Replication of Location-Dependent Data in Mobile Ad Hoc Networks

Vaidyanathan Ramany*
School of Computer Science and Information Technology
RMIT University
Melbourne, Victoria 3000
vramany@cs.rmit.edu.au

Peter Bertok
School of Computer Science and Information Technology
RMIT University
Melbourne, Victoria 3000
pbertok@cs.rmit.edu.au

ABSTRACT

Replication aims to improve accessibility, shorter response time and fault tolerance. When data is associated with geographical location in the network and valid only within a region around that location, the benefits from replication will apply only within this region. In mobile ad-hoc networks (MANETs), nodes move in and out of a region and can even leave the network completely, which leads to frequent changing of replica-holders. As mobile nodes have usually constrained processing power and memory, replica holders need to be selected carefully in such networks, to reduce communication overhead.

This paper proposes a solution for replication of location dependent data in mobile ad hoc networks. It will be shown that an improvement of 20% in hit ratio is achieved in accessing data items with only a moderate increase in total traffic generated. The scalability of the solution with regards to the increase in the number of nodes or data items in the network will also be shown to be good.

1. INTRODUCTION

Mobile ad hoc networks are wireless networks that are formed in an ad hoc manner, and each node performs the role of client and server. Routing of packets is also undertaken by each node. An inherent issue in ad hoc networks is unreliability and instability. This is because the nodes in an ad hoc network keep moving constantly and they may or may not be connected to other nodes at all instants of time. Mobile ad hoc networks find application, for example, in sensor networks and in disaster management systems, where each rescue group or rescuer can be considered a mobile node.

Location dependent data are data items that are valid at a particular position in a network. Along with the position, the data item also has a scope or range where it is valid. Replication in such scenarios would mean that the replicas would be held only within the scope of the data item and not spread throughout the network. For instance, in the case of a train network comprising of different stations and their data, like timetables, the data of a particular station is associated with the position of the station. When a mobile node wants to access the timetable in the network, the information from the nearest station is retrieved.

The next chapter reviews some existing solutions and examines their effectiveness and fallacies in the scenario of location dependent data and ad hoc networks. The system model and the detailed solution is presented in the next section. Experimental results are presented and analyzed in 4. A discussion of the results obtained, the positives and negatives of the solution proposed, and future work are presented in 5. The paper is concluded in 6.

1.1 Replication Issues in MANETs

Clustering, when the replicas are disseminated among neighbors or adjacent nodes, can make replicated data unavailable for nodes farther away from the replica holders.

Maintaining consistency among replicas is challenging in MANETs, as not all replicas may be accessible when a data item is updated. The replication system has to make its best effort to return the latest version of the data to any client.

Resource limitations impose a number of constraints. Processing power and memory capacity limitations require that management and state information held by each node should be minimum. To save processing power and battery life needed to send and receive messages, the amount of traffic generated should be kept low.

1.2 Outline of the Solution and Contribution

In our solution the Base Station selects a Primary Data Holder (PDH) from among the nodes in the vicinity of the data item, and then the PDH finds a backup node plus additional replica holders in the area. The PDH selected is the one closest to the data item. Compared to other approaches, our method offers an improved hit rate and shortened response time, with only a moderate increase in generated traffic. It also avoids clustering of replicas.

*The author's current email address is vaidyar@microsoft.com

MobiDE '08 Vancouver, B.C Canada

2. PREVIOUS WORK

Various solutions have been proposed for replication in mobile ad hoc networks. The solutions address two major issues in replication:

- Identifying nodes to hold replicas of data items and dissemination of replicas to the replica holders.
- Propagating updates to the replicas and maintaining consistency among the replicas.

The issue of dissemination of data items to replica holders has been addressed in [7], [9], [8], [11], [21], [18] and [1]. The solutions proposed consider different scenarios and propose mechanisms for those scenarios to decide which nodes will hold the replicated data items. The main focus is to reduce clustering of replicas and to increase the accessibility of data items.

The solutions proposed above in [7], [9] and [8] minimize clustering of replicas among neighbors, by not replicating in adjacent nodes, which ensures a good spread of the replicas across the network. By considering only bi-connected networks, that is, networks that are connected by two or more links, it tries to reduce the possibility of the networks getting disconnected in future. However, in a practical scenario, constraints like no updates or periodic updates might be too stringent. Updates to data items might be critical and may require being done and seen immediately in the network. The calculation of the correlation of data items in [9] is done at a single node. This results in a single point of failure. Resource utilization is also low as the other nodes will be idle waiting for the result of the computation. The amount of traffic generated in this mechanism is also high.

The solutions provided in [18] and [1] ensure good distribution of the data items without clustering of replicas. The amount of traffic generated is also minimum. However, the solution in [18] requires that a *Global Positioning System* be available to provide each node with its latest position. Since each node stores information about the position and holder of each replica when a request propagates through it, the amount of memory consumed is high. Both the solutions, [18] and [1], require a degree of stability in the network which might be stringent in the context of mobile ad hoc networks.

