinator node will find whether the data item is cached in other clients within its home cluster. Another approach for data discover other than the mentioned schemes can be seen in [4] and [10].In [4] a distributed group based approach in which each node maintains a group table and self table. Whenever a data request comes in it first checks in its self table, if data is not found checking is done in the group table which stores the data id and the node which posses the data. Here the mobile nodes interchange information about where the data is present in the network. In [10], a cache resolution technique based on adaptive flooding broadcast is used for searching data in the network. According to this scheme a mobile node uses three schemes; adaptive flooding, profile-based resolution and road side resolution. In adaptive flooding, a node uses constrained flooding to search for items within the neighborhood. In profile-based resolution, a node uses the past history of received requests. In road side resolution, forwarding nodes caching the requested item, reply to the requests instead of forwarding them to the remote data source.

## 4. SYSTEM MODEL

### 4.1. Basic Assumptions

The system environment is assumed to be a mobile ad hoc network which includes a collection of nodes  $n_i$  ( $i=1,2,\dots,N$ ) that communicate with each other in a coordinated manner and  $V_i \subseteq \{n_1, n_2, \dots, n_N\}$  such that  $V_i$  is the set of neighboring nodes for a node 'n'. Data is originated in the server which is assumed to be in a fixed position. Mobile hosts are identified as  $\langle \text{Hostid}, \text{Name} \rangle$  for  $1 \leq i \leq N$ , where  $N$  is the density of the network which is decided by the user. Each data item is uniquely identified by means of data item id  $D_i$  for  $1 \leq i \leq n$  where 'n' is the size of the data base. Each node  $n_i$  maintains a local cache of data items which it frequently accesses. Whenever a node receives a request for the data item, local cache is checked. If the item is found the request is satisfied from the local cache, otherwise the request is forwarded to the neighboring nodes. The cache uses TTL based consistency.

### 4.2. Proposed Scheme

The objective of our cache resolution scheme is to reduce the overall access cost by reducing searching time and data traffic. Most of the current cooperative caching technique uses the entire list of one hop neighbors to find the cached data location. This will place unnecessary load on each node because they need to search all the entries in the table or broadcast a request to all of its neighbors to find the required data. The complexity of data discovery increases with the number of nodes in the network. In this paper we consider a cache resolution technique which uses a split table approach. In the given network each node  $N_i$  maintains a table  $nct_{i1}$ , whose entries represent a set of top level neighbors  $nei_1$ , which are located inside a wireless coverage of  $WR_1$ . The second table  $nct_{i2}$  represents a set of next level neighbors  $nei_2$ , that are located in a wireless range of  $WR_2$ .

The table entries in each node of an ad hoc network are maintained as follows. The nodes coming under the wireless communication range  $WR_1$  and  $WR_2$  are taken and the distance between the current position and the node position is calculated using the Euclidian distance [5]. When a node is farther away from the given node  $N_i$  the Euclidian distance will be more. The nodes are arranged on the basis of distance, signifying the first entry to be closest one. The correctness of the entries in the table is ensured by periodically updating the neighbor node information present in the tables. The distance of each node from node  $N_i$  is determined by using the Euclidian distance. For cache discovery, the top level neighbors are initially checked, and if we are not able to retrieve the data, request packet is forwarded to the nodes in the range  $WR_2$ . By this the redundancies in flooding is reduced. The communication cost is also reduced by avoiding multiple transmissions.

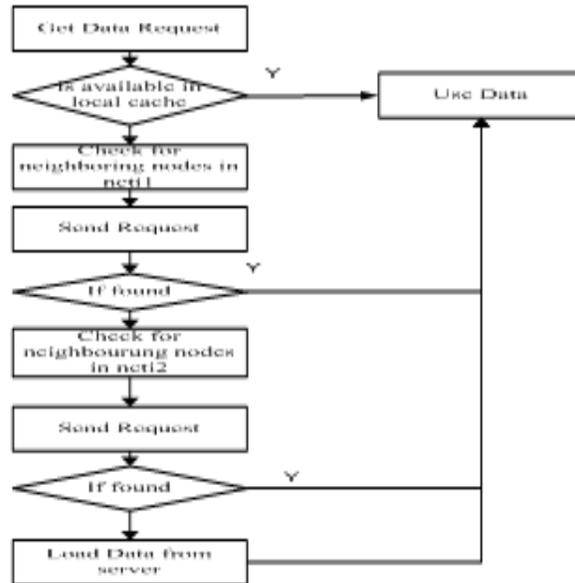


Fig.1 The cache resolution process

### 4.3. Cache Admission and Replacement

In cooperative caching nodes share the cache contents of neighboring nodes to utilize the full advantage of caching. The available cache replacement mechanisms for ad hoc network can be categorized into coordinated and uncoordinated depending on how replacement decision is made [15]. In uncoordinated scheme the replacement decision is made by individual nodes. Coordinated cache replacement considers the information present in neighboring nodes for replacement.

