Improving Efficiency in Mobile Adhoc Networks using Dynamic Data Caching Scheme

Komal M. Sharma
Department of Computer Science and Engineering
G.H. Raisoni College of Engineering
Nagpur, India

Archana Raut
Department of Computer Science and Engineering
G.H. Raisoni College of Engineering
Nagpur, India

ABSTRACT
Mobile Ad hoc networks formed between mobile nodes are networks without the need of any fixed substructure. Mobile Ad hoc networks are temporary networks which are constituted with mobile nodes such as laptop, personal digital assistant, Tablets, mobile phones etc. In Mobile Adhoc Networks accessibility of data items is a problem. So to enhance the accessibility of data items in the network we can do caching of data items. Caching of data items is known as Cache Management. In this paper, an intelligent caching scheme called Dynamic Group Caching is used which allows grouping of mobile hosts at one hop distance. Group formed will be managed by the Group Master and the Head of the group. This Cache Management can improve the performance of MANETs.

KEYWORDS
MANETs, dynamic group caching, data accessibility.

1. INTRODUCTION
Mobile Ad hoc networks are under the category of the temporary wireless networks. Mobile nodes having heterogeneous nature form Ad hoc network between them. MANETs do not need any centralized administration because these networks are self organizing and can manage themselves accordingly. The two main characteristics of MANETs are mobility and multi hop communication. MANETs applications include military scenarios, sensor networks, rescue operations etc. Cache memory is treated as one of the possible techniques that can enhance networks performance. Caching is widely exploited in MANETs for query processing with minimum access latency[1].

There are many strategies available to use caching for the better performance of the network. Traditionally we have two caching techniques identified as Push and Pull based caching. In the first technique push caching the node that has the cache content and will itself advertise that with me there is data item, which can be used in the near future. Secondly in pull based caching the node which requires the data item will have to itself approach the nodes in the neighbourhood for the cache content. Temporal and spatial locality is the two main components in which the whole concept of data caching lies. Data items that are used again and again have the possibility to be used in the near future, so it is optimal to store that data item in the cache memory. This will reduce the unnecessary communication and data retrieval from source again and again. This will increase the data accessibility and the performance of the MANETs[2].

2. TERMINOLOGIES

2.1 Data Caching Techniques

2.1.1 Traditional caching techniques
The first approach to caching on the Web was the establishment of a client local disk and/or in memory cache in Web browsers. It was soon realized that the notion of memory hierarchy could be extended to consider Web servers as another external level of memory. Because the effectiveness of caching is dependent on the number of times the same document is requested, it was clear that the gains realized would noticeably increase if the cache were shared among several users. These facts led to the development of a second approach of caching now in use on the Web; that of caching proxies. A proxy acts as a middleware between the users' machine and the outside world. From the users' point of view, the proxy acts like a Web server: each request is sent to and answered by the proxy. From the servers' point of view the proxy acts like a client: it forwards requests to the original web server. The proxy is therefore an ideal place to a second level of cache. It is shared among several users, so there is an increased probability of data being accessed more than once. It also confers the significant benefit of being able to act as a firewall between the local machines and the outside world .[3]

2.1.2 Push and Pull Caching
It is necessary to differentiate between Push and pull caching before going any further. Pull caching is client initiated whereas the Push caching is Web Server Initiated.

The push caching was created to alleviate problems facing users of the Internet, e.g. information overload and low bandwidth.

In the push-based data delivery, the server tracks all proxies that have requested objects. If a web page has been modified, it notifies each proxy and when the client requests for the file, it is served from the proxy's cache instead of the request going directly to the server. This improves network utilization.
There are 2 kinds of notifications between the server and the proxy
1. Indicate that the object has been changed (Invalidate)
2. Send new version of the object (Update)

The kind of notification that has to be selected by the Server depends on the rate at which the

modifications occur. The Update notification can be used for frequently changing objects
whereas invalidation can be used for rest.

Pull-based data delivery is also known as demand data delivery. In pull based caching approach, a proxy explicitly requests data items from the server when the client requests for a document. The proxy instead of the request going directly to the server form the client can service subsequent requests for the same file by clients. In the pull-based approach, the proxy is entirely responsible for maintaining consistency. The proxy maintains data constancy by setting a TTL on the document cached and this copy is served until the TTL expires.

The final utility of the caching technique should be to impose fewer overheads on the network and take fewer resources, while at the same time making more data available to the users. Thus the caching system has many desirable features, which include transparency, scalability, efficiency, stability, and simplicity[4].

2.2 Data access

Data access typically refers to software and activities related to storing, retrieving, or acting on data housed in a database or other repository. Data Access is simply the authorization we have to access different data files. Data access can help distinguish the abilities of Administrators and users. E.g. Admin's may be able to remove, edit and add data while a general user may not be able as they don't have the access to that particular file.
Historically, different methods and languages were required for every repository, including each different database, file system, etc., and many of these repositories stored their content in different and incompatible formats.
In more recent days, standardized languages, methods, and formats, have been created to serve as interfaces between the often proprietary, specific languages and methods. Such standards include SQL, ODBC, JDBC, XQJ, ADO.NET, XML, X Query, X Path, and Web Services[13].

