TECHNIQUE FOR IMPROVING THE DATA RETRIEVAL PERFORMANCE IN MOBILE ENVIRONMENTS

Mrs. K.Shanmugavadivu
Professor, Department of MCA,
K.S.Rangasamy College of Technology,
Tiruchengode -637215
E-mail: mahesh_ksv@rediffmail.com

Dr M.Madheswaran,
The Principal,
Muthayammal College of Engineering,
Rasipuram.

ABSTRACT

This article proposes a Neighbor Group caching scheme for Mobile Ad Hoc Networks (MANETs). In this scheme, Neighbor caching makes a node possible to expand its caching storage instantaneously by storing its data in the storage of idle neighbors. To reduce communication and computational overhead, we use a clustering architecture for the network organization. A weak consistency based on time to live value was used to maintain data consistency. The effects of cache size, mobility on the network performance were investigated in a discrete event simulation environment. The simulation results indicate that the proposed scheme improves both data accessibility and query delay at relatively larger cache sizes, and moderate mobility.
Keywords-Ad hoc networks, cooperative caching, data management, mobile networks, wireless networks.

1. INTRODUCTION

Recent advances in radio communication and computer technologies have led to the development of mobile computing environments. In mobile computing environments, by utilizing wireless networks, users equipped with portable computers, called *mobile hosts*, can change their locations while retaining network connections. As one of the research fields in mobile computing environments, there has been an increasing interest in *ad hoc networks* which are constructed by only mobile hosts[2], [4]. In ad hoc networks, every mobile host plays the role of a router, and communicates with each other. Even if the source and the destination mobile hosts are not in the communication range of the two mobile hosts, data packets are forwarded to the destination mobile host by relaying transmission through other mobile hosts which exist between the two mobile hosts. Multi-hop ad hoc network suffers from poor performance because the cost of wireless communications is very high due to the following reasons. First, compared with wired communication, wireless communication channel has limited capacity and weak connectivity so it is unstable and unreliable. Moreover, because several nodes share this feeble channel capacity, individual and instantaneous wireless capacity of channel can be surprisingly low**.** In addition, transport protocol like TCP decreases the efficiency of wireless networks**.** Second, distributed and decentralized computing environment in ad hoc networks increases the overheads of high level communication layer like routing and data searching. Third, because of forwarding each others' packets the capacity of multi-hop ad hoc network can rapidly fall off as the number of nodes increases [2]. One of the most important methods that reduce communication overheads is data caching. Most mobile hosts, however, do not have sufficient caching storage that the impact of caching on communication is limited. In this paper, for the purpose of overcoming this problem we present a *neighbor group caching* strategy that makes neighboring nodes possible to share each other's caching storage. Neighbor group caching is a multi-level caching strategy by which a node uses its neighbor temporarily as secondary storage. The simulation results show that our method improves the overall performance.

This article provides the following contributions to increasing the efficiency of data management in MANETs. First, we use a clustering architecture that allows Neighbor group caching mechanisms to increase data accessibility and reduce latency in the presence of host mobility. Simulation results show that the proposed schemes can significantly improve the performance in terms of the query delay and the message complexity when compared to other caching schemes

The rest of the paper is organized as follows: Section 2 reviews the related works for cooperative caching schemes in mobile ad hoc networks. The proposed Neighbor Group Caching scheme is presented in Section 3. Section 4 shows the experimental results of neighbor Group Caching compared with existing schemes. The conclusion is presented in section 5.

2. RELATED WORK

Data caching techniques used in traditional wireless networks can be extended to be used in MANETs. In this article, we investigate the use of caching for improving data accessibility and reducing latency in MANET environments. Caching has been utilized extensively in wired networks, such as the Internet, to increase the performance of Web services [4], [5], [6], [7]. However, existing cooperative caching schemes cannot be implemented directly in MANETs due to host mobility and resource constraints that characterize these networks. Consequently, new approaches have been proposed to tackle these challenges [8],[9] [10] [11]. These approaches have been introduced to increase data accessibility and reduce query delay in MANETs.

