Adaptive Clustering Algorithm for Data Accessibility in Ad Hoc Networks

S.Gopinath[1], C.Nivedityaa[2]
[1,2]Assistant Professor, Department of ECE
Karpagam Institute of Technology, Coimbatore.
yrskgroups@yahoo.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 algorithms to improve data accessibility and reduce delay in MANETs. The basic idea is to distributively group mobile nodes with similar mobility pattern into a cluster, which can then interchangeably share their resources for overhead reduction and load balancing, aiming to achieve efficient and scalable routing in MANET. Due to lack of continues communication Distributed Adaptive Clustering Algorithm (DACA) is proposed to further improve the performance of data accessibility and reduce delay. The experimental results in the NS-2 simulation environment demonstrate that the proposed approach achieves higher delivery ratio and significantly lower overhead compared with its non-clustering counterpart.**

Keywords-MANETs, Cluster, data accessibility, routing, 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. It is a self-configuring system of mobile routers linked by wireless links which consequently combine to form an arbitrary topology. Thus, the network's wireless topology may alter rapidly and unpredictably. However, due to the lack of any fixed infrastructure, it becomes complicated to exploit the present routing techniques for network services, and this provides some huge challenges in providing the security of the communication, which is not done effortlessly as the number of demands of network security conflict with the demands of mobile networks, largely due to the nature of the mobile devices.e.g. low power consumption, low processing load.

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. Due to the lack of continuous communications, mobile nodes may have inconsistent information and therefore respond differently. As a result, it becomes challenging to acquire necessary information to form clusters and ensure their convergence and stability. Although it is largely understood by the research community that clustering helps to improve network scalability and reduce traffic congestion around the server to improve server response time and improve client capacity by adopting clustering and restricting communication within clusters.

II. RELATED WORK

Wenrui Zhao et.al [1] has analyzed the problem of efficient data delivery in sparse MANETs where network partitions can last for a significant period. MF is a mobility-assisted approach which utilizes a set of special mobile nodes called message ferries (or ferries for short) to provide communication service for nodes in the deployment area. The main idea behind the MF approach is to introduce non-randomness in the movement of nodes and exploit such non-randomness to help deliver data.

Jason LeBrun et.al [2] compares five different opportunistic forwarding schemes, which vary in their overhead, their success rate, and the amount of knowledge about neighboring nodes that they require. He proposed MoVe algorithm, which uses velocity information to make intelligent opportunistic forwarding decisions. Using auxiliary information to make forwarding decisions provides a reasonable trade-off between resource overhead and performance.

Mirco Musolesi et.al [3] proposed the Context-Aware Routing (CAR) algorithm. CAR is a novel approach to the provision of asynchronous communication in partially connected mobile ad hoc networks, based on the intelligent placement of messages. We discuss the details of the algorithm, and then present simulation results demonstrating that it is possible for nodes to exploit context information in making local decisions that lead to good delivery ratios and latencies with small overheads.

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.

D.B. Johnson et.al [5] proposed protocol in IETF (Internet Engineering Task Force). It improves connectivity among mobile hosts at the network level. These protocols are useful for applications where users equipped with mobile hosts directly communicate with each other, e.g., video conference systems. However, in ad hoc networks, there are also other applications where mobile hosts access data items held by other mobile hosts.

M.J. Carey and M. Livny et.al [6] proposed many strategies for data replications. In distributed database systems, data replication offers the benefits of shortening response time for database operations and improving data availability. In traditional systems, shortening response time is usually considered the most important issue and, thus, most replication strategies address this issue. This issue includes the propagation of update operations to replicas. The latter is achieved by replicating databases and using the replicas when the site which holds the original database fails. This approach is considered to be similar to our approach because both approaches address improving data availability (accessibility).

D. Barbara and T. Imielinski et.al [7] proposed several strategies for replicating the data. Most of these strategies assume an environment where mobile hosts access databases at sites in a fixed network and replicate/cache data on the mobile hosts because wireless communication is more expensive than wired communication.

G. Karumanchi, S. Muralidharan et.al [8] proposed a few consistency management methods based on the quorum system that has been proposed for distributed database systems.