2.3 Dynamic Caching

To further improve performance for dynamic applications, new output cache gives administrators the ability to cache dynamic content (output from an ASP.NET, Classic ASP, PHP or other dynamic pages) in memory, which provides significant performance gains by removing the step of running the script used to generate the dynamic output for each request. Administrators can configure the output cached based on query string values as well as HTTP headers sent from the client to the server. The output cache, along with SSL and Windows authentication, is deeply integrated with Windows kernel mode, providing the fastest possible access[6].

3. PROPOSED SYSTEM MODEL

In this we describe a general system model of MANETs. We assume that this is a heterogeneous mobile Ad hoc network where all nodes are of different capability and capacity. All nodes will communicate with each other for the sharing of the data items. MANETs consists of a set of mobile nodes that communicate with each other using the Ad hoc communication protocols. In this proposed architecture, nodes are aggregated into sub groups (Sub Group 1, Sub Group2…..Sub Group 4) according to their capability and capacity and then grouped into a group (Group1). There will so many groups (Group1, Group2 etc) in the network that may contain several subgroups as shown in figure1.Here Group1 contains four subgroups(Sub Group1, Sub Group 2, Sub Group 3, Sub Group 4). The sub group members will communicate via Sub Group head and then to the Group Master. Group Master will maintain the information about all the Sub Group Heads in a table. Sub Group Head will maintain the information about the member nodes. The problem came in this technique is that how groups and subgroups will be created and who will be responsible for group management. There are two issues here first one is sub group creation then group creation and second is Sub Group Head selection then Group Master selection.

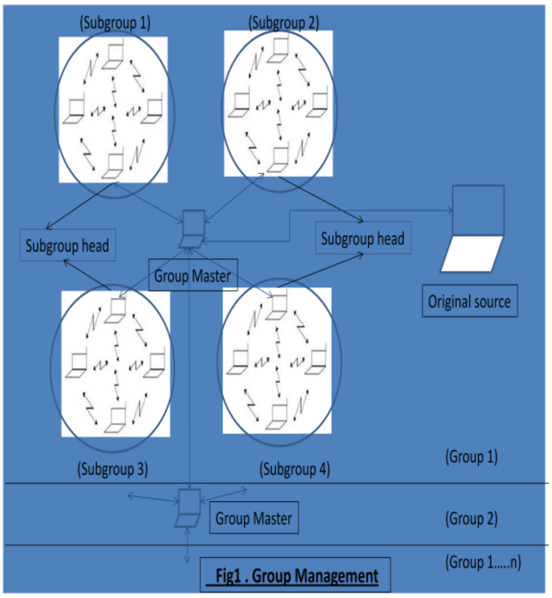

Figure 1 Group Management

3.1 GROUP MANAGEMENT

Due to mobility and resource limitations of nodes group management [10-12] is important aspect for resource saving. For efficient information retrieval, search and sharing of content at various nodes group dynamics are mandatory. We identified three main functions in group management: Sub Group Member Discovery, Member Discovery means discovering mobile nodes that are eligible to join the Sub Group according to the Group Management rules. Here we are considering that all nodes that are in a sub group must be one hop distance from each other. Sub Group Head Election, in this head is elected by the nodes of the network according to the capability and capacity of particular node with the help of limited flooding concept. Group Master Election, in this Master is elected by the Sub Group Head's with the help of some parameters associated with the Sub Group Head's. Group Dynamics Management means updating group membership according to the dynamics of the network's topology. Attributes related to the mobile node should be updated in the Group Head. In next section we will see how to elect Group Head and Group Master within mobile nodes according to the capability and capacity of mobile node.

3.1.1 Sub Group Head and Group Master Election

Initially there are several heterogeneous mobile nodes in mobile Ad hoc network. All are connected to one another or may not be in any topology. It is very difficult for a mobile node to get data from the source every time because it will increase the communication cost in terms of latency and resources. In this architecture we are designated a Head that will server all its members. To determine the Sub Group Head four parameters are considered. These are:

- Mobility
- Energy Level (Battery Life)
- Memory Size
- Processing Power

Every MN in network has mobility, energy level, memory size and processing power value. We are not bothering about how to calculate mobility, energy level etc. our aim is to consider these parameters and to calculate a value called Stability Factor (SF). Nodes flood their capacity to the neighboring nodes at one hop distance. For head election SF1 can be calculated as in (1):

$$SF1 = \frac{E(x)*S(x)*P(x)}{M(x)} \qquad (1)$$

SF1 is calculated according at each node and noted down in a table and then sorted in the decreasing order of Stability Factor. The node on the first position i.e. highest SF1 will be elected as Sub Group Head as shown in algorithm 3.1.1.1. All nodes at one hop distance will join to this node and they will not participate in the election if it message from the other node then they deny this message because they already associated with one sub group. There is one assumption that if Sub Group Head leaves then who will take responsibility of Head. Then we re-elect the Head with in the Sub Group members according to their capability and capacity.

