2011 International Conference on Information and Network Technology
IPCSIT vol.4 (2011) © (2011) IACSIT Press, Singapore

Efficient Data Replication Algorithm for Mobile Ad-Hoc Networks

Nishant Gupta
Computer Science, Acropolis Institute of Technology and Research Indore, India
Nishantgupta.cs@gmail.com
Kopal Bafna
Computer Science, Acropolis Institute of Technology and Research Indore, India
Kopalbafna.cs@gmail.com
Neeraj Gupta
Computer Science, Acropolis Institute of Technology and Research Indore, India
Neeraj.gupta.aitr@gmail.com

Abstract. A mobile ad-hoc network (MANET) is a self-configuring network of mobile routers (and associated hosts) connected by wireless links. The routers and hosts are free to move randomly and organize themselves arbitrarily. It allows mobile nodes to communicate directly without any centralized coordinator. Significant examples include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts, and military networks. Such network scenarios cannot rely on centralized and organized connectivity, and can be conceived as applications of Mobile Ad Hoc Networks. In all these application data or file sharing occurs between the nodes. As in ad hoc networks, since mobile host move freely network partition occurs frequently thus data availability in Ad-hoc network is reduced .In this paper, an efficient Data Replication technique for Mobile Ad-hoc networks is proposed that improve data availability by considering all the issues related with MANET such as power consumption, resource availability, response time and consistency management.

1. Introduction

A MANET is a self-configure, autonomous collection of mobile nodes that communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is decentralized, where all network activity including discovering the topology and delivering messages must be executed by the nodes it selves, i.e., routing functionality will be incorporated into mobile nodes. But data availability in MANET is reduced due to dynamic topology. Data Replication is technique which enhances data availability by making copies of data items. Furthermore there are various issues arise in MANET which leads to problem in data replication. Some of these are:

A. Power Consumption

One of the major issues in replicating data over MANET is battery limitation. If a node with less power is replicated with many frequently accessed data items, it soon gets drained and it cannot provide services any more.

B. Node mobility

MANET has an arbitrary topology; it is hard to locate any node in the domain. Thus data of each mobile node is not available at all the time.

C. Resource availability

Since nodes participating in MANET are portable devices, memory capacity is limited. If node has insufficient memory to replicate the all the data item is not possible.

D. Response time

It is defined as the time taken for the client to access the data from the servers. If a server is far away from the network; response time will be more to service the client.

E. Consistency Management

Another issue in data replication is to synchronize the replicas of data on all nodes. But if a replica is frequently updated, other replicas becomes invalid. Such invalid accesses consume the power of mobile hosts, who sometimes need rollbacks, which is serious problem for mobile hosts that usually have poor resources.

Our algorithm enhances data accessibility by addressing the various issues. Algorithm is composed of four main parts which are as following:

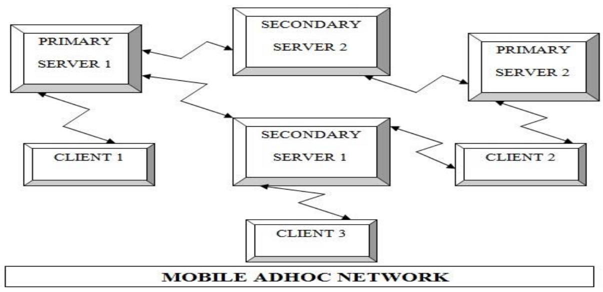
- What data is to replicate?
- Where to replicate data?
- How to access data?
- How to synchronize data?

2. Related Works

A data replication technique that replicates data items based on their access frequencies and the current network topology is proposed in [2]. Highly access data are replicated before least access data items. If the access characteristics of data items are similar, there could be replica duplications at many mobile nodes. Hence, two other techniques to reduce replica duplication between mobile nodes are proposed in [2]. They also detect network partitioning and replicate highly access data items before such a partitioning occurs to improve data accessibility. A data replication technique that replicates data items at multiple nodes and employs quorum based strategies to update and query information is proposed in [3]. It sends the update information to nodes in such a way that other nodes while querying for this update information gets the most updated information, and thereby, reducing inconsistency and dirty read transactions. A new metric for evaluating wireless link robustness that is used to detect network partitioning is proposed in [6]. Its decision to replicate data items is taken not only at the time of detecting network partitioning, but also during the time when the wireless connections become bad in terms of reliability, bandwidth and delay.

3. Architecture

As MANET does not have any fixed architecture but for developing our Algorithm we referred [4] and assume architecture as given below. Through this architecture we can only make classification between the mobile nodes.



