Cooperative Caching Techniques for Mobile Ad hoc Networks

Preetha Theresa Joy, K. Poulose Jacob

Dept. of Computer Science, Cochin University of Science and Technology
Kochi, Kerala, India.

Abstract—**Data caching can remarkably improve the efficiency of information access in a wireless ad hoc network by reducing the access latency and bandwidth usage. The cache placement problem minimizes total data access cost in ad hoc networks with multiple data items. The ad hoc networks are multi hop networks without a central base station and are resource constrained in terms of channel bandwidth and battery power. By data caching the communication cost can be reduced in terms of bandwidth as well as battery energy. As the network node has limited memory the problem of cache placement is a vital issue. This paper attempts to study the existing cooperative caching techniques and their suitability in mobile ad hoc networks.**

Keywords — Data Caching, MANET, Cooperative caching, cache consistency, cache placement.

I. INTRODUCTION

Improvements in wireless networks and small device processing power have increased mobile computing usage recently. New wireless network and telecommunication standards allow devices to communicate using higher speeds with longer range and more reliability. The ad hoc networks have self organizing capability and can be effectively used where other technologies either fail or cannot be deployed effectively. This situation makes extensive use of mobile computing more attractive. It also gives a number of problems like message loss, unpredictable disconnections of mobile devices, and network partitioning. However, in order to offer high quality and low cost services, several technical challenges still need to be addressed in ad hoc networks.

Most of the researches on MANETs have focused on the development of Media Access Control protocols and routing protocols to increase connectivity among mobile hosts in a constantly varying topology. Because reliable and efficient access to information is the ultimate goal of networking, the design of higher layer protocols to meet user requirements is necessary. Although routing is an important issue in ad hoc networks, other issues such as data access are also very important. High data availability is of increasing interest in mobile computing environments. In ad hoc networks, due to frequent network partition, data availability is lower than that in traditional wired networks. The two approaches used in distributed computing to enhance data availability are data replication and data caching. Data replication is the maintenance of multiple copies of data at multiple locations. The copies of data are not necessarily identical at any particular point in time. Data caching improves the data retrieval process in mobile networks. In a cached environment since the requests are served from the local cache, latency of fetching the data from the server is reduced.

The main purpose of this work is to explain the basic concepts concerning cooperative caching in ad hoc networks and to expose some of the techniques developed for cooperative caching with its benefits and drawbacks.

The remainder of the paper is structured as follows. Section II compares data replication and data caching. Section III discusses the cooperative caching in MANET Section IV presents cooperative caching design and architecture. Section V surveys some related work carried out in the field. Section VI discusses about the open research issues and section VII concludes the paper.

II. REPLICATION AND CACHING

The major factors influencing data availability in a mobile computing environment are network partition and disconnected operation. Mobile users deliberately disconnect their computers or may become unintentionally disconnected from a wireless network as they move around. In order to be able to work in these circumstances, multiple copies of heavily used data are made. With replication, data generated one time is copied or replicated to multiple nodes. This requires an

978-1-4673-2149-5/12/$31.00 ©2012 IEEE

earlier knowledge of the operational environment and is susceptible to node mobility. This makes data replication unsuitable for highly dynamic network topologies. A common requirement when data is replicated is consistency which can vary in strength between applications. Increasing the number of replicas will increase the replication cost and update cost. So the replication protocol should make a tradeoff between data availability, replication cost and update cost.

In caching-based systems the nodes cache data or the path to a data item, in order to increase data availability and to reduce query delay. Unlike data replication schemes, caching do not rely on copying data in advance. Client node always caches the received data and subsequent requests are served from the cache. If a cache miss occurs it has to get the data from the server. Here our focus is on caching protocols proposed for ad hoc networks as they are more suitable to improve data availability in a frequently changing network environment.

