Improve Data Accessibility and Reduce Query Delay in MANETs.

A.Thirumurthiraja
Asst. Prof, Department of CS,
Vel's University,Chennai – 17.
thirumurthiraja@gmail.com

Abstract— **Mobile Ad Hoc Network is a collection of mobile nodes forming an ad hoc network without the assistance of any centralized structures. There are many protocols to improve data accessibility and reduce query delay in MANETs. Some of these protocols have adopted the cooperative caching scheme, allowing multiple mobile hosts within a neighbourhood to cache and share data items in their local caches. Cross-layer optimization has not been fully exploited to further improve the performance of cooperative caching. A cluster based cooperative caching scheme is proposed and a cross-layer design approach is employed to further improve the performance of cooperative caching and prefetching schemes. The cross-layer information is maintained in a separate data structure and is shared among network protocol layers. The experimental results in the NS-2 simulation environment demonstrate that the proposed approach improves caching performance in terms of data accessibility, query delay and query distance compared to the caching scheme that does not adopt the cooperative caching strategy.**

Keywords-MANETs, Cluster, data accessibility, cooperative caching, query delay.

I. Introduction

A. Mobile Ad-hoc Networks

Mobile Ad hoc network (MANET) is a set of mobile nodes which communicate wirelessly over radio frequencies with no centralized infrastructure. This is in stark contrast to the infrastructure of other networks such as Local Area Networks, or even peer to peer networks. The properties of mobility and wireless communication present huge problems to the creation of such networks and the maintenance of services on these networks.

B. Problem in MANET

Despite the wide range of opportunities that MANETs provide, there are still research problems that need to be dealt with before it gets a vote of confidence from the public. Some of which are as follows. First, accessing remote information station via multihop communication leads to longer query latency and causes higher energy consumption. Second, when many clients frequently access the database server, they cause a high load on the server and reduce server response time. Third, if there are many nodes in multihop communication across the whole network, the network capacity will drop rapidly. Many packets will be dropped due to congestion and the limitation of packet queue length. Traffic congestion and high work load cause the server or base station to become a bottleneck. Hence in integrating MANETs with the Internet for database access, designing efficient and effective mechanisms are required.

The objective of the work is to improve the overall network performance

- To improve data accessibility
- To reduce the client query delay.
- To reduce traffic congestion around the server to improve server response time.
- To improve client capacity by adopting clustering and restricting communication within clusters.

II. Related Work

Yin.L et.al [1] has analyzed Cooperative Caching in ad hoc networks focus on routing and not much work has been done on data access. Two schemes are used that is Cache Data, which caches the data, and Cache Path, which caches the data path. After analyzing the performance of those two schemes, This scheme propose a hybrid approach (Hybrid Cache), which can further improve the performance by taking advantage of Cache Data and Cache Path while avoiding their weaknesses.

Yang.N et.al [2] suggests Ad hoc network has emerged as an important trend of future wireless system that will provide ubiquitous wireless access. The performance optimization challenges of ad hoc network and cross-layer processing to improve its performance. Here cross layer processing was implemented between physical (PHY), Medium Access Control (MAC) and network (NET) layers.

Chand.N et.al [3] proposed Caching is one of the most attractive techniques that improve data retrieval performance in wireless mobile environment. With caching, the data access delay is reduced since data access requests can be served from the local cache.

Chiang.C.C et.al [4] has proposed a cluster head token infrastructure for multihop, mobile wireless networks has been designed. In this paper, a clustered multihop routing scheme implemented for mobile wireless networks.

III. OVERVIEW OF THE PROPOSED MECHANISM

A. Overview of the proposed Mechanism

We propose a Cluster Based Cooperative Approach (CBCA) in MANETs without using any centralized infrastructure. First, the information search and cache admission control schemes are based on clustering architecture. Clustering is an effective way to organize MANETs. It reduces traffic overhead, flooding, and collisions in MANETs. It also makes the network more scalable. Second, cross-layer design is explicitly adopted to improve caching performance. A data structure called a stack profile is created, which is independent of the protocol stack and works as a data exchange buffer for different protocol layers. Stack profile supports full cross-layer design because all protocol layers can get cross-layer information via the stack profile. Finally, the prefetching technique is introduced to increase the cache hit ratio and reduce user-perceived data request response time.

IV.IMPROVING DATA ACCESSIBILITY USING CBCA

A. Cluster Based Cooperative Caching (CBCA)

The system architecture of CBCA is illustrated in figure 1 CBCA is a cluster-based middleware which stays on top of the underlying network stack and provides caching and other data management services to the upper layer applications in Manet's environment.

CBCA includes stack profile, clustering, information search, cache management, and prefetching modules. Among them, the stack profile module is the one that provides cross-layer information exchange. Through the stack profile, cached item IDs which are in the middleware layer can be retrieved by the network layer. The instances of CBCA run in each mobile host. The network traffic information which is in the data link layer can be retrieved by the middleware layer for prefetching purposes.

