Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

IMPROVING PERFORMANCE ENHANCEMENT & RELIABLE ROUTING OVER A MANET BY AVOIDING SELFISH NODE

Saravanan S [1]
[1] Department of Information Technology
[1] PSNA College of Engg & Technology
[1] ssaravananme@gmail.com

ABSTRACT

A Mobile Ad Hoc network consists of wide range of mobile nodes that actively participate in data transmissions .The mobility and resource constraints of mobile nodes may lead to network partitioning or performance degradation. To avaid performance degradation several replication techniques have been proposed .The main objective is to reduce traffic overhead while achieving high data accessibility. Some nodes may selfishly decide to cooperate partially or not at all with other nodes. These selfish nodes could reduce the overall data accessibility in the network. In this Project using cluster replica allocation techniques with the developed selfish node detection method. The cluster replica allocations methods each cluster contain the cluster head. This cluster head are connected to nodes in single hop method. This cluster head are detect any selfish node cluster head are detect the another shortest route and transmission the data to the destination node. So we can achieve high data accessibility with low-communication cost and the reduced the delay time of the packet transmission in the presence of selfish nodes.

Keywords: Mobile adhoc Network, Selfishnode Detection, Replica Allocation

1. INTRODUCTION

MOBILE ad hoc networks (MANETs) have attracted a lot of attention due to the popularity of mobile devices and the advances in wireless communication technologies. MANET is two types open and close [4]. A MANET is a multihop mobile wireless network that has neither a fixed infrastructure Nora central servers. Each node in a MANET act as a router,and communicates with each other. A large variety of application have been developed . For example, a MANET can be used in special situations,where installing infrastructures may be difficulty, or even infeasible, such as a battle field or a disaster areas. A peer-to-peer file sharing system is another interesting MANET applications [5]. Network partitions can occur frequently, since nodes moves freely in a MANET, causing some data to be often inaccessible to some of the nodes. Hence, data accessibility is often an important performance metrics in a MANET . Data are usually replicated at nodes, other than the original owner, to increase data accessibility to cope with frequent network partition. A considerable amounts of research has recently been proposed for replica allocation in a MANET[7].

In general, replication can simultaneously improve data accessibility and reduces query delay, query response time, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replica and the original data. For example, the response time of a query can be substantially reduces, if the query accesses a data item that has a locally store replica. However, there is often a trade-off between data accessibility and query delay, since most node in a MANET have only limited memory spaces .For example, a node may hold a parts of the frequently accessed data items locally to reduce its own query Delays. However, if there is only limited memory spaces and many of the nodes hold the same replica locally, then some data item would be replaced and missing. Thus, overall data accessibility would be decrease. Hence, to maximizes data accessibility, a

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

nodes should not hold the same replicas that is also held by many other node. However, this will increases its own query delays. A nodes may acts selfishly, using its limited resources only for its own benefits, since each node in a MANET has resource constraints, such as battery and storage limitation.

A node would like to enjoy the benefit provided by the resources of other node, but it may not make its own resources available to help other. Such selfish behavior can potentially lead to a wide range of problem for a MANET. Existing research on selfish behaviors in a MANET mostly focus on the network issues [3], [9] . For example, selfish node may not transmits data to others to conserved their own batteries. Although network issues are important in a MANET's, replicas allocation is also crucial, since the ultimate goals of using a MANET is to provides data service to user.

In this paper, we address the problem of selfishness in the contexts of replica allocations in a MANET, a selfish nodes may not share its own memory spaces to store replica for the benefit of other node. We can easily find such case in a typical peer-to-peer applications. For example, in Gnutella [2] , nearly 70 percent of users do not share their storage for the benefit of other. The number of selfish users has increased to 85 percents of all Gnutella users over five years [6] . In this paper, we shall refer to such a problem as the selfish replica allocations. Simply, selfish replica allocation refer to a node's non cooperative action, such that the node refuse to cooperate fully in sharing its memory spaces with other node. To our knowledge, this work is one of few works to cope with selfish nodes in the context of replica allocation over a MANET.