In order to improve the content diversity in the cooperative cache, our scheme does not cache any data coming from the neighboring nodes. This increases the availability of information for the user, as more data items are cached and also avoids additional request to the server. Previous studies [16] have shown that the requests for smaller objects are more compared to bigger objects, so the probability of achieving high hit rate is increased if we store more number of small objects. In our proposed model we set a threshold value for the size of the data item admitted to cache. The threshold value is set as 50% of the total cache size. Any object bigger than this threshold isn't brought in to cache. The cache replacement is based on a key based algorithm, which takes in to account the inter arrival time of recent references, size and consistency for page replacement.

## 5. SIMULATION SET-UP AND METHODOLOGY

We have developed a simulation model in JAVA. In our simulation, nodes are randomly placed in an area of 800X800 m<sup>2</sup>. Each node is identified by a node id and a host name. The data server is implemented as a fixed node in the simulation area. The data server contains all the data items requested by the mobile nodes. The size of each data item is uniformly distributed between  $s_{min}$  and  $s_{max}$ . The nodes in the network move randomly based on a random path. Simulation starts with each node having empty caches and, then, iteratively caches the data item in each node's cache and subsequent request for data are served from the cache. The nodes that generate data request are selected randomly and uniformly. The time interval between two consecutive

queries generated from each node/client follows an exponential distribution with mean of 10sec. Each mobile node generates a single stream of read only queries. After a query is sent out, the client does not generate new query until the pending query is served. The data access pattern follows a Zipf distribution [14] with a skewness parameter  $\alpha$  as 0.8.

### 5.1. Simulation Parameters

Table.1 Simulation Parameters

Parameter	Value
Simulation Time	3600sec
Simulation Area	800X800 m <sup>2</sup>
Database	1000 items
Cache size	20 -70 % of total database size
$S_{min}$	1
$S_{max}$	10
Nodes Density	20-70
Mobility model	Random waypoint
Transmission Range	500m
Speed of the mobile host	1-10
Pause time	200sec
Mean query generation time	10s

### 5.2. Evaluation Metrics

Our evaluation metrics include message overhead and cache hit ratio. The evaluation of these parameters is done by varying the number of nodes in each simulation run. The cache size is taken as 50% of the total size of the database. The hit ratio is defined as the percentage of requests that can be served from previously cached data. Message overhead is the overhead messages needed to manage the cache discovery process in cooperative cache.

## 6. EXPERIMENTS AND RESULTS

We compared the performance of our method with broadcasting for different node densities. Fig 2-3 shows the different performance comparison of two schemes, cooperative caching with split table approach (ccs) and cooperative caching with broadcasting (ccb), as a function of different node densities. Fig 2 shows the message overhead for both schemes under different node densities. The figure show that ccs outperforms ccb at all node densities. As the node density increases, the difference become more significant which implies that ccs can benefit from larger node density. Fig 3 depicts the comparison of cache hit ratio for different node densities. From the figure we can see that the cache hit ratio for ccs and ccb perform quite closely. The relative performance of cache hit ratio remains relatively stable for higher network densities.

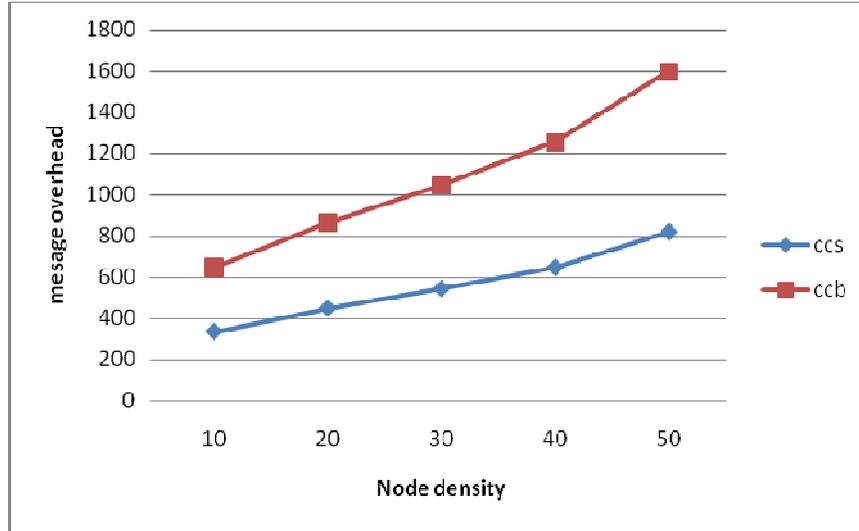


Fig. 2. Message overhead for different node densities.