A cooperative cache based data access scheme is subsequently proposed for ad hoc networks [2], [13]. Three caching techniques, namely CacheData, CachePath, and HybridCache are utilized as caching approaches. In CacheData, the intermediate hosts, which are located along the path between the source host and the destination host, cache frequently accessed data items. In CachePath, the intermediate hosts record the routing path information of passing data. CachePath only records the data path when it is closer to the caching host than the data source. The HybridCache technique represents a combination of CacheData and CachePath. The hybridcache technique performs better than either the CachePath or CacheData approach. The cache replacement algorithm in HybridCache is based upon the access frequency of a data item and the distance to the same cached copy or to the data source.

However, due to the inherent mobility of the host, such distances can change frequently. In [10], a similar approach is proposed for data caching in a network that integrates ad hoc networks with the Internet. In [9], a replica allocation scheme with periodic data item updates is proposed. The scheme in [9] is focused on improving data accessibility with the main goal of decreasing the data access failure in response to network division. The schemes presented in [14], [12] are based on a specific routing protocol. The scheme in [14] used popularity, access cost, and coherency as criteria to replace cached data items when a mobile host's cache space is full. In [12], a transparent cache-based mechanism based on a new on-demand routing protocol called dynamic backup routes routing protocol (DBR2P) is proposed. The routing protocol and the cache mechanism allow the caching of data. In order to guarantee data access, the scheme allowed the cached data to be moved to a backup host in response to a link failure. The experimental results indicated that the scheme improved data accessibility by reducing response time in the presence of host mobility. Cooperative caching is an effective mechanism for increasing data accessibility in both wired and wireless networks. However, caching alone is not sufficient to guarantee high data accessibility and low communication latency in dynamic systems with limited network resources. In this article, we propose an Neighbor group cooperative caching mechanism for MANETs.

3. THE PROPOSED SYSTEMS ARCHITECTURE

3.1 *Network Model*

Not only *active time* to access data to other nodes but also *idle time* to do nothing except relaying other node's packets. When a node is in idle time, it can present its unused caching space to its neighbor nodes whose caching space is not enough. So an active node can use the space in idle neighbors to cache data that would have been replaced but for neighbor caching.

3.2. System Environment

The system environment is assumed to be an ad hoc network where MH access data items held as originals by other MHs. A MH that holds the original value of a data item is called data source/server/center. A data request initiated by a host is forwarded hop-by-hop along the routing path until it reaches the data source and then the data source sends back the requested data. Each MH maintains local cache in its hard disk. To reduce the bandwidth consumption and query latency, the number of hops between the data source/cache and the requester should be as small as possible. Most MHs, however, do not have sufficient cache storage and hence the caching strategy is to be devised efficiently. In this system environment, we also make the following assumptions:

- Unique identifier is assigned to each host in the system. The system has total of M hosts and MHi ($1 \leq i \leq M$) is a host identifier. Each host moves freely.
- The network is divided into several one hop non-overlapping groups where in each group a node could be in one of two roles: MHi or ordinary node. MHi is a node that maintains information of different ordinary nodes in its group.
- We assign a unique data identifier to each data item located in the system. The set of all data items is denoted by $D = \{d1, d2, . . . , dN \}$, where N is the total number of data items and dj

($1 \leq j \leq N$) is a data identifier. *Di* denotes the actual data of the item with *id di* . Size of data item *di* is *si* (in bytes).

- Each MH has a cache space of *C* bytes.
- Each data item is periodically updated at data source. After a data item is updated, its cached copy (maintained on one or more hosts) may become invalid.

3.2.1 Caching Control Message

In this caching scheme, design a caching control message to exchange the caching status in a group periodically. In experimental, every MHs exchange interval at every second. The caching control message contains the fields: {Cached data id, Timestamp, Remaining available cache space}. The caching control message is periodically sent by MHs. Each MH can maintain localized caching statues of one-hop neighbors for performing cache placement and replacement. Figure 1 illustrates the group in the view of MH D.

In NGC scheme, it proposes how and where to place the data object in a group member when an MH receives a data object from the destination. Based on the usage of caching control message, each MH knows the remaining available cache space of other MHs in a group and the IDs and timestamps of their cached data objects.

First of all, when an MH receives a data object (called receiving MH), it caches the data object if the cache space is enough. Otherwise, the receiving MH checks the available cache spaces of its group members. If the available cache space of any group member is sufficient to store the data object, the receiving MH puts the data object to the group member randomly.