A. **Host**: mobile hosts can be classified into two groups:

1. Clients 2. Servers

1. Server- Servers are referring to as nodes with more resources. Server are classified in two category

a. Primary copy server- Server that holds the original copy of a data item is termed its primary copy server.

b. Secondary copy servers- Other servers that hold the replicas of the data item are termed its secondary copy servers.

2. Client- Nodes that request for data or access data are clients. Clients are also referring to the nodes with fewer resources.

Each server can store only a certain number of data items, called maximum capacity

B. Data Types: They are classified into read-only and read-write data items. Read-write data items can be further classified into two types.

1. Temporal data item: Those data items the values of which are valid only for a certain period of time.

2. Persistent data items: These are valid throughout their existence in the database.

All read-write data items can be further classified into periodic and a periodic update data items.

1. Periodic update data items: Those data items that are updated periodically at fixed intervals of time.

2. Aperiodic update data items: Those data items that are updated at random intervals of time.

This paper assumes a decentralized architecture, where clients are free to communicate (single-hop or multi-hop) and submit their transactions to any of the available servers in the network as shown in Figure 1. This architecture does not place reliance on any centralized server and, thus, improves system resilience by avoiding a single point of failure.

4. What Data and on which Server it Has to Replicate?

The first step of the algorithm is to calculate the factor called as access frequencies of data items. After determining their access frequencies, data items with higher access frequencies are replicated before data items with lower access frequencies on nodes that have the maximum remaining battery power. Decision to replicate data items at appropriate servers is taken after a certain period of time called the relocation period.

Prioritize data items based on Access Frequencies

Let say, Access frequency of a data item is d at a particular node or server s, the number of times that d is accessed at s is given by Access_Frequency ds . The access frequency of each data item at each server is computed. The resulting access frequencies are thus the weighted ones to reflect replication prioritization as shown in the following sections.

1. Persistent Data Item: A data item gets higher priority than any other data item if it is accessed more frequently by a node than another data item. Node's data access time can decrease considerably if their required data items reside in the servers to which the requests were sent. The replication priority is set by assigning access frequencies to data items that are accessed by node.

2. Temporal Data Item: A temporal data item is exists only for a certain time period called its age. The probability for successfully accessing temporal data item until the next relocation period would be high, if the time remaining during which a temporal data item exists is greater than the relocation period as the temporal data item is exists throughout the entire relocation period. However, if the remaining valid time interval is less than the relocation period, then the temporal data item is valid for only some portion of the relocation period. The ratio between the remaining valid time interval and the relocation period is called the Age Relocation Ratio of a temporal data item. A data item with a higher age relocation ratio is replicated before the one with a lower age relocation ratio. Hence, the access frequency of a temporal data item is calculated based on its age relocation ratio using the following formula:

Access_Frequency ds = Access_Frequency ds* Age_Relocation_Ratio ds

Allocation of Replica to the servers and redundant data elimination:

For making the decision of allocating replica to the appropriate server, the link stability has to be considered. From [4] Disconnection time between two server can be calculated, the time in which they would be disconnected can be estimated. If the disconnection time of two servers is greater than the relocation time period, these two servers can share data reliably until the next relocation time period.

Another thing to be considered is remaining battery power .Nodes might not be able to communicate with each other if they are out of power. Hence, the link stability connecting two servers until the next relocation period is given by the formula:

Reliability Ratio = Percentage of node power remaining * (disconnection time /relocation period)

A higher reliability ratio between two servers means that the link connecting them is more stable to share data between them.

After determining the access frequencies and the reliability ratio of all servers, data items in the descending order of their access frequencies are assigned to servers until the max capacity of data items in those servers has been reached. If the access frequencies of data items are similar at many servers, the same data items would be replicated at those servers. Hence, the decision to eliminate redundancy is referred from [4] in which decision is taken only if it improves data accessibility.

a) ***One is Primary copy server and other is Secondary server of data dp*** : In this case, Let say Primary copy server is *P* and the other Secondary server is *S* .The next highest accessed data item on other server S, *ds,* is computed. If the link connecting P and S is stable so that all requests for *dp* in *S* can be successfully executed by forwarding the request to *P,* then only data item *ds* can replace *dp* in S. As discussed above, the reliability ratio of two servers indicates the stability of the links connecting them. It is beneficial to replace *dp* by *ds* in S only if the sum of the number of times that *ds* can be accessed from *S* (Access_Frequency ds S) and the number of times that *dp* can be accessed from *P* (Reliability_Ratio P S * Access_ Frequency dp reliability ratio of two servers indicates the stability of the links connecting them. It is beneficial to replace *dp* by *ds* in S only if the sum of the number of times that *ds* can be accessed from S (Access_Frequency ds S) and the number of times that *dp* can be accessed from *P* (Reliability_Ratio P S * Access_ Frequency dp S) is greater than the number of times that *dp* can be accessed from *S* (Access_Frequency dp S).
b) ***Both of the servers are secondary copy servers of data****:* when these servers, *si* and *sj*, hold only the replica and not the original copy of *dx*, the decision to remove this redundancy is based on the access frequencies of the next frequently accessed data items in both *si* and *sj*. The next frequently accessed data items, *du* and *dv*, for *si* and *sj*, respectively, are computed.