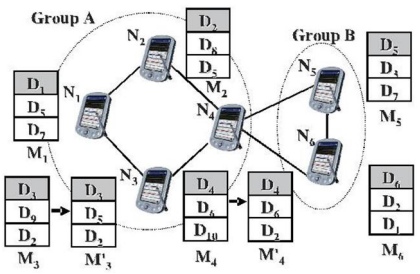

Figure.1: selfish replica allocation

Fig 1 illustrates an existing replica allocation scheme, DCG , where nodes N1;N2; . . .;N6 maintain their memory space M1;M2; . . .;M6, respectively, with the access frequency information in (In Fig. 1, a straight line denotes a wireless link, a gray rectangle denotes an original data items, and a white rectangle denotes a replica allocated. The gray colored area shows three data items that are accessed frequently by N3 and N4). As shown in Fig. 1, DCG seeks to minimize the duplication of data item in a group to achieve high data accessibility [1] [7]. Let us consider the cases where N3 behave "selfishly" by maintaining M03 , instead of M3, to prefers the locally frequently accesses data for low query delay. In the original cases, D3, D9, and D2 were allocated to N3. However, due to the selfish behaviors, D3, D5, and D2, the top three most locally frequently accessed items, are instead maintain in local storage. Thus, other nodes in the same group, N1, N2, and N4, are no longer able to accesses D9. This show case degraded data accessibility, since N1, N2, and N4 cannot fully leverage N3's memory space as intended in cooperative replica sharing.

As another example, a node may be only "partially selfish" in a MANET. For instance, node N4 may want to locally hold D2, one of the locally frequently accessed data items. In this case, N4 uses only a part of its storage for its own frequently accessed data, while the remaining part is for the benefit of overall data accessibility. Thus, N4 may decide to maintain M04 , instead of M4. Even with only partial selfishness, data accessibility is still degraded, since the other nodes in the same group, i.e., N1, N2, and N3.

We believe that the partially selfish node should also be taken into account, in addition to the fully selfish node to properly handle the selfish replica allocations problem. Therefore needs to measures the "degree of selfishness" to appropriately handle the partially selfish node. Motivated by this concepts of "partial selfishness," we borrow the notions of credit risk (CR)[1] [10] from economics to detect selfish node. Since the credit risk is calculated from several selfishness features in this paper, it can measures the degree of selfishness elaborately. In our scheme, a node can measures the degree of selfishness of another node, to which it is connect by one or multiple hop in a MANET

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

2. PRELIMINARIES

A. Monitoring and Detection:

Node can form a belief about the behaviors of other node by keeping track of direct observation and experiences. By the use of the so-called passive acknowledgments they can monitor their neighborhoods. Passive acknowledgments means that instead of waiting for an explicit acknowledgments for each packet by the next-hop node on the routes, a node assume the correct reception of the packets when it overhears the next-hop node forwarding the packets. A passive acknowledgment is possible in environment with bidirectional link and is a standard alternative to explicit acknowledgments, where node sends an acknowledgments to the previous hop upon receipt of a packets. The simple passive acknowledgments not only for an indication of correct reception at the next hop, but also to detects if nodes fail to forward packet. We enhanced the passive acknowledgment mechanism to detects several kind of misbehavior. We added capabilities to compare packet to detect the illegitimate modification of header fields and the fabrication of message

A mobile ad hoc network is an independent networks of mobile devices connect by wireless link. Each device in a MANET can moves freely in any direction, and will therefore changes its link to other devices easily. Each must forward traffic of others, and therefore be called a routers. The main challenges in building a MANET is in terms of security. we are presenting the mathematical model to detects selfish node using the probability density functions. A selfish Node minimizes efficiency of packets transfer and maximizes the packet delivery time and packet loss rates that divides a networks into smaller networks.