III. OVERVIEW OF THE PROPOSED MECHANISM

A. Overview of the proposed Mechanism

We propose a Distributed Adaptive Clustering Algorithm (DACA) in MANETs without using any centralized infrastructure. First, the information search schemes are based on clustering architecture. Clustering is an effective way to organize MANETs. It reduces overhead, flooding, and collisions in MANETs. It also makes the network more scalable. Each node learns direct contact probabilities to other nodes. It is not necessary that a node stores contact information of all other nodes in network. 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.

IV. PROBABILISTIC ROUTING

Random way-point mobility model is popular to use in evaluations of mobile ad hoc protocols, real users are not likely to move around randomly, but rather move in a predictable fashion based on repeating behavioral patterns such that if a node has visited a location several times before, it is likely that it will visit that location again. The observations have been made use of these observations and this information to improve routing performance by doing probabilistic routing i.e. Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) is proposed.

A. Forwarding strategies in PROPHET

Forwarding a message is usually a simple task; the message is sent to the neighbor that has the path to the destination with usually the shortest path. Generally the message is also only sent to a single node since the reliability of paths is relatively high. In starting stage, when a message arrives at a node, there might not be a path to the destination available so the node have to buffer the message and upon each encounters with another node, the decision must be made on whether or not to transfer a particular message. In some cases it may also be sensible to forward a message to multiple nodes to increase the probability that a message is really delivered to its destination. Unfortunately, these decisions are not trivial to make. In some cases it might be sensible to select a fixed threshold and only give a message to nodes that have delivery predictability over that threshold for the destination of the message. On the other hand, when encountering a node with low delivery predictability, it is not certain that a node with a higher metric will be encountered within reasonable time. Thus, there can also be situations where we might want to be less strict in deciding who to give messages to. Furthermore, there is the problem of deciding how many nodes to give a certain message.

V.IMPROVING DATA ACCESSIBILITY USING DACA

A. Formation of Cluster

Clustering is defined as a partitioning a network into several virtual groups (known as clusters) based on certain predefined criteria.The following algorithm which contains steps are used to define cluster formation.

1. At the start we use lowest-id cluster algorithm or highest-connectivity cluster algorithm to create initial clusters.

2. When a non-clusterhead node in cluster i move into a cluster j, no clusterhead in cluster i and j will be changed (only cluster members are changed).

3. When a non-clusterhead node moves out its clusters and doesn't enter into any existing cluster, it becomes a new clusterhead, forming a new cluster.

4. When clusterhead C(i) from cluster i moves into the cluster j, it challenges the corresponding clusterhead C(j). Either C(i) or C(j) will giveup its clusterhead position according to lowest-id or highest-connectivity (or some other well de_ned priority scheme).

5. Nodes which become separated from a cluster will recompute the clustering according to lowest-id or highest-connectivity.

In a cluster, the number of hops between any two nodes is no more than two. In the whole network, there is no direct connection between cluster heads. After the formation of the clustering architecture, frequent changes of cluster heads will cause the clustering architecture to be unstable and will increase traffic overhead. In addition, the instability of clusters will degrade the performance of protocols based on the clustering architecture. The algorithm enhances by making the cluster head change as little as possible. Initially, algorithm is adopted to create clusters.
Then it operates as described in the following steps 1–4.
These steps are as follows.

1: Initially, the clusterhead gets the permission token to access the radio channel. It transmits any messages it has in its transmission queue.

2: The clusterhead passes the token to one of its neighbors according to a separately de_ned scheduling algorithm.

3: The cluster node (regular node or gateway) returns the token to its clusterhead after it has transmitted its message(s) (if any).

4: Repeat 1 to 3.

A group of nodes which moves out of a cluster will form a new cluster according to the algorithm.When a node enters a cluster, it will never challenge the status of the current cluster head in the algorithm. In contrast, in the algorithm, this node will become the cluster head if it has a lower ID than the current cluster head. Therefore, in this algorithm, the changes in cluster heads can be greatly reduced.

B. Fractional Clusters:

Because of possible errors in the estimation of contact probabilities and unpredictable sequence of the meetings among mobile nodes, many unexpected small size clusters may be formed. To deal with this problem, a merging process is employed that allows a node to join a “better” cluster, where the node has a higher stability as to be discussed in the next section. The merging process is effective to avoid fractional clusters.