The issue of updating data items and maintaining consistency among replicas has been addressed in the works of Hara *et al.* in [10], Luo and Hubao in [15] and Sawai *et al.* in [17]. Hara *et al.* in [10] propose a solution considering two common scenarios - already connected networks, that is, propagating updates among the connected replicas, and recently connected networks, that is, propagating updates to a network that was previously unconnected and becomes connected due to topology changes. The solution by Sawai *et al.* in [17] uses a quorum-based approach where the nodes form read and write quorums for access and update operations respectively. A quorum is a group of replicas, which can take a decision to go ahead with an operation like a read or write, or not allow the operation. The solution considers a basic rule in replication using quorum-based systems, which is, the read quorum can be smaller than the write quorum. This is because a read can get the information from a minimum number of nodes while a write has to be updated to the maximum number of nodes. Construction of a read quorum is, hence, faster than constructing a write quorum.

Luo and Hubaux, in [15], propose a probabilistic solution to ensure that a request to a data item retrieves the most recent update to the data item. Since it is a probabilistic solution, it effectively tries to maximize the probability that the most recent update is retrieved. The solution also supports total ordering of all updates using a quorum-based approach and a *Gossip Architecture*([13]).

The solution in [10] requires that each node maintain a table of all data items and their timestamps, which might not be space efficient considering the memory constraints in mobile nodes. The amount of traffic generated is also huge, similar to the authors' previous solutions in [7], [9]and [8]. The solution in [17] requires that certain nodes called *proxies*, which do not move or exhibit ad-hoc properties, remain within a specific quorum. This assumption might be too stringent, as we cannot expect to always have nodes that are permanently present close to the data items in a network. This also introduces a form of client-server approach, which is not in keeping with the ad-hoc nature of ad-hoc networks. The probabilistic solution in [15] requires that accesses to a data item after an update retrieve the value before the update for a short period of time.

The solutions discussed so far are for scenarios where accesses to data items can come from anywhere in the network and hence replication needs to be provided throughout the network. In case of location-dependent data, however, data items are valid only within a region and replication needs to be provided only within that region.

Dunham and Kumar propose a solution where the same data item has different values based on its location [6]. The area of the network is divided into cells similar to a cellular phone network, and only those replicas within a cell or region are managed as replicas. The same data item in a different cell is considered a different data item. The solution uses the *Base Station* concept of cellular networks and a data item is assumed to be stored in the base station of each region. This introduces a client-server approach which is not in keeping with the nature of ad-hoc networks.

A solution for location-dependent replication was proposed by Tsuchida *et al* in [19]. Data items are replicated within a fixed distance or range from the position where they are generated. The solution uses *Geocasting*([3], [12]) to route access requests to the nodes near the position of the data item. The nodes in the network do not have information about any data item and accesses are generated based on the position or area the nodes are interested in. Replicas are present within a fixed distance from the position of the data item, at s-hop distances, where s is a parameter that can be configured. However, the solution uses flooding to identify the replicas, which is expensive, and a node might receive the same message from multiple other nodes, which makes it difficult to know the exact number of hops from the node which generated the message. The solution does not support updates of data items. The scenario considered is such that a data item is generated dynamically and is read-only while it remains in the network. Within the range of the data item, the number of replicas is not limited. This means that the node that generated the data does not know who the replicas are. Maintaining consistency among replicas will be a difficult task if the solution is scaled up for updates. There is a probability of the data item being lost as it is not ensured that the data item is held by at least one node near its position.

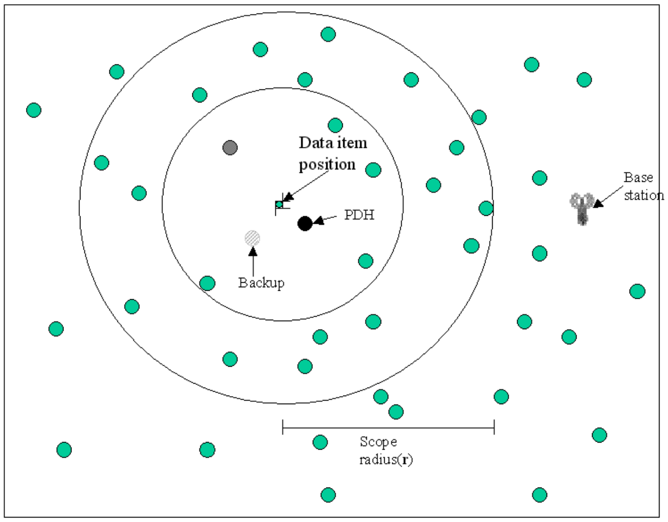

Figure 1: Basic Model with PDH, Backup and Replicas, and Showing Replica Selection

2.1 Summary

The solutions reviewed, except for [6] and [19], are for networks where the data items can be accessed from anywhere in the network and need to be replicated throughout the network. These are, hence, not suited for location-dependent data items which is the focus of this paper. The solutions discussed in [6] and [19] are for location-dependent data items, where the data items are bound to their positions. However, [6] does not consider the scenario of mobile ad-hoc networks, and [19] does not support updates to data items and the issue of the data generating node moving out of the scope of the data item is not handled.