3.1.1.1 Algorithm for Sub Group Head Election

1. flood (node ID, M, E, S, P)
2. for each node
3. calculate SF1
4. arrange in the decreasing order of stability factor
5.highest SF node will be elected as Sub Group Head

SF2 will be calculated for the selection of Group Master. To elect group Master we considered two parameters.

- Bandwidth
- Range

SF2 can be calculated as in (2) given below:

$$SF2 = B(x) + R(x) \qquad (2)$$

SF2 is calculated at each Head of sub group. The Head having largest value of B and R will have the highest SF2 value. Then this head will be the group master as discussed in algorithm 3.1.1.2.

3.1.1.2 Algorithm for Group Master Election

1. flood (Sub Group Head ID, B, R)
2. for each Sub Group Head
3. calculate SF2
4. arrange in the decreasing order of stability factor.
5. highest SF2 will be the Group Master

Using the above two algorithms 3.1.1.1 & 3.1.1.2 the Sub Group Head & the Group Master will be selected to manage the group. Whenever any data is required by any Mobile Node, it will first check that whether the data is present in the cache of SubGroup Head Node or not, if the data is available with the SubGroup Head then the SubGroup Head will directly provide the data to the Mobile Node, but if the data is not available with the SubGroup Head then it will forward the request to the Group Master then the Group Master will check that with which corresponding node the requested data is present then it will request the corresponding node to provide the required data. In this way the data management is done.

4. CONCLUSION

Due to some problems in caching techniques an intelligent caching scheme is used in which data is cached at the group head and at the group master so that efficiency can be increased in terms of accessibility, minimized query latency and less power consumption at Mobile Node. Also in that caching scheme the replication of data objects is reduced. Therefore the objective of the paper is to present an intelligent caching scheme which will provide a solution to the caching problems so that performance of the MANETs can be improved.

5. REFERENCES

[1] Hassan Artail, Member, IEEE, Haidar Safa, Member, IEEE, Khaleel Mershad, Zahy Abou-Atme, Student Member, IEEE, and Nabeel Sulieman, Student Member, IEEE, “COACS: A Cooperative and Adaptive Caching System for MANETs” IEEE transactions on mobile computing, vol. 7, no. 8, august 2008.

[2] Han Ke, Department of Computer and Information Engineering, Harbin University of Commerce Harbin, Heilongjiang Province, China, “Cooperative Caching Algorithm based on Grouping Nodes in Mobile Ad Hoc Networks ”, Proceedings of the 2010 IEEE International Conference on Information and Automation.

[3] Ihn-Han Bae, Stephan Olariu, “Design and Evaluation of a Fuzzy Cooperative Caching Scheme for MANETs” 2010 IEEE.

[4] Mohamed Hefeeda, Senior Member, IEEE, Cheng-Hsin Hsu, Member, IEEE, and Kianoosh Mokhtarian, Student Member, IEEE “Design and Evaluation of a Proxy Cache for Peer-to-Peer Traffic”, IEEE transactions on computers, vol. 60, no. 7, July 2011.

[5] Kassem Fawaz, Noor Abbani, Hassan Artail Department of Electrical and Computer Engineering American University of Beirut Beirut, Lebanon “ A Privacy-Preserving Cache Management System for MANETs”, 2012 IEEE.

[6] R. Nandhakumar , A. Saravanan, “ Improving Data Accessibility in Mobile Ad Hoc Networks”, 2010 IEEE.

[7] Therence Houngbadji, Samuel Pierre, “Distributed Data Sharing In Mobile Ad Hoc Networks”, 2010 Australasian Telecommunication Networks and Applications Conference.

[8] Mohamed R. Atassi, Sherif G. Aly, Amr El-Kadi, "Cooperative Web Caching of Dynamic Web Content", 2011 IEEE.

[9] Atul Rao, Prashant Kumar, Naveen Chauhan, "Energy Efficient Dynamic Group Caching in Mobile Ad hoc Networks for Improving Data Accessibility", 2012 IEEE.

[10] Chi.Yin. Chow, H.V.Leong and A.T.S.Chan, "Group-based Cooperative Cache Management for Mobile Clients in a Mobile environment" IEEE Proceddings of the International Conference of Parallel Processing (ICPP'04), 2004.

[11] Matthias R. Brust "Topology-based clustered candidate selection in wireless hoc and Sensor Network." 2nd International Conference on Communication System Software and Middleware and Workshops, 2007.

[12] Yi-Wei Ting and Yeim-Kuan Chang, "A Novel Cooperative Caching Scheme for Wireless Ad Hoc Networks: Group Caching" International Conference on Networking, Architecture, and Storage (NAS), 2007.

[13] MANET: Mobile Ad hoc Networks. http://www.ietf.org/html.