III. COOPERATIVE CACHING IN MANET

Cooperative caching is the sharing and coordination of cached data among several nodes to improve the data availability in wireless mobile networks. A set of distributed caches can improve their performance by cooperating in serving each other's request and making storage decisions. The criteria for selecting nodes that participate in cooperative caching may vary with different caching techniques. It may be the nodes that lie within a geographical area or in a zone or in a cluster. Cooperative caching takes advantage of local caching to facilitate data access by allowing geographically neighbouring hosts to share their cached contents. By doing so long-distance data accesses from the data center can be reduced. Here each node has to know whether there is any node in the vicinity that has cached the data it requires and where it is, if any. In cooperative caching in addition to the local cache service, the network's nodes cooperates with their neighboring nodes for serving data requests.

IV. COOPERATIVE CACHING DESIGN AND ARCHITECTURE

In a cooperative cache based data access architecture mobile nodes cache the data or the path to the data to reduce query latency and to enhance data availability. Here the data is stored in a distributed way and is shared in the entire network. A good cooperative cache management technique should have proper data discovery, cache admission control, cache replacement and cache consistency maintenance schemes. The cache admission control module decides whether a received data item is cacheable or not, and a cache discovery module is to find and deliver requested data items from the neighboring nodes. 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. 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. The cache replacement policy helps in deciding the items to be removed in the existing cache to make way for new ones. Replacement algorithms should keep track of the usage information to improve the cache hit ratio.

A. Classification of Cooperative caching Techniques

All cooperative caching techniques implicitly assumes that a node will cooperate each other in delivering the

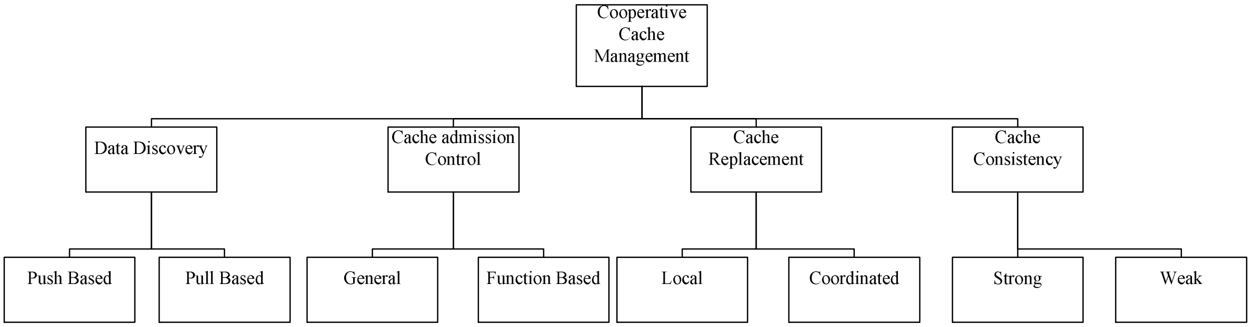

Fig . 1. Taxonomy of different cooperative cache management schemes.

desired data. Based on the system architecture ie, how the mobile nodes communicate each other for cooperative caching ,the proposed caching techniques can be divided in to decentralized group based architecture and centralized group based architecture. In decentralized group based architecture, mobile nodes are arranged in to different groups based on various parameters. Each node will belong to only group. There is no control node in this group and the clients can access data from any other node in the group. In the later case, a mobile node belongs to different groups based on various criteria. Here each group is associated with group control node to handle query and update operations.

Depending on functionality, the different modules in cooperative cache management can be further divided into different groups. The data discovery process can be either push based or pull based. In push based approach, the data requesting node can directly find the node which has the cached copy of data. In pull based approach when a local cache miss occurs, the request is broadcasted to its neighboring nodes and a node which has the data will respond. Flooding is the technique used for broadcasting.

In cache admission control, data is admitted to cache in two different ways. In the general scheme all the incoming data are cached. In function based cache admission control, a cost function will decide whether to cache the incoming data. The value function is formed by combining different parameters like distance and access frequency. The cache replacement policies were classified in to two groups, local and coordinated. In local replacement the data item to be evicted is determined independently by each node based on its local access information. In coordinated replacement policy the mobile nodes which forms cooperative cache collectively takes the replacement decision. There are two kinds of consistencies: strong and weak. A cache access mechanism can either use strong or weak consistency depending on applications. Cooperative caching techniques mainly uses weak consistency model based on time to live parameter. Fig.1 shows the classification of different cooperative caching techniques.