There are various methods to detect selfish nodes. These methods are categorized in incentive-based methods or reputation-based methods. In the first method it discourages a node to become selfish by giving virtual money or credits when a node forward packets of others because to send or receive its own packets the node requires enough credit. MANETs are not perfect. The challenges of security, scalability, mobility, bandwidth limitations, and power constraints of these networks have not been completely alleviated to date. But the security problem can be minimized by using the proposed mathematical model that expresses the detection of selfish node in MANETs to secure the network. This mathematical model is verified by experimentation and gives acceptable accuracy and provides a solution for secured routing in independent environment, because it uses heuristic model rather than deterministic. So, this model gives more accurate information using the defined probabilistic mathematical model.

B. Reputation and Trust

The most relevant properties of a reputation systems are the representation of reputations, how the reputation is built and update, and for the latter, how the rating of others are considered and integrated. The reputation of a given nodes is the collection of ratings maintain by other about this nodes. In our approach the reputation system is fully distributed, and a node maintain ratings about every other nodes that is care about. The reputation rating represent the opinion formed by node parallel about node's behaviors as an actor in the base system. The trust rating represent node parallel's opinion about how honest node is as an actor in the reputation system.

Response: The Path Manager:

A node classifies another node as misbehaving, isolates from communication by not using for routing and forwarding and by not allowing using. This isolation has three purposes. The first is to reduce the effects of misbehavior by depriving the misbehaving node of the opportunity to participate in the networks. The second purpose is to serve as an incentive to behave well in orders not to be denied services. Finally, the third purpose is to obtain better services by not using misbehaving node on the path.

C. DSR Protocols:

Dynamic Source Routing is a protocol developed for routing in mobile ad-hoc network. Node send out a ROUTE REQUEST message, all node that receive this messages forward it to their neighbor and put themselves into the source routes unless they have received the same request before. If a receiving node is the destination, or has a route to the destination[8], it does not forward the request, but send a REPLY messages containing the full source routes. It may send that reply along the source router in reverse order or issue a ROUTE REQUEST including the routes to get

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

back to the source, if the former is not possible due to asymmetric link. ROUTE REPLY message can be triggered by ROUTE REQUEST messages or gratuitous. After receiving one or several route, the source picks the best and the sooner the REPLY arrived at the sources, the higher preference is given to the route and the longer it will stay in the caches. In case of a link failure, the node that cannot forward the packet to the next nodes send an error messages toward the source.

Routes that contain a failed links can be 'salvaged' by taking an alternates partial route that does not contain the bad links. Replica allocation for performances improvement in the field of fixed networks has been an extensive research topics. In many researches, the communication cost is used as costs function. However, because these researches are for fixed network, they do not consider the effects on the data replication caused by the node mobility. A minimum- spanning- tree (MST) write policy is introduce.

However, this cost model is not suitable for the MANET environments because the communication costs and the algorithm complexity of building spanning trees are very high in MANET. In node forward read request to the nearest replica node and write request to all replica node along the shortest paths. However, this scheme requires that every nodes should maintain information of all replica node. When a replica node changes, every node must be notified. Thus it is not suitable for the mobile environments as well. A distributed dynamic adaptive replica allocation algorithm is pro-posed for the MANET environments. The communication cost is used as the cost functions in the algorithm because the communication costs become the most important factor which influences the performances of data access in these environments. Our algorithm can dynamically adjust the replica allocations scheme towards a local optimal one according to the access request distribution and topology changes. The concept of "stable neighbor" is proposed in our algorithm and the access request are collected only from stable neighbor while replica node expanding or relinquish.

D. Replica Allocation Method:

Data items are periodically updated. Each mobile host creates replicas of the data items, and maintains the replicas in its memory space. When a mobile host issues an access request for a data item, the request is successful in either case: (i) the request issue host itself holds the original/replica of the data item or (ii) at least one mobile host which is connected to the request issue host with a one-hop/multihop link holds the original/replica. The request issue host checks whether or not it holds the original/replica of the target data item. If it does, the request succeeds on the spot. If it does not, it broadcasts the request for the target data item. Then, if it receives a reply from another host which holds the original/replica of the target data item, the request is also successful. Otherwise, the request fails.