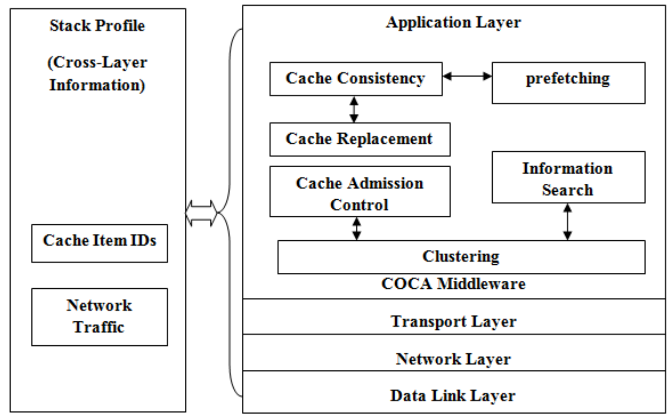

Figure 1 **.**System architecture of CBCA

B. Cross-Layer Information Exchange

Implementations of the cross-layer design follows an approach described in [5] of vertical calibration across layers enabling shared data to be accessed to by the whole protocol stack. The module stack profile is independent of the protocol stack and provides a data exchange buffer for protocol layers in the protocol stack.

In CBCA, two kinds of cross-layer information is implemented that are exchanged among layers. The first information is Network Traffic Status, provided by the data link layer. The middleware layer needs it for the prefetching operation. A mobile host initiates the prefetching operation only when the network traffic is low, and the DC will not reply to the prefetching request if the network traffic is high. The second information is Cached Item IDs. It is provided by the CBCA middleware layer. The network layer needs the information. One of the steps in the information search operation is sending a request to the DC. When the request packet is passing along the route to the DC, it will be checked by forwarding nodes. If a forwarding node has the copy of the requested data item, it drops the request packet and returns the requested data item to the requester. Because the above cross-layer information is exchanged only between two protocol layers.

C. Cluster Formation and Maintenance

Clustering is used to partition a network into several virtual groups (known as clusters) based on certain predefined criteria. Figure 2 illustrate the clustering architecture. In this paper we propose Adaptive Least cluster change algorithm (ALCC), which undergoes the following steps. First, each node learns direct contact probabilities to other nodes. Second, a node decides to join or leave a cluster based on its contact probabilities to other members of that cluster. Since our objective is to group all nodes with high pair-wise contact probabilities together, a node joins a cluster only if it's pair-wise contact probabilities to all existing members are greater than a threshold value. A node leaves the current cluster if its contact probabilities to some cluster members drop below threshold value. It is a event drive algorithm, the events are given below.

1. Slot-Timeout Event: Update Contact Probability: A Slot-Timeout event is generated by the end of every time slot, triggering the process of updating the contact probabilities by using the EWMA scheme.
2. Meet-A-Node Event: Update Cluster Information: The Meet-A-Node event is generated upon receiving the Hello message (exchanged between two meeting nodes).
3. Gateway-Outdate Event: Reset Gateway: When the Time Stamp of any entry in the gateway table is older than a threshold; a Gateway-Outdate Event is generated for that entry.

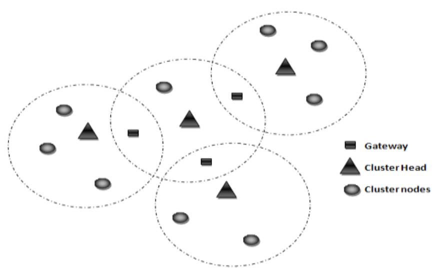

Figure 2 Illustration of clustering architecture

D. Local Cache and Neighbors

The information search operation requests data items in the following order: local cache, neighbors, and eventually the Data Center as shown in Figure 3 The solid lines represent data item requests, and dashed lines represent data item replies. When a mobile host requests a data item, it first checks its own local cache. If the requested item is found in the local cache, the request succeeds, which has the least communication overhead and the least query latency. Otherwise, the search operation will continue to the next step: sending requests to neighbors. In this step, a mobile host searches the requested data item within the neighborhood. By exchanging hello messages, each mobile host has a record of all its neighbors. First, it checks whether the DC is its neighbor. If it is, the request is sent to the DC. If it is not, there are two situations, the mobile host is a cluster head and hence it checks its ID list or it is a cluster member and it sends the request packet to its cluster head. When a cluster head receives a request packet from its cluster member, it sends the requested item to the requester if it has already cached the requested item. If the cluster head does not cache the requested item, it forwards the request packet to the DC directly or via intermediate nodes.