5. Battery Power Prediction

In MANET, every node is run by battery. A node gets killed (disconnected) if the battery power finishes. To predict the remaining battery power we assume that the transmit power is fixed. As in [7], energy required for each operation like receive, transmit, broadcast, discard on a packet is given by:

*E (packet) = b * (packet_size) + c*

Coefficient b denotes the packet size dependent energy consumption whereas c is a fixed cost that accounts for acquiring the channel and for MAC layer control negotiation. Each node has to maintain a table to record the remaining energy of its neighboring node. This data is used by the node to predict the remaining energy of the neighboring node.

Assume the remaining energy, of a neighbor node at time t1 and t2 are rengy1 and rengy2. The prediction of remaining energy of this node at time t is given by

*rengy = rengy2 +[(rengy2 - rengy1) / (t2 - t1)] * (t - t 2)*

Every node has to calculate the rengy by itself and send it to its neighbor and to its primary server. If a node is frequently accessed for a particular data its power drains off quickly and the server has to find an alternate node to replicate data. This lead to poor utilization of other nodes containing replicas and frequent replication creates communication overhead. To avoid this, a parameter called Access Frequency A is recorded in all the nodes which determine how many times a data is accessed. If access frequency of a node reaches a particular value say approximately 10 then it forwards the client’s request to some other node that contains the data thus saving energy.

6. Proposed Concept

Periodically, the primary server has to check the value of R at secondary servers’ .Primary server also maintain a periodic update table which has entry for its each neighbor and its corresponding Reliability ratio with that neighbor. If the reliability ratio of any secondary server is less say 40% then primary server assumes that the particular server is going to be disconnected and search for other neighbor nodes for replicating data. If any neighbor node reliability ratio is high, and if it is not been a secondary server for that primary server before, it accepts the replica. Otherwise primary server broadcast the request message to all the neighboring nodes to replicate data. They in turn forward the request to their neighbor nodes until a suitable node is found. After finding a suitable node primary server replicate data in that node and label it as one of the secondary servers and remove the old secondary server from its table. If the client wants to access the data it route a request to neighbor nodes. If any of the neighbor nodes are the servers of that data and if the access frequency F is less than 10, the data is granted to the client. If not the server, forward the request to nearby servers to give the data. If primary server itself is about to move or its remaining energy is low it chooses one of the secondary servers as primary server based on the reliability ratio. From then on that server decides when and where to place the replicas.

7. Algorithm

This algorithm must perform in steps as given below:
Let say

- **Data item** **DI (d)**
- **Server** **S**
- **Data type** **DT**
- **Access Frequency** **AF_ds**
- **Relocation Period** **RP**
- **Age Relocation Ratio** **ARR_ds**
- **Reliability Ratio** **RR**

ds: data d on server s.
dy: next highest frequency data item on server s.
Some Terminologies

- Valid Remaining Time (**VRT_ds**)– time remaining until which the temporal data item d in the server s is valid
- Absolute Validity Interval (**AVI_ds**)– the absolute validity interval of data item *d* in server *s;*
- Timestamp Current Value d (**TCV_ds**) – the time when the current value of the temporal data item *d* is obtained in server *s*

Start
1. Primary server check the RR of secondary servers
 If (RR< threshold) then
 Search for a node to replicate the data
2. Calculate the remaining battery power of nodes using above described method under the heading Battery power prediction.