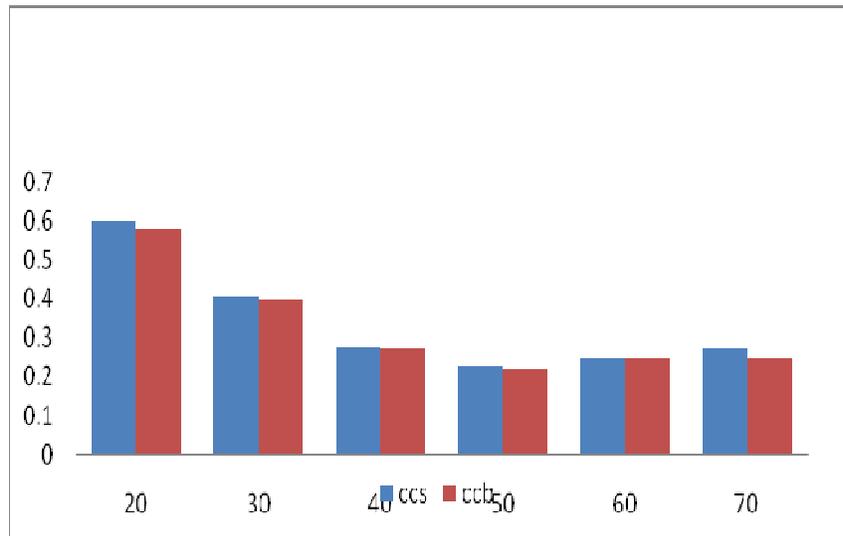


Fig. 3 Cache Hit Ratio at different node densities

## 7. CONCLUSION

In this paper we addressed the problem of cache resolution technique for peer to peer cooperative caching in ad hoc networks. The objective of our problem was to minimize the number of messages flooded in to the network, which in turn reduces the communication cost and bandwidth utilization. We designed a data discovery process based on split table approach. We evaluated the performance our algorithm with broadcasting approach through extensive simulations. Experimental results show that the proposed cache resolution algorithm can significantly reduce the message overhead when compared to broadcasting.

## REFERENCES

- [1] N Chand, RC Joshi, M Misra, Cooperative caching in mobile ad hoc networks based on data utility. *Mobile Inf Syst.* 3 (1), 19 (2007).
- [2] J Zhao, P Zhang, G Cao, CR Das, Cooperative caching in wireless P2P networks: design, implementation and evaluation. *IEEE Trans Parallel Distributed Syst.* 21(2), 229 (2010).
- [3] Joonho Cho, Seungtaek Oh, Jaemyoung Kim, Hyeong Ho Lee, Joonwon Lee, Neighbor caching in multi-hop wireless ad hoc networks, *IEEE Communications Letters*, Volume 7, Issue 11, 525 – 527(2003).
- [4] Yi-Wei Ting, Yeim-Kuan Chang. A novel cooperative caching scheme for wireless ad hoc networks: Groupcaching. In *NAS '07: Proceedings of the International Conference on Networking, Architecture, and Storage*, (2007).
- [5] S Lim, WC Lee, G Cao, CR Das, A novel caching scheme for improving Internet-based mobile ad hoc networks performance. *Ad Hoc Netw.* 4 (2), 225 (2006).
- [6] L Yin, G Cao, Supporting cooperative caching in ad hoc networks. *IEEE Trans Mobile Comput.* 5 (1), 77 (2006)
- [7] M Fiore, F Mininni, C Casetti, DF Chiasserini, To cache or not to cache? In *Proceedings of the IEEE Conference on Computer and Communications (INFOCOM 2009)*, Rio de Janeiro, Brazil 235 (2009).
- [8] B Tang, H Gupta, SR Das, Benefit-based data caching in ad hoc networks. *IEEE Trans Mobile Comput.* 7 (3), 289 (2008)
- [9] Denko, M.K., Tian, J., Cross-Layer Design for Cooperative Caching in Mobile Ad Hoc Networks, *Proc. of IEEE Consumer Communications and Networking Conf* (2008).
- [10] Y. Du, Gupta, S. K. S.: COOP-A cooperative caching service in MANETs, in *Proc. Joint Int. Conf Autonomic and Autonomous Systems and Int. Conf. Networking and Services*, 58-63 (2005).
- [11] Dan Hirsch, Sanjay Madria, A Resource- Efficient Adaptive Caching Scheme for Mobile Ad-Hoc Networks. *29th IEEE International Symposium on Reliable Distributed Systems* (2010).
- [12] González-Cañete et al : A cross layer interception and redirection cooperative caching scheme for MANETs. *EURASIP Journal on Wireless Communications and Networking* 2012.
- [13] Tiance Wang, Pan Hui, Sanjeev R. Kulkarni, Paul Cuff, Cooperative Caching based on File popularity Ranking in Delay Tolerant Networks, *Proc. Of ExtremeCom*, 2012, Zürich, Switzerland
- [14] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Sheker, Web Caching and Zipf-Like Distributions: Evidence and Implications, *Proceedings of IEEE INFOCOM*, pp.126-134, March 1999.
- [15] Preetha Theresa Joy and K. Polouse Jacob, Cache Replacement Strategies for Mobile Data Caching, *International Journal of Ad hoc, Sensor & Ubiquitous Computing*, August 2012, Volume 3, Number 4.
- [16] S. Jin, A. Bestavros, GreedyDual Web caching algorithm: exploiting the two sources of temporal locality in web request streams. *Computer Communications*, vol. 24, no. 2, Feb. 2001, pp. 174-83.