At a relocation period, each mobile host broadcast its host identifier and information on access frequencies to data item. After all mobile host complete their broadcasts, from the received host identifier, every host knows its connected mobile host. In mobile host which are connected to each others, starting from the mobile hosts with the lowest suffix (i) of host identifiers (Mi), the following procedures are repeated in the order of the breadth first search. When there is duplication of a data items (original/replica) between two neighboring mobile host, and if one of them is the original, the hosts which holds the replica replaces it with another replicas.

The mobile adhoc if both of them are replicas, the host whose PT values to the data item is lower replace the replica with another replicas. When replacing the replica, from among data item whose replicas are not allocated at either of the two hosts, a different replicated data items are selected whose PT values are the highest. A relocation period, a mobile host might not connect to another mobile hosts which has an original or a replica of a data item that the host should allocate. In this cases, in the same ways as in the E-SAF method, the memory space for the replica is temporarily filled with another replicas, and is later filled with the valid replica when data access to the data item succeed.

M1–M2 D3 →D4 (M1), D5 → D8 (M2),
M1–M3 D5 →D7 (M3),
M2–M4 D2 →D7 (M4),

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

M3–M4 D7 →D8 (M3),
M4–M5 D5 →D1 (M4), D4 → D8 (M5),
M4–M6 no duplication,
M5–M6 D3 →D7 (M6).

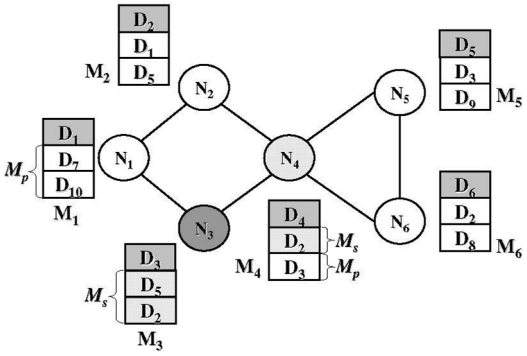

Figure. 2: E-DAFN Method

E. Network Assumption:

1. The request issue host broadcasts a data search packet in the network.
2. A mobile host which has received the data searches packet checks whether it has the original or a replica of the data item requested by the request issue host.
   - If it has the original, it returns a message notifying the request issue host of that fact and whether the update has occurred or not. At the same time, it stops the broadcasting of the data search packet.
   - If it does not have the original but has a replica, it returns a message notifying the request issue host of that fact, and continues the broadcasting of the data search packet.
   - If it does not have either the original or a replica, it does not reply to the request issue host, and continues the broadcasting of the data search packet.
3. The request issue host behaves according to the received reply packets.
   - In the case of receiving a reply packet from the owner of the original: If the request issue host has a replica of the target data item and the update has not occurred, it accesses the replica.
   - Otherwise, it sends a data request packet to the owner of the original.
   - In the case of not receiving a reply packet from the owner of the original but receiving it from the owner(s) of replica(s):If the request issue host has a replica, it makes an interim access to its own replica. However, the interim access will fail when the request issue host connects to the owner of the original and finds that the update has occurred, i.e., the interim access becomes a dirty read.
   - In the case of not receiving any reply packet: If the request issue host has a replica, it makes an interim access to its own replica. Otherwise, the request fails.
4. A mobile host which has received the data request packet sends the requested data/replica to the request issue host.

The request issue host which has sent the data request packet accesses the received.