3. SYSTEM MODEL

The proposed solution emphasizes on replica identification and maintenance, dissemination of data items to the replicas, accessing data items, updating data items and maintaining consistency among replicas. The access model is write-one-read-any, which is commonly used in mobile environments, e.g. in [14].

The following assumptions are made regarding the system environment:

1. Each data item is associated with a geographical location.

2. The data items in the network and their positions are known to the nodes.

3. The data items are initially stored in nodes called *Base Stations*. These are usually expensive to reach because they may be at the edge of the network. The delay in accessing data items from the base stations is long and they are used as the last resort to access a data item.

4. Routing of messages to their destinations is done using the Ad-hoc On-demand Distance Vector (AODV) routing mechanism([16]).

At the start, a base station tries to find a *Primary Data Holder* (*PDH*) for the data item. The PDH is a node that is closest to the position of the data item. To find the PDH, the base station broadcasts a *Primary Search* message, which contains the id of the base station and the id of the data item for which the PDH is being sought. A node on receiving the message, checks if it is within half the *Scope Radius* of the data item. This is to restrict the distance between the primary and the data item's position, at the same time taking into consideration the ad-hoc nature of the PDH.

The scope radius is the radius of the circle where the data item is valid. If a node is within half the radius it will respond with a *Primary Reply* message which contains the id of the sending node and its distance from the position of the data item. The distance is used to calculate which node is nearest to the data item's position. The base station sends a *Primary Designate* message, which includes the data item, to the node with the shortest distance to the position of the data item. The node on receiving the data item becomes the PDH and stores the data item.

The PDH looks for a backup by broadcasting the *Backup Search* message, which contains the id of the data item and the id of the PDH. All nodes which are within half the scope radius of the data item respond with a *Backup Reply* message.

The backup reply message, similar to the primary reply message, includes the distance of the node from the position of the data item. The primary will choose the node nearest to the data item as the backup. The backup is then sent a *Backup Designate* message which includes the data item. The backup and the PDH then exchange periodic heartbeat messages to make sure they are both within half the scope radius. The timeout for this message has to be short so that the likelihood of both the PDH and the backup moving out of the inner circle of half the scope radius is reduced. If the backup is lost, the PDH initiates a new search to find a backup. If the PDH is lost, the backup assumes the role of the PDH and finds a new backup. The backup is a hot backup, which means that any changes in the PDH are updated to the backup immediately.

After finding the backup, a PDH tries to find replicas. To find replicas, a PDH broadcasts a *Replica Search* message. A replica search message includes the id of the PDH and the id of the data item. A node that receives the replica search checks if it is outside half the scope radius, but within the scope radius from the data item's position. If it is, it computes the direction of the data item from itself and the distance from it to a point which is three-quarters the scope radius along an axis. This is to ensure that the replicas are near a point midway between the inner circle and the scope radius. Hence, a node to the right of the data item would compute its distance to the point at $(x_c + (3/4)r, y_c)$ where (x_c, y_c) is the position of the data item and r is the scope radius. Similarly, a node to the left would compute its distance to $(x_c - (3/4)r, y_c)$. Nodes on top of or below the data item would compute their distance to $(x_c, y_c - (3/4)r)$ and $(x_c, y_c + (3/4)r)$ respectively. The nodes then take part in an election to identify the nodes that are to be the Replica Holders based on the distance computed above. The nodes elected will send a *Replica Reply* message to the PDH. This message includes the id of the node, the id of the data item and the direction of the replica relative to the position of the data item. The PDH on receiving the replica reply messages will send out the data item to the replicas through *Replica Data* messages. If the PDH receives more than one replica reply message from a direction, only the first message is considered.

The PDH maintains communication with the replicas by sending out periodic *Replica Refresh* messages and the replicas respond with *Replica Refresh Reply* messages. If the PDH does not receive a reply for a refresh message within a specific period of time, it deletes the replica from its tables and tries to find a new replica in that direction.

3.1 Data Access

When a node wants to access a data item, it calculates its position relative to the data items and the access is made to the nearest data item only if it is within the scope of that data item. The node broadcasts an *Access Request* message. The message is sent with a low time-to-live to avoid flooding the network and includes the id of the data item being requested and the id of the node making the request.

A node, on receiving the message, checks if it is a replica holder or the PDH for the data item. If it is, it responds with an *Access Reply* message. The access reply message contains the timestamp of the data item and the data item itself. If a node which is not a replica holder or a PDH receives the message, it forwards the message after reducing the time-to-live field. If the time-to-live is zero, the message is discarded. The requesting node collects the replies and chooses the one with the highest timestamp as this represents the latest update to the data item. The access mechanism is thus $read - any$.