V. Data Caching Approaches for MANETS

To increase the data accessibility in ad hoc networks, several data caching schemes have been proposed in the literature.These schemes allocate every mobile node a certain amount of cache space. In a simple data caching scheme, a node that requests data always caches the received data and consecutive requests for the data are served from the cache until the cache expires. Here we compare some representative cooperative caching strategies for mobile ad hoc networks which combine the different aspects of the caching policies mentioned above.

In [1], three caching schemes, *Cache Data*, *Cache Path* and *Hybrid Cache*, were proposed to provide efficient data access in MANETs. In Cache Data scheme the intermediate nodes in the routing path between the source and the destination caches the data item when it finds that the data item is popular, i.e., there were many requests for the data item, or it has enough free cache space. There is no cooperative caching among the mobile hosts. Each mobile host independently performs the caching tasks such as placement and replacement. Cache Path works in a similar manner to Cache Data, but the forwarding nodes cache the data path information for future use, instead of just the data. Hybrid Cache takes advantage of the above two schemes for further improving the caching performance. Here a mobile node caches the data or the path based on some criteria. These criteria includes the data item size and the Time to Live TTL. If data item size is small, Cache Data is used since the data item needs only a small portion of the available cache; otherwise, Cache Path is used. If TTL is small, Cache Data is used because the data item might soon be invalid, using Cache Path can result in changing the path frequently. Here the cache consistency is ensured by a time to live (TTL) mechanism. A routing node considers a cached copy valid if its TTL hasn’t expired. If the TTL has expired, the node removes the cached data item.

A caching scheme for Internet based MANETs was presented in [2]. In this scheme, an A*ggregate Caching* mechanism is proposed. Here each mobile node stores data item in its local cache and these local caches are aggregated to unified large cache. The data items can be received from local caches of the mobile nodes or through an access point or a data center connected to the Internet. When a node needs a data item, it broadcasts a request to all of the adjacent nodes. The node which has the data item in its local cache will send a reply to the requester; otherwise, it will forward the request to its neighbors. Thus a request is broadcasted to the other connected nodes and eventually acknowledged by an access point or some nodes with cached copies of the requested data item. A broadcast based approach, called Simple Search (SS) algorithm, is implemented on the top of existing routing protocols, to locate the requested data items.

In [3], *COOP* addresses two basic problems in co operative caching ie, cache resolution and cache management. In cache resolution COOP tries to discover a data source which includes less communication cost by utilizing historical profiles and forwarding nodes. For this 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. In cache management, in order to minimize the caching duplication between neighboring nodes it uses iter-category and intra-category rules. Cache consistency is maintained by Time To Live (TTL) scheme.

In *Zone Cooperative caching* [4] each mobile client belongs to a zone and has a cache to store the frequently accessed data items. In this scheme each node has a local cache and these nodes share cache with other nodes that are in a zone. Zone consists of mobile hosts that are accessible in one-hop. So mobile hosts belonging to the zone of a given host forms a cooperative cache system. The data items in the cache satisfy not only the client's own requests but also the data requests passing through it from other clients. When there is a miss in the local cache, the client first searches the data in its zone. If the data is found it is retrieved otherwise the request is forwarded to the next client that lies on a path towards server. A cache replacement policy based on least utility value (LUV) has been used. For computing the LUV of a data item the access probability (Ai), size of the data item (Si), coherency which can be known by TTLi field and distance (δ) between the mobile client and data source were considered. Eq. for *utility*i function for a data item (di)

$$utili\ tyi = \frac{Ai\ .\delta i\ .TT\ Li}{si}$$

A cache admission control based on distance and a cache consistency based on time to live parameter is also considered in this scheme.