3. PROPOSED STRATEGY

A. Overview

In an ad hoc network, mobile nodes communicate with each other using multihop wireless link. There is no stationary infrastructures; for instances, there are no base stations. Each nodes in the networks also act as a router, forwarding data packet for other node. A research issue in the design of ad hoc network is the development of dynamic routing protocol that can efficiently find route between two communicating nodes. The routing protocol must be able to keep up with the high degree of node mobility that often changes the network topology. In a large network, flat routing schemes produce an excessive amount of information that can saturate the networks. In addition, given the nodes heterogeneity, node may have highly variable amount of resources, and this produce a hierarchy in their role inside the networks. Node with large computational and communication power and powerful batteries are more suitable for supporting the ad hoc network function than other node.

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

We address the problem of selfishness in the contexts of replica allocation in a MANET.A selfish nodes won't share its own memory space to store replica for the benefit of other node. We shall refer to such a problem as the selfish replica allocations. In our proposed system we are using cluster replica allocation techniques with the developed selfish node detection method.

Can achieve high data accessibility with low-communication cost in the presence of selfish nodes. Propose a localized algorithm that creates a set of clusters called; stable K-hop direct acyclic graphs (SKDAG). SKDAG is a K-hop DAG, in which each node has strong paths toward the sink nodes. Each cluster must ensures the following property: The cluster is a direct acyclic graphs at the cluster head. Each node of the cluster is K hops away from the cluster head. The path between a nodes and its cluster head consists only of long-lived nodes and link, i.e. nodes and links whose residual life time is greater than a given thresholds.

Whenever data of particular node is update, that node send the new version of data to its cluster head. Upon receiving such an updates, the cluster head periodically generate an update message containing all data item that was changed during the last time periods. This approach reduces the update cost since it assemble a number of data item in one update message, but it increase the outdated data item returned by query operation.

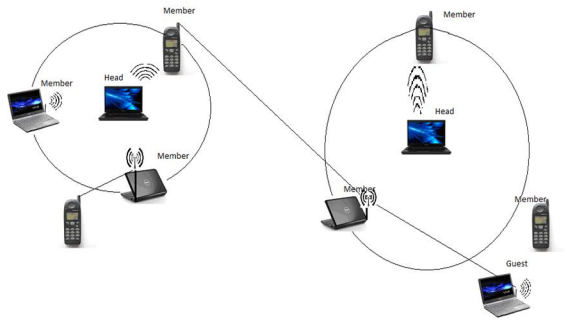

Figure. 3: System Architecture

B. Selfish Node Detection

The notion of credit risk can be described by the following equation:

$$CreditRisk = \frac{expectedrisk}{expectedvalue}$$

In our strategy, each node calculates a CR scores for each of the node to which it is connected. Each node shall estimates the "degree of selfishness" for all of its connected nodes based on the scores. We first describe selfish features that may lead to the selfish replica allocation problems to determine both expected values and expected risk[1] [10].

Each node will calculate Credit Risk (CR) score of each node connected with them. It estimates the degree of selfishness of connected node based on the CR scores. Detect and exclude strategy avoids selfish node from the routing paths. This scheme uses two types of trust namely first hand trust and second hand trust.

- First hand trust: The node's personal observation about the neighboring nodes.
- Second hand trust: The observation communicated by neighboring node about the other neighbors of the network.

C. Replica Allocation

Cluster-based routing is a solution to address nodes heterogeneity, and to limit the amount of routing information that propagates inside the network. The idea behind clustering is to group the network nodes into a number of overlapping clusters. Clustering makes possible a hierarchical routing in which paths are recorded between clusters instead of between nodes. This increases the routes lifetime, thus decreasing the amount of routing control overhead. Inside the cluster one node that coordinates the cluster activities is cluster head (CH).

Inside the cluster, there are ordinary nodes also that have direct access only to this one cluster head, and gateways. Gateways are nodes that can hear two or more cluster heads. Ordinary nodes send the packets to their cluster head that either distributes the packets inside the cluster, or (if the destination is outside the cluster) forwards them to a gateway node to be delivered to the other clusters. By replacing the nodes with clusters, existing routing protocols can be

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

directly applied to the network. Only gateways and cluster heads participate in the propagation of routing control/update messages. In dense networks this significantly reduces the routing overhead, thus solving scalability problems for routing algorithms in large ad hoc networks.