3.2 Data Update

Updating a data item is similar to accessing. A node checks its position relative to the data items and the update is made to the nearest data item only if it is within the scope of that data item. An *Update Request* message is broadcast which is similar to an Access Request Message. The main difference between updates and accesses is that, updates are handled only by the PDH, to maintain consistency among replicas. The update mechanism is thus $write - one$.

A replica holder on receiving the update, sets the destination of the message to the PDH and sends the message to the PDH. This is to reduce the likelihood of the message getting lost or timing out. The PDH, on receiving the update, sees if an update with the same or later timestamp was received. If so, an *Update Reply* message is sent to inform the updating node that the update failed. If no such messages have been received by the PDH, it sends out the update reply message with a different status to inform the updating node of the success of the update. A node, which is neither a PDH nor a replica holder, on receiving the update request message behaves in the same manner as receiving an access request message. The replica holders are updated by the PDH in the next replica refresh message, by including the data item in the message.

4. EXPERIMENTS

The simulation tool Omnet ([20]) was used to simulate the environment and the system model. The mobility framework for Omnet as developed in [5] was used to simulate the mobility of nodes in the Mobile Ad hoc network.

In the simulations conducted, the nodes move around in a 500m X 500m space according to the *Random Waypoint* model ([2]). AODV routing protocol ([16]) was used for routing the messages. The replication mechanism was implemented on an existing simulation of mobile ad hoc networks considering different mobility models and using the AODV routing mechanism by Concer, [4].

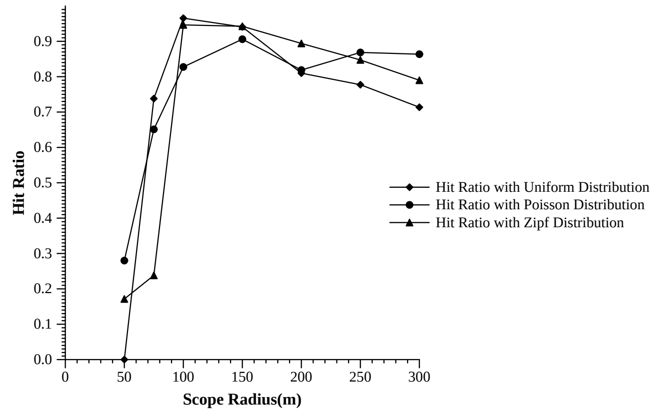

Figure 2: Graph of Scope Radius vs Hit Ratio

4.1 Data Access

As can be seen from the graph in figure 2, the hit ratio for accesses increases with the scope radius until the scope radius become 100m. This is because, at lower scope radii there might not be enough nodes to hold the data items as PDHs or replica holders. The hit ratio drops gradually after a scope radius of 150m. This is because, the distance between the replica holders and the PDHs, and the accessing nodes and replica holders or PDHs will be higher leading to increased drop probability. We can also observe that the difference in hit ratio where the scope radius is 100m and 150m is less. However, with a scope radius of 100m the nodes within the scope are pretty close by and might be able to reach the PDH in a single hop. With a scope radius of 150m, the chances of access and update requests getting lost are much higher because of the distance to the PDH from different nodes and also because of the larger number of nodes within the scope. For scopes with radii larger than 150m, the percentage of the total field area covered will be too high and the location dependency factor will be minimum. Thus, 150m is chosen as the scope radius.

Figure 3 shows how the access hit ratio varies with the number of nodes. The number of data items in the network is kept constant at three. When the number of nodes is really low, the connectivity in the network is also low as the nodes are distributed across the network and it is difficult to find replicas or PDHs to hold the data items. The hit ratio will thus be low as the accessing nodes might not be connected to any replica holder or there might not be any replica holders for the data items. With increase in the number of nodes, the connectivity in the network also increases, leading to an increase in the hit ratio. However, after a certain number of nodes, the hit ratio reaches a value after which there will be no increase. This is because, for this particular number of nodes, the connectivity from the accessing nodes to the replica holders is ensured and the presence of PDHs and replica holders is also ensured. The access misses happen only because of the ad hoc nature of the network, with nodes