Second, if the available cache space of every group member is not sufficient to cache the received object, the receiving MH lookups the group table to see if there exists a group member that already caches the data object. If yes, the data object is not cached. If no, the receiving MH selects next neighbor MHs. The receiving MH checks the available cache spaces of next 1-hop neighbor members of its neighbor member. If the available cache space of any next 1-hop neighbor member is sufficient to store the data object, the receiving MH puts the data object to the next 1 hop neighbor member.

The process of data discovery performs the searches in the caching nodes for the requested object. In Neighbor Group Caching, when a requester (source) wants to retrieve a data object from the data source, it first checks its MHs to see if the data object exists locally. If yes, it returns the data object (cache hit) to the application. If no, it lookups its group_table for the data object, if yes, the requester redirects the data request to group member, and waits the replied data object (remote cache hit). If the requester cannot find any cached record for the

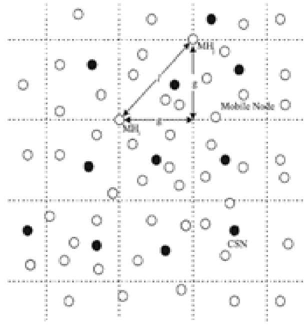

Figure 1. Partitioning of a Manet into Clusters

desired data object in the MHs and its one hop neighbor, it starts to execute the data discovery process in the next neighbor group member. Again if the requester cannot find any cached record for the desired data object in the MHs and its one hop neighbor and its one hop neighbor, it starts to execute the data discovery process. Initially, the requester constructs a routing path to the destination and sends the data request to the next neighbor MH in order to reach the data source (destination). When the intermediate nodes receive a data request in the routing path, they look up their self_table and group_table and its one hop neighbor of group member for the data request. The process of lookup first searches self_table and then searches the group_table and its neighbor group member. If the receiving MH cannot find the record the request in its selft_table and group_table and its one neighbor group member, it forwards the request to the next MH on the routing path. If the destination (data source) receives the data request, it replies the data object via the routing path. When the intermediate node receives the pass-by data object, it performs the cache placement and replacement according to their self_table and group table. There are two schemes that can deal with the cache consistency problem: weak consistency and strong consistency. Under the weak consistency, a cached data object is associated with an attribute, TimeToLive (TTL). If the TTL time expires, the cached data object is removed. Under the strong consistency, if a cached data object is requested, the caching node first asks the data source to see if the cached data object is valid or not. Because of the energy concern and the constrain of wireless bandwidth, we prefer using the weak consistency in mobile ad hoc networks. For dynamic data, a simple weak consistency [7],[13],[19] model based on the TTL mechanism is used. The DS assigns a TTL value to all dispatched data items.

Table 1: The simulated paremeters.

Simulator	Network Simulator (NS2) [9]
Simulation time	6000 seconds
Network size	1500m x 500m
Mobile host	100nodes
Transmission range of MH	100m
Mobility model	Random way point
Speed of mobile host	1~10 m/s randomly
Total of data item set	1000 data item
Average query rate	0.2 / second
Hot data	20% of total data item set
Probability of query in hot data	80%
Data size	10kBytes
Cache size	200kBytes, 400kBytes, 600kBytes, 800kBytes, 1000kBytes, 1200kBytes, 1400kBytes
Compared schemes	CacheData [1], ZoneCooperative [2], Proposed GroupCaching
Replacement policy	LRU

The TTL value of a data item is computed at the DS as follows:

$$TTL = \min\{\lambda(current - created), \rho\} \quad \text{---- (1)}$$

where λ and ρ are predefined constants. The parameters, *Current and created* refer to the current time and the creation time of the data item respectively. The parameter ρ represents a predefined threshold. To determine whether the TTL value of a data item is valid, a host computes the Current TTL (CTTL) as follows:

$$CTTL = (TTL - (current - initial)) \quad \text{----- (2)}$$

where *current* is the time when this data item was found in cache space, and *initial* is the time when this data item was dispatched from the DS. If the value of CTTL is less than or equal to zero, his data item expires. Otherwise, it is considered valid. When the TTL expires, the data is removed from the cache and the entry is marked with a flag to indicate the invalid status. This information will be sent to neighbors to avoid any request to this data later.