Each node is associated with cluster and each cluster has its Cluster Head (CH). CH will maintain ART (Available Replica Table). When a node needs data item, then it sends a request to the CH. CH will check in ART, and then request is forward to the corresponding node. When a node receives the data item, it will make a replica for future use. Update Message is sent to CH.

D. Route Maintenance

Routing in a reactive protocol typically consists of three parts: route discovery, data forwarding, and route maintenance. This paper studies the route maintenance problem in a MANET. When a mobile host wants to communicate with another host, it first tries to discover a good route to the destination, on which the data packets are forwarded.

Route maintenance, by its name, should address the problem when a route becomes worse or even broken due to host mobility. However, in existing protocols, such as a sending host will stick with the discovered route until it is expired or broken, even if some better routes are newly being formed in the system. One straight forward solution is to run the route discovery procedure more frequently to detect such possibility. However, this is very costly as route discovery will typically activate a network flooding. This observation has motivated the first work in this paper: we propose to use route optimization to refine or improve the routes on-the-fly while they are being used for transmission. Not only can the data packets be sent with less hops and latencies, but also may the chances of route breakage be reduced, lowing the number of times the costly route discovery process being called.

A link state change has to be periodically updated by all nodes to maintain updated information. Any node found link failure; it has to be informed to all of its neighbors through control messages. On receiving, each has to update its route information.

4. SIMULATION RESULTS

A. Effectiveness of Detection Method

We first compare the overall selfishness alarm of DCG with that of DCG to demonstrate the effectiveness of our detection method. We expect that the overall selfishness alarms will be reduced in query processing by detecting selfish node effectively with DCG, since many selfish nodes will be remove from the replica allocation phase and many reliable nodes will serve data requests from node. However, recall that the selfishness alarm may also occur due to network disconnections, false alarms . Actually, it is desirable to observe truly selfish nodes to evaluate the effectiveness of the detection methods As mentioned earlier, a data requester cannot tell an expected node's selfishness from network disconnection, since their impacts are identical to the requester, no reply from the expected node. Although the false alarm exists from the viewpoint of nodes, we realize that the true selfishness can be identified in the simulation results by identifying which data request has not been served by the expected, connected node in query processing. Obviously, the expected and connected nodes are only involved in a true selfishness alarm, whereas the expected but disconnected nodes in query processing may lead to a false alarm. Therefore, we plot two additional methods, DCG (selfishness only) and DCG (selfishness only) . The overall selfishness alarm of DCG (selfishness only) and DCG (selfishness only) is obtained by counting data requests that have not been served by the expected, connected nodes in query processing, excluding false alarms caused by disconnections.