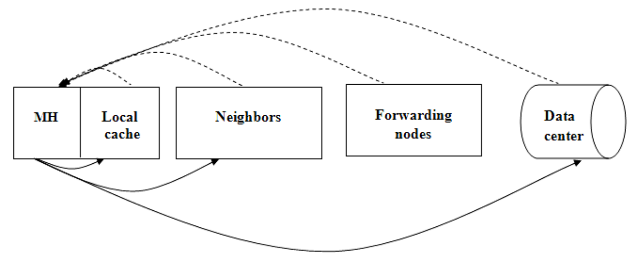

Figure 3 Information search operation

E. Information Searching at Neighbors

Upon failing to get a reply from its neighbors, the mobile host initiates the request to the DC. When the request packet is passing along the route to the DC, it will be checked by each forwarding node. Because it is not a broadcast packet and the destination is not for the forwarding nodes, the checking process is executed in the network layer. Upon finding a passing-by request packet, the routing protocol queries the requested item ID to the middleware layer via the stack profile module. If the forwarding node owns the requested item, the routing protocol drops the request packet and replies with the requested data item to the requester. The request packet will not be further forwarded to the remote DC. This method is a typical example of using cross-layer design approach in a cooperative caching system to improve performance. If no forwarding node has cached the requested item, the request packet will eventually reach the DC. When the DC receives the request packet, it replies with the requested data item to the requester.

V. Performance Evaluation

A.Simulation Model and Parameters

We use NS2 to simulate our proposed algorithm. In our simulation, 50 mobile nodes move in a 1000 meter x 1000 meter square region for 50 seconds simulation time. We assume each node moves independently with the same average speed.

All nodes have the same transmission range of 100 meters. The simulated traffic is Constant Bit Rate (CBR).

Our simulation settings and parameters are summarized in table 1

Table 1.Simulation settings and Parameters

No. of Nodes	50
Area Size	1000 X 1000
Mac	802.11
Radio Range	100m
Simulation Time	50 sec

Traffic Source	CBR
Packet Size	80
Mobility Model	Random Way Point

B. Performance Metrics

We evaluate mainly the performance according to the following metrics.

Data Accessibility Ratio: the percentage of successful requests. The number of successfully serviced requests divided by the total number of data item requests generated by all mobile hosts in the network.

Average Query Delay: the average response time for successful requests, from sending a request until receiving the response. This metric reflects the response latency of a caching system.

Average Query Distance: the average number of hops covered by successful requests. It is defined by the number of hops covered by the successful request over the number of successful requests.

The comparisons between SC (simple caching) and CCX (Cluster caching with cross layer) are used to demonstrate the performance of the cross-layer based cooperative aching scheme.

C. Results

Figure 4 shows the data accessibility ratio as a function of Queue size. The figure shows that CCX always outperforms SC at all different Queue sizes. When Queue size is greater than 100kB, there is no obvious increase in accessibility ratio for both SC and CCX.

Figure 5 shows the average query delay as a function of Queue size. The figure shows that query delay drops for both SC and CCX when Queue size increases. At all Queue sizes used in the simulation, CCX performs better than SC. When Queue size is greater than 100kB, there is no obvious decrease in query delay.

Figure 6 show the performance of SC and CCX for pause times varying between 10 seconds and 60 seconds. Figure shows the data accessibility ratio as a function of pause time. It shows that both SC and CCX have a slight increase in accessibility ratio as pause time increases.

Figure 7 shows the data accessibility ratio as a function of mean TTL time. The figure shows that when the mean TTL is small, CCX has a much higher data accessibility ratio than does SC.