The concept of *Neighbor Caching* (NC) [5] is to deploy the cache space of inactive neighbors for caching tasks. In this method a node uses its neighbor temporarily as follows. When a node takes data from a remote node, it puts the data in its local cache for reuse. A least recently used replacement algorithm is used to remove the data from the cache. The data evicted from the cache is stored in the idle neighbor node's storage. If the node needs the data again, it will retrieve it from the near neighbor that keeps the copy of the data. This scheme exploits the available cache space of neighbor to improve the caching performance. In order to find a neighbor node with the highest possibility to keep the data for a long time a ranking based prediction algorithm is used.

In [6] a scheme called *Group Caching* (GC) in which each node maintains a localized and group caching status of its 1-hop neighbors is given. Each mobile host and its 1-hop neighbors form a group by using the "Hello" message mechanism. In order to utilize the cache space of each node in a group, the nodes periodically send their caching status in a group. By checking the cache status of the group, each mobile host knows the remaining available cache space of other mobile host in a group, the IDs and timestamps of their cached data object. Thus, the mobile host selects the proper group member for cache placement and replacement. The cache replacement is based on Least Recently Used (LRU) policy and a cache consistency based on Time to Live (TTL) attribute is used in this scheme.

In [7], a *Cluster-based Cooperative Caching (COCA)* scheme that uses a cross-layer approach to improve caching performance is proposed. COCA is a middleware which implements cross-layer optimization for cooperative caching. It consists of a stack profile, clustering, information search, cache management and prefetching modules. The stack profile module provides cross-layer information exchange, the clustering module for cluster formation and maintenance, the information search module for locating and fetching the data item requested by the client. The basic task of prefetching module is to anticipate user's request and to fetch and cache highly likely data items in advance. The cache management module includes a cache admission control, cache replacement and cache consistency. For cache admission control, a node caches all received data items until its cache space is full. When the cache is full the data items are not cached if the cluster has a copy of it. For cache consistency this scheme uses Time To Live (TTL) strategy and uses Least Recently Used Minimum (LRU-MIN), for cache replacement.

TABLE.1. FEATURES OF EXISTING DATA CACHING SCHEMES IN MOBILE AD HOC NETWORKS

Caching Scheme	Centralized/Decentralized	Cache admission Control	Data Discovery	Cache Replacement	Cache Consistency	Performance Metric Used	Advantages	Disadvantage
Cache Data /Path Hybrid cache[1]	Decentralized	Based on no: of hops	By message exchange	Based on distance and access frequency	TTL based	Average delay	Increased data accessibility	Cache path become invalid when topology changes, space needed is more.
Aggregate Cache [2]	Decentralized	Based on no :of hops	Broadcast based simple search	Based on distance and access frequency	Not Present	Throughput, Avg no of hops	Distinct data is stored increases through put	Multiple copies of data returned to server extra communication cost.
COOP [3]	Decentralized	Based on no of hops	Adaptive flooding	LRU	TTL based	Request Success ratio, Avg response delay	Limited flooding	Overhead in maintaining cache resolution.
Zone Cooperative [4]	Decentralized	Based on distance no of hops	Broadcast request to neighbors	Value Based	TTL based	Avg no of hops, Avg query latency	Simple to implement, Value based replacement	Latency is more if a neighbor does not have a cached copy.
Neighbor Caching [5]	Decentralized	Not Present	Neighbor nodes address stored in local node	LRU	Not Present	Avg Ending time, traffic	Simple and Costless, utilizes the cache space of inactive neighbors	Data thrown out without reusing lacks efficient cooperation among nodes.
Group Caching [6]	Centralized	Based on group member status	Search in local table and group table	LRU	TTL Based	Cache hit ratio, Avg latency	Power efficient protocol, redundancy is reduced	Extra cost to maintain local and group tables.
COCA [7]	Centralized	Cluster based	Hierarchical	LRU Min	TTL Based	Data accessibility ratio, Avg query delay	Cross layer design with pre fetching	Difficult to maintain when the number of nodes increases.
Adaptive Cache [8]	Decentralized	Data is replicated	Shared Common index	Based on TTL	TTL Based	Mean Query response time, Hop Count, Energy saving	Reactive caching, resource efficient	Duplication of data is high if the replicas are closer.