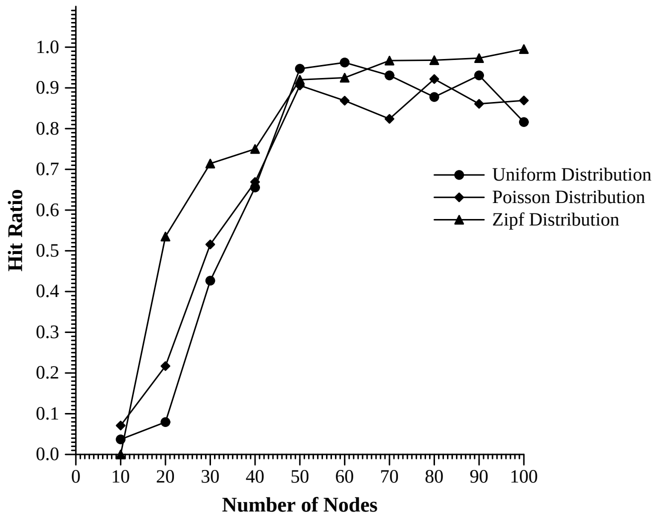

Figure 3: Graph of Number of Nodes vs Hit Ratio

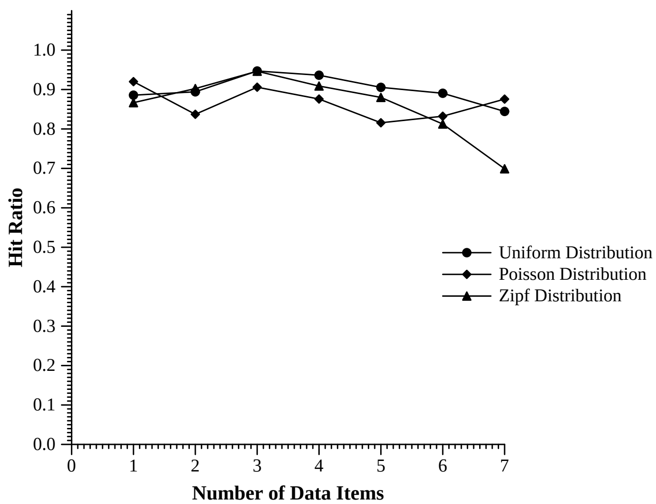

Figure 4: Graph of Number of Data Items vs Hit Ratio

moving to different positions and, the routing table entries timing out or becoming invalid, and not because of lack of connectivity or lack of replica holders. After this state is reached, increase in the number of nodes does not lead to any increase in the connectivity or, the number of replicas or PDHs, and the access hit ratio saturates. In the figure 3 the saturation can be observed at 50 nodes and hence, the number of nodes in the network is maintained at 50 for the experiments.

Since each access is independent of other accesses and also other data items in the network, the hit ratio does not depend on the number of data items in the network. It depends only on the node's current position and its nearest data item. As can be seen from the figure 4, the hit ratio does not vary much with increase in data items and remains between 0.85 and 0.95. With increase in the number of data items, the hit ratio with Zipf distributed nodes comes down as the nodes tend to be more frequent near particular data items. As more data items are added at different locations,

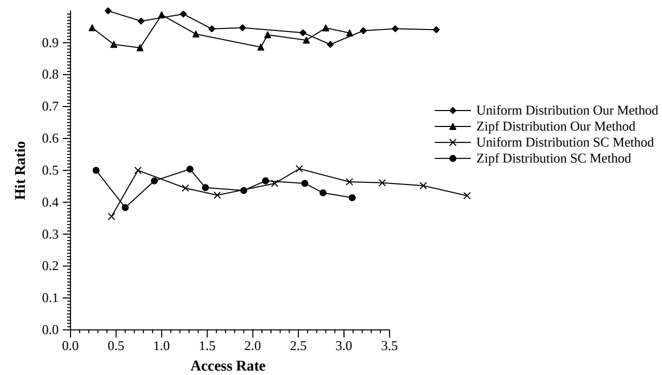

Figure 5: Graph Comparing Access Rate vs Hit Ratio of Our Method with that of the Skip-Copy Method

the number of nodes being found near those data items will be lower. The chances of finding replica holders or PDHs for those data items becomes lower. With uniform distribution of nodes, however, regardless of the number of data items, the number of nodes near that should be more or less the same. The access hit ratio in this case should not be affected. As can be observed from the graph, the hit ratio with uniform distribution stays between 0.9 and 0.95. The number of data items for the experiments can thus be any value without affecting the results of the experiments. The number of data items for the experiments is chosen to be three because, with three data items and a scope radius of 150, most of the nodes can generate access/update requests as they are within the scope of some data item or the other.

The results observed from the proposed solution were compared to those obtained using the Skip-Copy method([19]). Since in the skip-copy method accesses can originate from anywhere in the network, only those accesses originating from within the scope of a data item were considered to make it comparable to our method. A skip value of 2 was used, which means every alternate node from the node generating the data item stores the data item as a replica holder.

Hit Ratio : The access hit ratio does not depend on the number of accesses because, each access is independent of other accesses. As seen from figure 5, the access rate in our mechanism remains more or less constant between 0.85 and 0.95 regardless of the distribution used. The access misses happen due to the ad hoc nature of the network as nodes move after accesses, and routing entries in routing tables expire or get recalculated. The hit ratio is similar when using Zipf or uniform distribution and remains at around 0.9. Our method provides an improvement of close to 100% on the Skip-copy method as can be seen in the figure 5. While the skip-copy method had an average hit ratio of 0.45 for both uniform and Zipf distributions, our method improves the hit ratio up to 0.95 for these distributions.