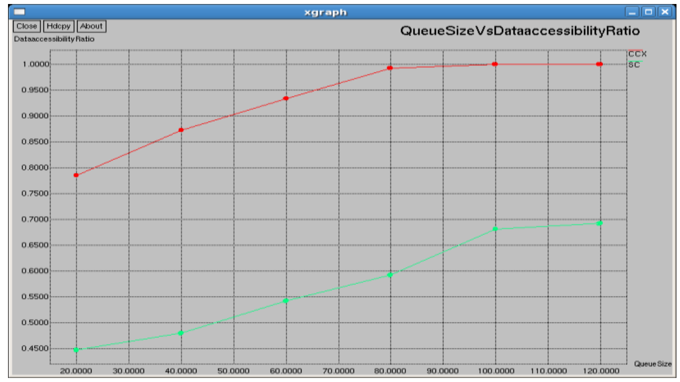

Figure 4.Queue size Vs Data Accessibility Ratio

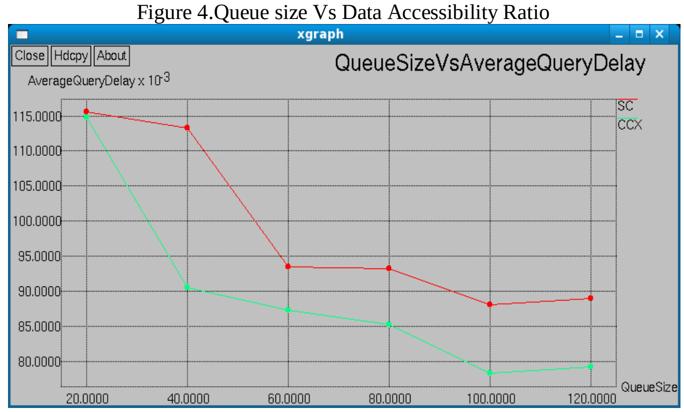

Figure 5. Queue size Vs Average Query Delay

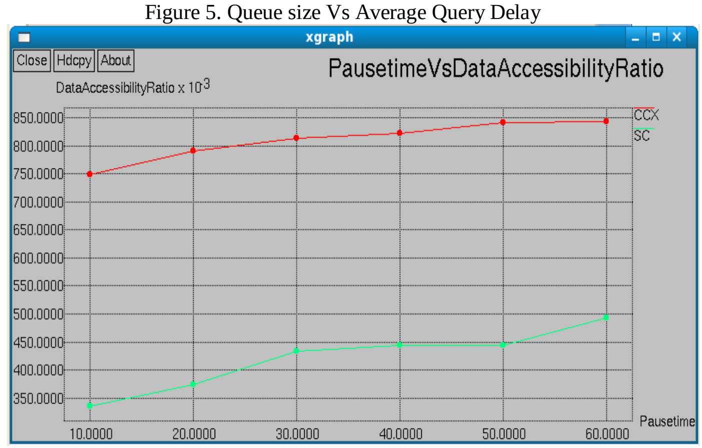

Figure 6. Pause time Vs Data Accessibility Ratio

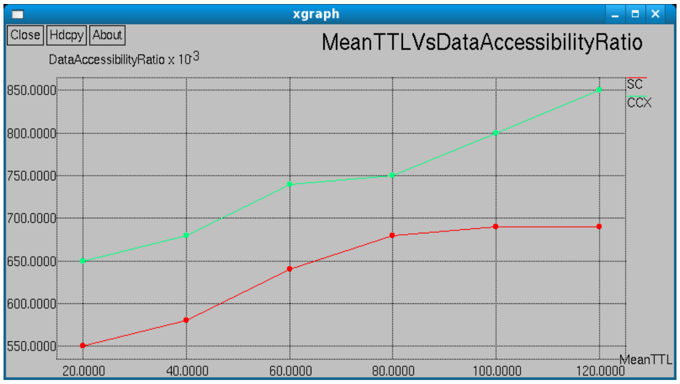

Figure 7. Mean TTL Vs Data Accessibility Ratio

VI. CONCLUSION

The main focus is on the design and evaluation of cooperative caching using cross-layer design optimization and introduction of cluster-based cooperative caching approach. To further improve data accessibility and reduce query delay, a prefetching scheme is adopted to complement the cooperative caching scheme. Cross-layer design is fully adopted in the cooperative caching approach for further improving caching performance and making the system more adaptive. The experimental results show that the cooperative caching approach which integrates and adopts cross layering is an efficient way to reduce data query delay and improve data accessibility in a MANET environment compared to the simple caching scheme.

REFERENCES

[1]. Yin.L and Cao.G, (2006) "Supporting cooperative caching in ad hoc networks,"IEEE Trans. Mobile Comput., vol. 5, no. 1, pp. 77–89.

[2]. Yang.N, Sankar.R, and Lee.J, (2005) "Improving ad hoc network performance using cross-layer information," in Proc. IEEE Int. Conf. Communications, pp. 2764–2768.

[3]. Chand.N, Joshi.R.C, and Misra.M, (2006) "Cooperative caching strategy in mobile ad hoc networks based on clusters,"Wireless Person. Commun. pp. 41–63.

[4]. Chiang.C.C, WuH.K, Liu.W, and Gerla.M, (1997) "Routing in clustered multihop, mobile wireless networks with fading channel," in Proc.IEEE SICON'97, pp. 197–211.

[5]. Su.W and Lim.T.L,(2006) "Cross-layer design and optimization for wireless sensor networks," in Proc. 7th ACIS Int. Conf. Software Engineering,Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD), pp. 278–284.

[6]. Lin.X, Shroff.N.B, and Srikant.R, (2006) "A tutorial on cross-layer optimization in wireless networks," IEEE J. Select. Areas Commun, vol.24, no. 8, pp. 1452–1463.

[7]. Gerla.M and Tsai.T, (1995) "Multiuser, mobile multimedia radio network,"J. Wireless Netw., vol. 1, pp. 255–265.

[8]. Zhao.J, Zhang.P, and Cao.G, (2008) "On cooperative caching in wireless P2P networks," in Proc. 28th Int. Conf. Distributed Computing Systems.