In [8], a R*esource Efficient Adaptive Caching* which uses an adaptive cache distribution and replication is implemented. The adaptive cache replication scheme known as Tidal Replication, considers a global demand for data for making replication decision. Here replication is done by caching the data items that consume significant portion of bandwidth for a control node towards a Dominant Request Path ie, the network path responsible for majority of request for a particular data item at the given node. When the demand wanes, the replicas are de-allocated and the burden for the data item returns to the control node. When replication request is received the available storage space for the node (RSA) is given by eqn,

$$RSA = \frac{s - m - c}{s - m}$$

Where s represents the total storage capacity of the node, m represents the total storage utilization of its non replica data items and c represents the total storage utilization of any cached replica it is holding. For resource efficiency an adaptive de allocation methodology based current utilization and age has been included. The current utilization is given by

$$U = \frac{xi}{zi}$$

Where xi is the windowed cumulative number of requests, and zi is the age of the replica, for the given data item i. The age based de- allocation threshold Tag e is represented in Eq. where yi is equal to the age threshold time in minutes, and zi is the age of the replication.

$$Tag\ e = \left\lfloor \frac{yi}{zi} \right\rfloor$$

The cache distribution scheme, *Magnetic Distribution* redistributes data items according to their demand, balancing the demand in a place of equilibrium, in order to reduce hop count and response time. The different cooperative caching schemes are summarized in Table 1.

VI. OPEN RESEARCH ISSUES

Though different techniques for cooperative caching in MANET were proposed the area is still largely open to research. In order to find the location of a cached copy of a data item in cooperative caching mechanism the protocols relies on flooding. But flooding can increase network contention and collision and this causes many problems in network communication. In group based approach used in cooperative caching, the group maintenance is hard and costly. When the number of nodes increases, it is difficult to handle the local and group tables. So the algorithms must be improved to deal with these issues. Data dissemination in a mobile computing environment is characterized by three resource constraints including energy, communication bandwidth and storage. The data caching techniques proposed deals with one or two aspects of these constraints but not all the three. Although data caching in mobile ad hoc networks has been well studied, the cache management issues like consistency and cache replacement are not adequately addressed. In the data caching techniques presented in this paper, to maintain cache consistency all of them uses a weak consistency model which uses TTL (Time-To-Live) based approach, in which a client considers a cached copy up-to-date if it's time to- live has not expired. But for some applications like application that depend upon location dependent data emphasis should be given to ensure strong consistency.

VII. CONCLUSION

In this paper we reviewed the challenges and basic concepts behind cooperative caching in MANETs and provided a summary of the work done in the area of data caching. Existing techniques were compared based on how they addressed the identified issues. It is observed that the different schemes referred above produce varying results with respect to the parameters like latency, hop count and cache hit ratio depending on the cache size, node density etc. None of the existing works give a combined and complete solution for efficient cache placement, discovery, replacement and consistency.

REFERENCES

[1]. L. Yin, G. Cao: *Supporting cooperative caching in ad hoc networks,* IEEE Transactions on Mobile Computing, 5(1):77-89(2006).

[2]. S. Lim, W. C. Lee, G. Cao, C. R. Das:*A novel caching scheme for internet based mobile ad hoc networks*. Proc .12th Int. Conf. Computer Comm. Networks (ICCCN 2003), 38-43 (Oct. 2003).

[3]. 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).

[4]. Chand, N. Joshi R.C., and Misra, M., *Efficient Cooperative Caching in Ad Hoc Networks* Communication System Software and Middleware.(2006).

[5]. 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).

[6]. 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).

[7]. 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).

[8]. Dan Hirsch, Sanjay Madria *A Resource- Efficient Adaptive Caching Scheme for Mobile Ad-Hoc Networks*.29th IEEE International Symposium on Reliable Distributed Systems.(2010).

[9].V. Srivastava and M. Motani, *"Cross-layer design: a survey and the road ahead,"* IEEE Communications Magazine, 43(12): 112-119, 2005.

[10].T. Hara. *Replica allocation in ad hoc networks with periodic data update*. Intl. Conf. on Mobile Data Management (MDM), 2002.