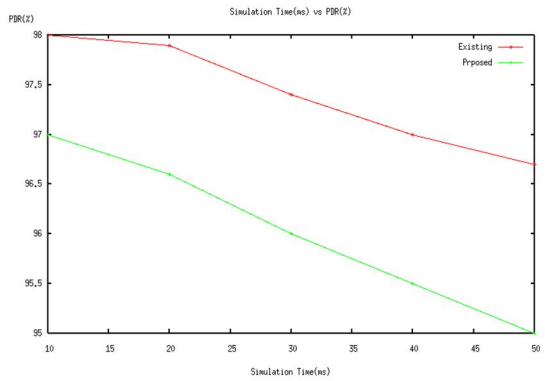

Figure..4 simulation time vs. PDR%

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

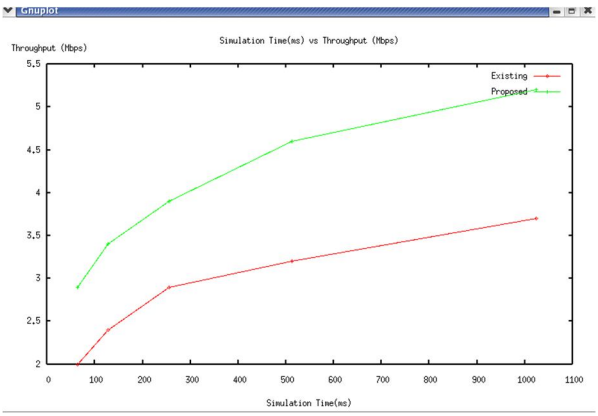

Figure..5 Iteration vs. Active Nodes

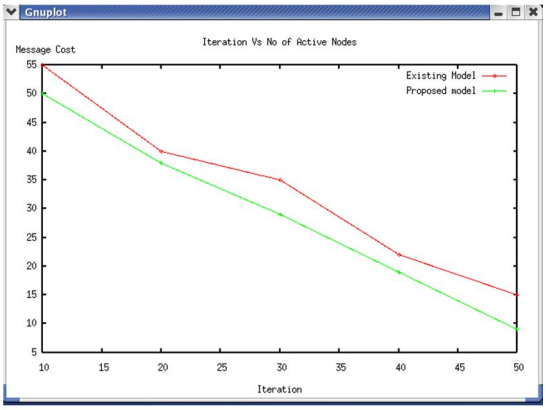

Figure.6 Time vs No of Triggered Of ch Processing

B. Communication Cost

We evaluate several replica allocation techniques in terms of communication costs. . This intuition is confirmed by the results DCG shows the worst performances in all case, since group members need to communicates with each others in detecting selfish node and allocating/relocating replicas. We report that, on average, about 70 percent of total communication cost in the DCG techniques is caused by replica allocation/relocation(fig:5), while about 30 percent is caused by selfish node detections. As expected, SAF shows the best performance, since no detection of selfish nodes or group communication is made.

C. Average Delay

Average query delay for various parameters. As expected, the SAF technique shows the best performance in terms of query delay, since most successful queries are served by local memory spaces. Our techniques show slightly better query(Fig:7) delay than does the DCG technique The DCG technique shows the worst performances. This can be explained as follows: the distance in hop count among group members in the DCG technique is longer than that in the DCG technique. Since most successful queries are served by group members in these technique, the long distance among group members affect query delay negatively. Among our techniques, the eSCF technique shows the best average query delay. In the eSCF technique, nearby selfish nodes can be added to the eSCF-tree. Consequently, some queries are possibly served by the nearby (partially) selfish nodes, whereas only nonselfish node, which maybe far away, serve queries in other techniques.

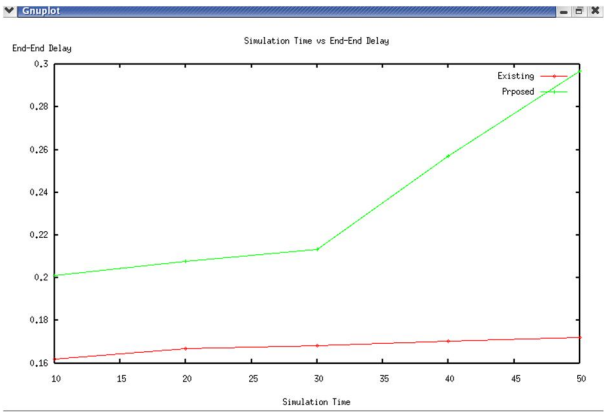

Figure..7 Simulation Time vs End to end delay

Asian Journal of Computer Science and Technology (AJCST)
Vol1.No.2 2013 pp 38-46.
available at: www.goniv.com
Paper Received :19-03-2013
Paper Published:18-05-2013
Paper Reviewed by: 1. John Arhter 2. G.Rasmesh
Editor : Prof. P.Muthukumar