Response Time : Since the number of hops to reach the replicas or the PDHs in our method is low, the response time is also expected to be low. The response time is measured as the time to receive the first response from a replica or a PDH for an access request. As can be seen from figure 6, the response time stays low at an average of 0.055 seconds for Uniform distribution of nodes and data items. For Zipf, the nodes are clustered around the data items. This means that

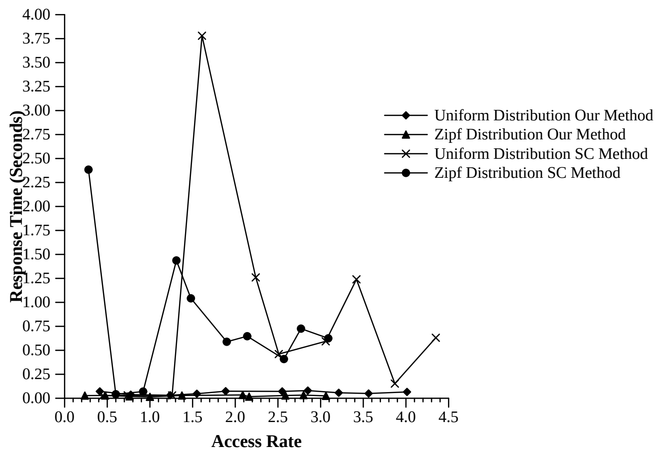

Figure 6: Graph Comparing Access Rate vs Response Time of Our Method with that of the Skip-Copy Method

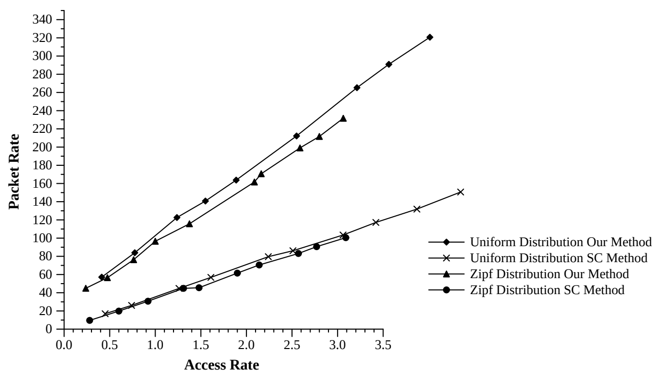

Figure 7: Graph Comparing Access Rate vs Packet Rate of Our Method with that of the Skip-Copy Method

the replicas or the PDHs will usually be reachable within a hop or a couple of hops. The response time is thus very low for Zipf distributed nodes and can be seen to be around 0.026 seconds. As can be seen from the graph, the average response time for the skip-copy method is around 0.27 seconds and 0.32 seconds for uniformly and Zipf distributed nodes respectively, while it is around 0.055 and 0.026 seconds respectively for our method. However, at certain times the skip-copy method also manages to achieve a response time of close to 0.03 seconds, but not consistently.

Network Traffic : Increase in the access rate leads to an increase in the number of packets transmitted to access the packets. Thus, the increase in the traffic with increase in access rate should be linear. As can be seen from figure 7, the increase in packet rate is linear with increase in the access rate. This trend can be observed regardless of the distribution of the nodes in the network. The increase in the number of packets transmitted per node with increase in access rate is linear for the skip-copy method also. However, the rate of increase for the skip copy method is only 50% of our method. This is because the skip-copy method does not involve any exchange of control packets to search and maintain replicas which is done in our method. The im-

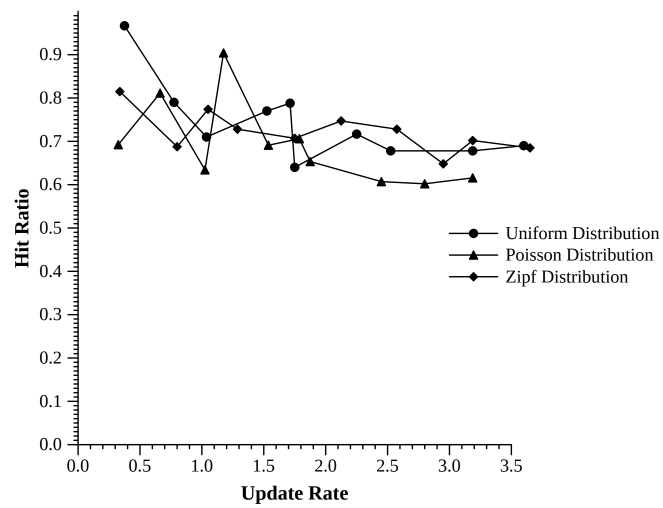

Figure 8: Graph of Update Rate vs Update Hit Ratio

provement in response time and hit ratio over the skip-copy method comes at the price of increased network traffic.

4.2 Data Update

The update rate will be lower than the access rate because of the write-one read-any mechanism of replication. Only the PDH can service updates while replica holders can also service accesses. The updates are propagated to the replicas during the next refresh. If the refresh message does not contain the data item, it means the data item has not been updated since the last refresh. The number of nodes is maintained at 50 and the number of data items at three.