C. Cluster Member:

A node with a very low nodal contact probability may still appear in the member list of another node. The main reason is that a mobile node may change its mobility pattern in real life applications. For example, a student may have his/her regular mobility pattern in a semester (i.e., visiting certain class rooms, libraries, dormitories, cafeterias, etc.). When entering a new semester, his/her mobility may change due to the new/rescheduled classes and after-school activities. Similar scenarios will happen at holidays, summer/winter breaks, or when the student changes his/her major. When mobility pattern changes, but the member list is yet updated, the problem stated above may happen. A possible solution for this problem is to use timeout for membership binding.

D. Distributed Adaptive Clustering Algorithm:

The key part of the algorithm lies on the meeting event between any pair of nodes. A node then decides its actions subsequently. Specifically, a node will join a new cluster if it is qualified to be a member. Similarly, a node leaves its current cluster if it joins a new cluster, or it is no longer qualified to be in the current cluster. When two member nodes meet, they trigger the synchronization process to update their information. To this end, we define three main functions, namely Join, Leave, and Sync for the algorithm. During initialization, Node which creates a cluster that consists of itself only and two empty tables. Its cluster ID is set to be its node ID appended with a sequence number. Each node maintains its own sequence number, which increases by one whenever the node creates a new cluster, to avoid duplication. The algorithm is event-driven. Hereafter, Node waits for three possible events, i.e., Slot, Meet and Gateway event.

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

VI. Performance Evaluation

A.Simulation Model and Parameters

We use NS2 simulator 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	100meter
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 Distributed Adaptive Clustering Algorithm (DACA) and Prophet are used to demonstrate the performance of the cluster based Mobile Ad hoc Network.

C. Results

Figure 1 shows the Delivery Ratio as a function of Queue size. The figure shows that DACA always outperforms Prophet at all different Queue sizes. When Queue size is greater than 100kB, there is no obvious increase in accessibility ratio for both Prophet and DACA.

Figure 2 shows the Number of Message Exchange as a function of Queue size. The figure shows that end to end delay drops for both DACA and Prophet when Queue size increases. At all Queue sizes used in the simulation, DACA performs better than Prophet.

Figure 3 show the performance of Prophet and DACA for Message exchanges Vs Queue Size varying like 10,20,…120. Figure shows the data accessibility ratio as a function of pause time. It shows that both Prophet and DACA have a slight increase in message exchanges while increase in Queue Size.

Figure 4 shows the Message Rate Vs Delivery Ratio. While increasing the message rate the delivery ratio of DACA is higher than the Prophet model.