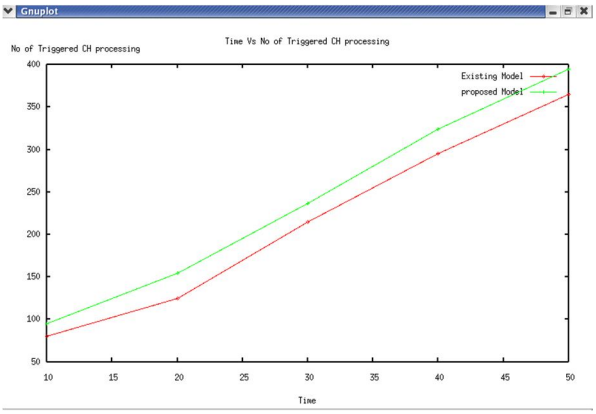

Figure8: Simulation Time(ms) vs Throughtput(mbps)

D. Data Accessibility

We evaluate the data accessibility of replica allocation methods under consideration. We expect that our techniques perform significantly better than other techniques in the presence of selfish nodes. the strength of our methodology: in all cases, our techniques outperform SAF, DCG, and DCGþ considerably, since our techniques can detect and handle selfish nodes in replica allocation effectively and efficiently. Among our techniques, the eSCF technique shows a slightly poorer performance. Our initial intuition was that, data accessibility is stable with relocation periods. This is confirmed by the results in that data accessibility is proportional to the size of memory space, as expected. The performance of our techniques improves faster than do others, since our techniques fully utilize the memory space of nodes.

5. CONCLUSION

The problem of selfish nodes is very common in ad hoc network. The major reason for the selfishness is the loss of power with time. As the time passes away the node lose their battery power and in a disasters hit area or battle field areas recharging may not be easily possible. The experimental result show that up to a concentration level of 10%, selfish nodes do not have remarkable negative effect on the networks activities. As the concentration increases QoS become poorer and poorer though the network never comes to halt even if the selfish node concentration reaches to nearly 100%. The average hop count reaches to a maximum 2.5 times, Probability of Reach ability and throughput comes down to nearly 50% at its peak and percentage of packet drop goes up to nearly 60% at the most. Since the concentration of selfish.

REFERENCES

[1]. Jae-Ho Choi, Kyu-Sun Shim, SangKeun Lee, and Kun-Lung Wu, Fellow" *Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network*" IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 11, NO. 2, FEBRUARY 2012.

[2]. E. Adar and B.A. Huberman, "*Free Riding on Gnutella*," First Monday, vol. 5, no. 10, pp. 1-22, 2000.

[3]. L. Anderegg and S. Eidenbenz, "Ad Hoc-VCG: *A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents*," Proc. ACM MobiCom, pp. 245-259, 2003.

[4]. K. Balakrishnan, J. Deng, and P.K. Varshney, "*TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks*," Proc. IEEE Wireless Comm. and Networking, pp. 2137-2142, 2005.

[5]. G. Ding and B. Bhargava, "*Peer-to-Peer File-Sharing over Mobile Ad Hoc Networks*," Proc. IEEE Ann. Conf. Pervasive Computing and Comm. Workshops, pp. 104-108, 2004.

[6]. M. Feldman and J. Chuang, "*Overcoming Free-Riding Behavior in Peer-to-Peer Systems*," SIGecom Exchanges, vol. 5, no. 4, pp. 41-50, 2005.

[7]. T. Hara, "*Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility*," Proc. IEEE INFOCOM, pp. 1568-1576, 2001.

[8]. B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C.H. Papadimitriou, and J. Kubiatowicz, "*Selfish Caching in Distributed Systems: A Game-Theoretic Analysis*," Proc. ACM Symp. Principles of Distributed Computing, pp. 21-30, 2004.

[9]. D. Hales, "*From Selfish Nodes to Cooperative Networks -Emergent Link-Based Incentives in Peer-to-Peer Networks*," Proc.IEEE Int'l Conf. Peer-to-Peer Computing, pp. 151-158, 2004.

[10]. L.J. Mester, "*What's the Point of Credit Scoring?*" usiness Rev.,pp. 3-16, Sept. 1997.