Hit Ratio : The hit ratio for updates will be much lower than that for accesses because of the read-any write-one approach. Hence, the difference between the hit ratios for accesses and updates will depend on how much of the accesses are actually serviced by the replicas. As can be seen from figure 8 the hit ratio starts at a higher value and drops to an average of 0.7 for higher update rates. Updates can fail due to two reasons - Update requests not reaching the PDH or if a later update is received earlier by the PDH. We can observe that the hit ratio does not depend on the update rate and remains constant with increase in update rate. However, with an increase in the update rate there is a slight decrease in the hit ratio for updates. This is because, at lower update rates the number of simultaneous updates for the same data item will be much lower and also the number of packets in the network will be much lower. With increase in the update rate, the number of packets transmitted increases leading to a higher drop probability and the number of simultaneous updates also increases leading to failures due to earlier updates being received later.

Network Traffic : Since the packets transmitted for updates are similar to those for accesses, increase in the update rate will lead to a linear increase in the number of packets transmitted per node. When the access rate equals the update rate, the number of packets transmitted in both the cases should be the same. As can be seen from figure 9, the increase in the number of packets transmitted per node is linear with the update rate.

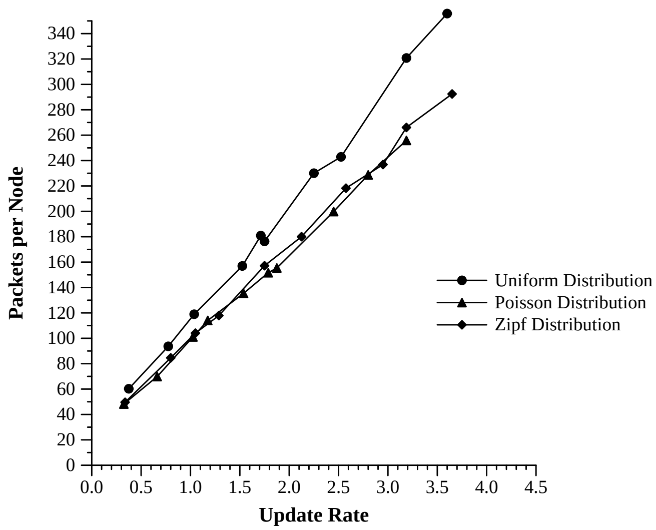

Figure 9: Graph of Update Rate vs Packet Rate

5. DISCUSSION AND FUTURE WORK

Replication is an important issue in mobile ad hoc networks because of the inherent unreliability of the network. Location dependent data in mobile ad hoc networks are those that are bound to a position in the network. The solution proposed to solve the issue of replication in mobile ad hoc networks with location dependent data. It addresses the issues of:

- Dissemination of data to the replica holders from the PDH
- Retrieval of data from the replica holders
- Updating of the data by the nodes
- Minimizing the amount of traffic generated in addressing these issues

From the experiments conducted and discussed in the previous section, an access hit ratio of around 94% was observed with Zipf and uniform distributions, and 87% with Poisson distribution. This is an improvement of around 21% observed over not replicating the data items, with uniformly and Poisson distributed nodes and 11% with Zipf distributed nodes. The solution achieves this improvement with minimum increase in traffic generated from the scenario of not using replication. Thus, the objectives of data retrieval and dissemination have been addressed and under the constraint of minimum traffic generation.

Since the increase in traffic generated is linear with the increase in the access rate, or the number of nodes in the network, or the number of data items, or the update rate, the scalability of the system for larger networks and for peak loads is also good.

The response time for accessing data items is very low and is 0.023 seconds for Zipf distributed nodes and 0.055 seconds on average for Poisson and uniform distributions. The improvement in response time with replication in case of Zipf distributed nodes is around 100% while it is smaller for Poisson and uniform distributions. Thus, in case of Zipf distributed nodes, the improvement is seen in the response time while for uniform and Poisson distributed nodes, the improvement is in the hit ratio.

The solution provides an improvement of around 100% over the hit ratio on the skip-copy method proposed in [19] and reduces the access response time to one-sixth and one-tenth of that achieved by the skip-copy method for uniformly and Zipf distributed nodes respectively. However, this improvement is achieved at the cost of higher network traffic than the skip-copy method.

Updates to data items is supported and uses a read-any write-one approach. Since the updates are ordered by the update request generation time, the update mechanism supports total ordering and immediate ordering.

As we have shown, the proposed solution improves read accesses while not adding too much to the network traffic and supports total and immediate ordering for updates.

Future work can be directed at addressing the following issues in the solution.

- The update mechanism can be improved to have replica holders also accept updates and have some sort of gossip architecture([13]) to maintain consistency among the replica holders.
- Support for causal ordering.
- Flooding used in the solution can be avoided by using a geocasting model for routing packets as suggested in [3] and [12]. Since Geocasting routes packets to nodes that are near a specified position, it might be better suited for location dependent data replication.