Select the node with highest remaining power for replicating data
End If
3. Compute the access frequencies of data items based on their data types
If (d is a Temporal Data Item)
3.1. Compute Valid Remaining Time using the formula
VRT_ds=TCV_ds + AVI_ ds – T
Now
If (VRT_ds >= RP) then ARR_ds = 1
Else ARR_ds = VRT_ds / RP
End If
1.2 Access frequencies of temporal data items are computed based on their remaining valid time period
AF_ds = AF_ds *ARR_ds
End If
2. Store maximum capacity number of data items in descending order of their access frequencies in server s.
For each server si
 For each server sj adjacent to si
 For all data items dx that is redundant between si and sj
2.1 If (one is a primary copy server (say P) and the other a secondary copy (say S)) then
Find the next highest access frequency data item, dy, in S
 If (AF_dy_S + RR_P_S * AF_dx_S > AF_dx_S) Replace dx by dy in S
End If
2.2 Else if (Both si and sj are secondary copy servers) Find the next highest access frequency data items du and dv for si and sj,, respectively.
 Can du Replace dx (**C_du_R_dx**) – Check if the data accessibility would be improved if dx is replaced by du in si.
C_du_R_dx = AF_du_si +RR_sj_si * AF_dx_si> AF_dx_si
Can dv Replace dx (C_dv_R_dx) – Check if the data accessibility would be improved if dx is replaced by dv in sj.
C_dv_R_dx = AF_dv_sj + RR_si_sj* AF_dx_sj > AF_dx_sj
2.3 If (Not C_du_R_dx &&Not C_dv_R_dx)
The redundancy is not eliminated in either of the servers
2.4 Else If (Can_du_Replace_dx && Can_dv_Replace_dx)

```
If (AF_dv_si+ RR_sj_si *AF_dx_si> AF_dv_sj   +RR_si_sj * AF_ dx_sj)
Replace dx by du in si
Else Replace dx by dv in sj
End If
2.5  Else If (Can_du_Replace_dx) Replace dx by du in si
Else Replace dx by dv in sj
End If
        End For
    End For
End For
End
```

8. For Accessing Replicas

After replicating the data on appropriate servers, data are accessed in different ways based on the proposed data types. When there is a request for a data item to a primary copy server of data, data can be accessed directly from that server, irrespective of its data type. If it is a data write operation, the primary copy server, after updating its original copy of data, broadcasts its update timestamp, to indicate to the other secondary copy servers, that their replicas are out of synchronization. In contrast, if the requested data item is a read-only data item, it can be accessed from any server that holds it.

9. For Synchronize Replicas

Periodically, say after every relocation period, the primary copy server tries to synchronize its data item with secondary servers. The primary copy server requests for the last updated timestamps from all replicas. Based on the update timestamp of the primary copy and the update timestamps of the replicas, the primary copy server determines if there is any other server that has a more recent value of its data item. If such a server exists, the new value of the data item is synchronized with all other servers.

10.Load Balancing over the Nodes

For balancing load over a particular node the capacity of replicated data is need to be fixed in terms of both space and resources. Replicated data may not use the available capacity of node so that there is no space remains for its own use. If all the data is replicated at one node all the request have to be served from that node which cause indefinite burden on that node also increase network traffic as there are more chance for congestion. Similarly resource constraint is needed so that no further resources are available for processing its own transactions. For these a factor called as maxCapacity is set for each server. Also after a particular time period if a replica is not accessed by any of the node, it can be marked as invalidate and later replica can be drop as in [1].

11.Conclusions and Future Work

This replication technique makes data replication effective as it replicate data items on the basis of access frequency of data items, current network topology and stability of wireless links. It improves response time and maintained consistency. However the performance of our Algorithm is yet to be measured in terms of the percentage of transactions successfully executed, energy consumption of servers and clients, and the average difference in energy consumption between two servers. The balancing of replicating data is also required to be considering along with the selection of appropriate node for replicating data.

12.Acknowlegment

The author would like to express sincere thanks to Dr. Durgesh Kumar Mishra (Head of Department ,C.S.E,AITR,Indore) and all respected faculty members (C.S.E,AITR ,Indore) doing research work on MANET for their continual support and guidance to carry out this work.

13. References

[1]. T. Hara, “Replica Allocation Methods in Ad Hoc Networks with Data Update”, ACM-Kluwer Journal on Mobile

Networks and Applications, Vol. 8, No. 4, Aug. 2003, pp. 343-354.

[2]. T. Hara, Y.H.Loh, S.Nishio, "Data Replication Methods Based on the Stability of Radio Links in Ad Hoc Networks", Journal of the Information Processing Society of Japan, Vol.44, No.9, Sept. 2003, pp.2308-2319.

[3]. G. Karumanchi, S. Muralidharan, R. Prakash, "Information Dissemination in Partitionable Mobile Ad Hoc Networks", Symposium on Reliable Distributed Systems, 1999, pp. 4-13.

[4]. P. Padmannabhan, L. Gruenwald, "Managing Mobile Ad hoc Network Databases,"(Invited Paper) IEEE Int. Conference , June 2006.

[5]. M. Hauspie, D. Simplot, J. Carle, "Replication decision algorithm based on link evaluation services in MANET", LIFL Univ, May 2002.

[6]. Morteza maleki,Karthik "Lifetime prediction routing in MANET",University of southern California,IEEE Wireless Communication and Networking Conference, 2003.