4. PERFORMANCE EVALUATION

4.1 The Simulation Model

The simulation is performed on NS2[16] with the CMU wireless extension. In our simulation, the AODV routing protocol [17] was tested as the underlying ad hoc routing algorithm. The simulation time is set 6000 seconds. The number of mobile hosts is set to 100 in a fixed area. We assume that the wireless bandwidth is 2MB/s and the radio range is 100m. There are totally 1000 data items distributed uniformly among all MHs. The number of hot data objects is set to 200 and all hot data objects are distributed uniformly among all MHs. The probability of queries for the hot data is set to 80%. The query rate of MHs is set to 0.2/second. In order to simulate the node join and leave operations, we set a join/leave rate. If the value of join/leave rate is 20, there will be ten MHs randomly joining and leaving the network every 20 seconds. If an MH joins or leaves the network, its content of cache will be cleared.

We model the movement of nodes in a 1500m x 500m rectangle area. The moving pattern follows the random way point mobility model [18]. Initially, nodes are placed randomly in the area. Each node selects a random destination and moves toward the destination with a speed selected ran do with randomly from (0 m/s, 10 m/s). After the node reaches its destination, it pauses for a random period of time and re*peats this movement.*

4.2 Data accessibility ratio (DAR):

The data accessibility ratio is defined as the ratio between the total number of data item requests and the total number of successfully received data items The effects of cachesize, pause time (PT) and the control parameter (δ) on data accessibility were investigated. Figure 2 shows that the data accessibility ratio increases with an increase in cache size for all caching schemes.DAR increases proportionally to pause time. This occurs because the higher pause times indicate lower mobility. Thus, the CA will be relatively static, resulting in better data accessibility.

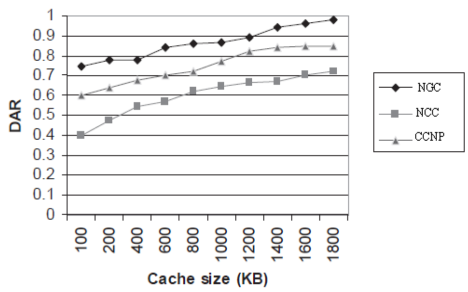

Figure 2. Data accessibility ratio cache size

4.3 Average query delay (AQD):

The average query delay is the average time interval between the generation of a query and the receipt of the query reply. Figure 4 shows that there is a decrease in average query delay associated with increased pause time for both schemes. The effects of cache size and pause time on the average query delay were investigated.

In MANETs, the number of hops is loosely related to the communication latency. Therefore, if more requests are fielded by a mobile host's own cache or by immediate neighbors, then the query delay will be much shorter than if the request is fielded by a remote DS.

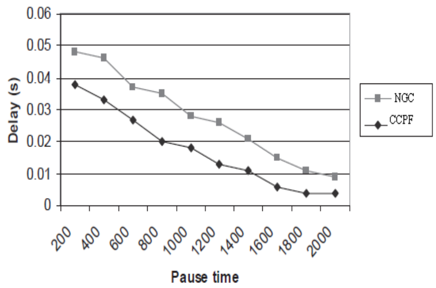

Figure 4. Average query delay as a function of pause time

5. CONCLUSION

In this article, we proposed and evaluated a cooperative caching and prefetching scheme for MANETs. The architecture for enabling cooperative data caching and a prefetching algorithm were both presented. Additionally, a cache replacement policy based on combined metrics for data access frequency and reference time was also presented. A simulation based experimental study was carried out to evaluate the performance of the proposed scheme using average data accessibility, average query delay, and network traffic overhead. The results confirm that caching coupled with prefetching increases the data accessibility ratio and reduces query delay. we intend to enhance the proposed scheme by adapting it to the integrated ad hoc network.

REFERENCES

[1] M. Frodgh, P. Johansson, and P. Larsson, "Wireless ad hoc networking—The art of networking without a network," *Ericsson Re-View*,no. 4, 2000.

[2] J. Li, C. Blake, D. Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. 7th Annu. Int. Conf. on Mobile Computing and Networking*, 2001, pp. 61–69.