6. CONCLUSION

In this paper, we proposed a solution to address the issues faced by replication for location dependent data in mobile ad hoc networks. The solution provides replication by identifying a *Primary Data Holder* (*PDH*) close to the location of the data item and replica holders along four directions - top, bottom, left and right - from the location of each data item and disseminating replicas to them. Accesses are serviced by the PDH or the replica holders and updates by the PDH alone. The solution proposed was able to achieve the following:

- Improvement in the access hit ratio by around 20% over not using replication for uniform and Poisson distributed nodes, and 10% for Zipf distributed nodes.
- A low response time for accesses to data items of around 0.055 seconds for uniform and Poisson distributed nodes which is the same as without replication, and 0.023 seconds for Zipf distributed nodes which is an improvement of around 100% over no replication.
- The increase in traffic generated was moderate.
- The solution provides an improvement of close to 100% on the hit ratio, and achieves accesses at one-sixth the response time for uniformly distributed nodes and one-tenth for Zipf distributed nodes, over an existing solution called the skip-copy method ([19]). However, this is done at the cost of higher network traffic.

The update hit ratio, however, was relatively lower when compared to the access hit ratio because of the read-any write-one approach for updates. The solution also introduces a limited amount of flooding to search for replica holders and backups.

Future work on the paper can explore improvements in updates and reducing the amount of flooding in the network.

7. REFERENCES

[1] P. Bellavista, A. Corradi, and E. Magistretti. Comparing and evaluating lightweight solutions for replica dissemination and retrieval in dense manets. *iscc*, 0:43–50, 2005.

[2] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2:257– 269, July-Sept 2003.

[3] C.-Y. Chang, C.-T. Chang, and S.-C. Tu. Obstacle-free geocasting protocols for single/multi-destination short message services in ad hoc networks. *Wirel. Netw.*, 9(2):143–155, 2003.

[4] N. Concer. Simulation of an ad hoc network.

[5] W. Drytkiewicz, S. Sroka, V. Handziski, A. Köpke, and H. Karl. A mobility framework for omnet++.

[6] M. H. Dunham and V. Kumar. Location dependent data and its management in mobile databases. In *Proceedings of the Ninth International Workshop on Database and Expert Systems Applications, 1998.*, pages 414–419, Washington, DC, USA, 1998. IEEE Computer Society.

[7] T. Hara. Replica allocation methods in ad hoc networks with data update. *Mobile Networks and Applications 8*, pages 343–354, 2003.

[8] T. Hara, Y.-H. Loh, and S. Nishio. Data replication methods based on the stability of radio links in ad hoc networks. *dexa*, 0:969, 2003.

[9] T. Hara, N. Murakami, and S. Nishio. Replica allocation for correlated data items in ad hoc sensor networks. *SIGMOD Record*, 33(1):38–43, Mar. 2004.

[10] H. Hayashi, T. Hara, and S. Nishio. Updated data dissemination methods for updating old replicas in ad hoc networks. *Personal and Ubiquitous Computing*, 9(5):273–283, Jan. 2005.

[11] J.-L. Huang and M.-S. Chen. On the effect of group mobility to data replication in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(5):492–507, May 2006.

[12] Y.-B. Ko and N. H. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. In *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, page 101, Washington, DC, USA, 1999. IEEE Computer Society.

[13] R. Ladin, B. Liskov, L. Shrira, and S. Ghemawat. Providing high availability using lazy replication. *ACM Trans. Comput. Syst.*, 10(4):360–391, 1992.

[14] K. Y. Lai, Z. Tari, and P. Bertok. Cost efficient broadcast based cache invalidation for mobile environments. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 871–877, New York, NY, USA, 2003. ACM.

[15] J. Luo and J.-P. Hubaux. Pan: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems. In *Proc. ACM MobiHoc 2003*, pages 1–12, Annapolis, Maryland, USA, 2003.

[16] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, page 90, Washington, DC, USA, 1999. IEEE Computer Society.

[17] Y. Sawai, M. Shinohara, A. Kanzaki, T. Hara, and S. Nishio. Consistency management among replicas using a quorum system in ad hoc networks. *mdm*, 0:128, 2006.

[18] M. Tamori, S. Ishihara, T. Watanabe, and T. Mizuno. A replica distribution method with consideration of the positions of mobile hosts on wireless ad-hoc networks. *icdcsw*, 0:331, 2002.

[19] G. Tsuchida, T. Okino, T. Mizuno, and S. Ishihara. Evaluation of a replication method for data associated with location in mobile ad hoc networks. In *ICMU'05*, pages 116–121, 2005.

[20] A. Varga. The omnet++ discrete event simulation system. In *ESM '01: Proceedings of European Simulation Conference*, Prague, Czech Republic, June 2001.

[21] H. Yu, P. Martin, and H. Hassanein. Cluster based replication for large-scale mobile ad-hoc networks. In *Proc. 2005 International Conference on Wireless Networks, Communications and Mobile Computing*, volume 1, pages 552–557, 2005.