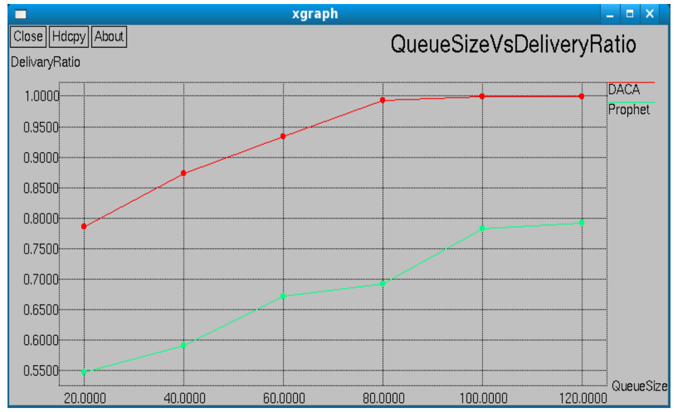

Figure 1.Queue size Vs Delivery Ratio

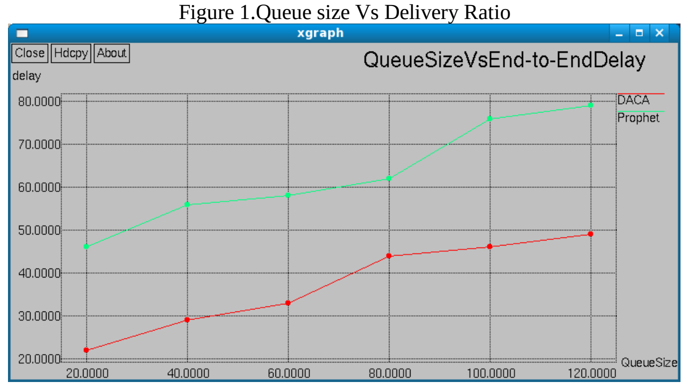

Figure 2. Queue size Vs End to End Delay

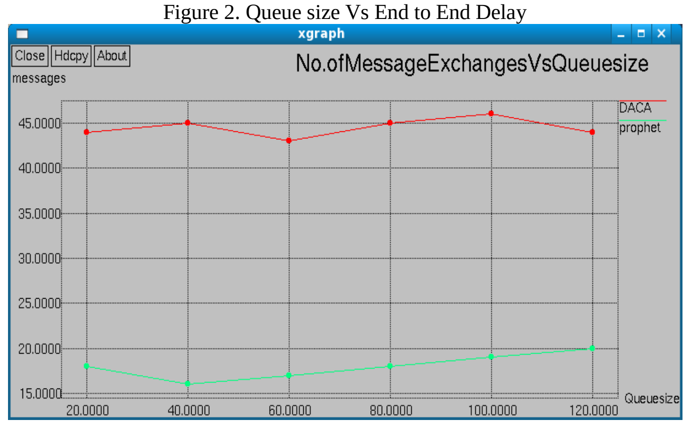

Figure 3. No of Message Exchanges Vs Queue Size

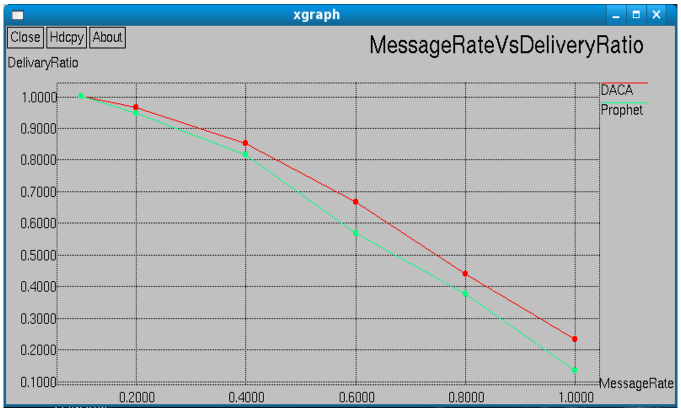

Figure 4. Message Rate Vs Delivery Ratio

VII. Conclusion

The main focus is on the design and evaluation of distributed clustering to improve the data accessbility. The basic idea is to distributively group mobile nodes with similar mobility pattern into a cluster, which can then interchangeably share their resources for overhead reduction and load balancing, aiming to achieve efficient and scalable routing in MANET. Due to lack of continues communication Distributed Adaptive Clustering Algorithm (DACA) is proposed to further improve the performance of data accessibility and reduce delay. The experimental results in the NS-2 simulation environment demonstrate that the proposed approach achieves higher delivery ratio and significantly lower overhead compared with Prophet.

References

[1]. Wenrui Zhao, Mostafa Ammar, (2004) “A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks,” MobiHoc’04,ACM 1-58113-849-0/04/0005.

[2]. Jason LeBrun, Chen-Nee Chuah, (2005) “Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks,” in Proc. 0-7803-8887-9/05/$20.00 (c)2005 IEEE

[3]. Chand.N, Joshi.R.C, and Misra.M, (2006) “Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks,”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]. D.B. Johnson, “Routing in Ad Hoc Networks of Mobile Hosts,”Proc. IEEE Workshop Mobile Computing Systems and Applications,pp. 158-163, 1994.

[6]. M.J. Carey and M. Livny, “Distributed Concurrency Control Performance: A Study of Algorithm, Distribution, and Replication,” Proc. 14th Very Large Data Bases Conf., pp. 13-25, 1988.

[7]. D. Barbara and T. Imielinski, “Sleepers and Workholics: Caching Strategies in Mobile Environments,” Proc. ACM SIGMOD ’94,pp. 1-12, 1994.

[8]. G. Karumanchi, S. Muralidharan, and R. Prakash, “Information Dissemination in Partitionable Mobile Ad Hoc Networks,” Proc.Int’l Symp. Reliable Distributed Systems (SRDS ’99), pp. 4-13,1999.