[3] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE J.*

[4]. Fan, L., Cao, P., & Almeida, J. (1998).Summary cache: A scalable wide area web cache sharing protocol. In *Proceedings of ACM SIGCOMM*, ACM Press (pp. 254-265).

[5]. Rousskov, A., & Wessels, D. (1999). Cache digests. *Computer Networks and ISDN Systems*, *30*(22-23), 2155-2168.

[6]. Wang, J. (1999). A survey of Web caching schemes for the Internet. *ACM SIGCOMM, Computer Communication Review*, *25*(9), 36-46.

[7]. Wessels, D., & Claffy, K. (1998). ICP and the squid Web cache. *IEEE JSAC*,*16*(1998), 345-357.

[8]. Cao, G., Yin, L., & Das, C. R.(2004). Cooperative cache-based data access in ad hoc networks. *IEEE ComputerSociety, 37*(2), 32-39.

[9]. Hara, T. (2002). Replica allocation in ad hoc networks with periodic data update. In *Proceedings of the 3rd International Conference on Mobile Data Management* (pp. 79-86).

[10]. Lim, S., Lee, W. C., Cao, G., & Das, C.R. (2003). A novel caching scheme for internet based mobile ad hoc networks. In *Proceedings of the IEEE (ICCCN)*, (pp. 38-43).

[11] Papadopoui, M., & Schulzrinne, H. (2001). Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices.In *Proceedings of ACM MobiHoc* (pp. 117-127)

[12] Wang, Y. et al. (2005). A transparent cache based mechanism for mobile ad hoc networks. In *Proceedings of the 3rd International Conference on Inform Tech and Applications (ICITA'05)*(Vol. 2, pp. 305-310).

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

[14]. Sailhan, F., & Issarny, V. (2003). Cooperative caching in ad hoc networks. In *Proceedings of the 4th International Conference on Mobile Data Management*
(pp. 13-28).

[15] Artail, H., Safa, H., & Pierre, S. (2005). Database caching in MANETs based on separation of queries and responses. In *Proceedings of the IEEE International Conference on Wireless and MobileComputing, Networking, and Communications* (pp. 237-244).

[16]. K. Fall and K. Varadhan, "The NS2 manual", the VINT Project, http://www.isi.edu/nsnam/ns/. Apr. 2002.

[17]. C. Perkins, E. Belding-Royer, and I. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing," IETF Internet draft, draft-perkins-manet-aodvbis-00.txt, Oct. 2003.

[18]. T. Camp, J. Boleng, and V. Davies, 2002. A Survey of Mobility Models for Ad Hoc Network Research. Wireless Comm. & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2 (5): 483-502.

[19]. Alex, V. (1992). A good file system. In *Proceedings of USENIX File System Workshop* (pp. 1-12).

BIOGRAPHY

Shanmugavadivu. K received the B.Sc Degree from Bharathiar university and MCA Degrees from Bharathidasan University and M.Phil (C.S) , Manonmaniam Sundaranar University. She is working as a Professor in Department of MCA at K.S.R College of Technology, Tiruchengode. Her area of interest Computer Networks, Network Protocols, Mobile Ad hoc Networks, Unix and Network Programming. She is a member of ISTE

M.Madheswaran received the BE Degree from Madurai Kamaraj University in 1990, ME Degree from Birla Institute of Technology, Mesra, Ranchi, India in 1992, both in Electronics andCommunication Engineering. He obtained his PhD degree in Electronics Engineering from the Institute of Technology, Banaras Hindu University, Varanasi, India, in 1999. At present he is a Principal of Muthayammal Engineering College, Rasipuram, India. He has authored over forty five research publications in international and national journals and conferences. His areas of interest are theoretical modeling and simulation of high-speed semiconductor devices for integrated optoelectronics application, Bio-optics and Bio-signal Processing. He was awarded the Young Scientist Fellowship (YSF) by the State Council for Science and Technology, Tamilnadu, in 1994 and Senior Research Fellowship (SRF) by the Council of Scientific and Industrial Research (CSIR), Government of India in 1996. Also he has received YSF from SERC, Department of Science and Technology, Govt. of India. He is named in Marquis Who's Who in Science and engineering in the year 2006. He is a life member of IETE, ISTE and IE (India) and also a senior member of IEEE.

M.Madheswaran received th