Handling Selfishness in Fast Detection Replica Allocation over Mobile Ad-hoc Network

E.Kodhai[#1], K.Indumathi[#2]
[#1]*Associate Professor Dept Of IT*, [#2]*M.Tech Networking , Pondicherry University*
[#1, #2]*Sri Manakula Vinayagar Engineering College, Madagadipet , Pondicherry-605107.*

[#1]kodhaiej@yahoo.co.in , [#2]indumathy2610@gmail.com

Abstract— **In Mobile ad-hoc Network (MANET) some of the nodes do not take part in forwarding packets to other nodes to conserve their resources such as energy, bandwidth and power. The nodes which act selfishly to conserve their resources are called selfish nodes. The selfish nodes are engaged to reduce data availability and produce high communication cost in terms of query processing. Many selfish node detection methods are found to detect the nodes which do not participate in packet forwarding but they fail to detect the selfish nodes which does not allocate replica for the purpose of other nodes. The methods are provided to detect selfish nodes in terms of allocating replica to other nodes. The methods are categorized according to detect the selfish nodes and reduce the impact of that nodes in mobile ad hoc network. The key features discussed are the selfish nodes and numerous replica allocation techniques. The introduction of new replica allocation techniques will considerably reduce the delay in query processing and produce high data availability. Thus a new proposed solution is an effective mobile replica node detection scheme using Sequential Probability Ratio Test (SPRT).These replica node attacks are hazardous because they allow the attacker to influence the compromise of a few nodes to make use of control over much of the network.**

Key Words— **Mobile ad hoc Networks, Degree of Selfishness, Selfish Replica Allocation, Sequential Probability Ratio Test.**

I. INTRODUCTION

Mobile ad-hoc network (MANETs) have attracted a lot of attention due to the popularity of mobile devices and the advances in wireless communication technologies [3], [8], [9]. In MANETs every node acts as a router and communicates with each other nodes. If the source and the destination mobile hosts are not in the coverage area, data packets are forwarded to the destination host through other nodes which exist between the two mobile hosts. MANETs does not require any infrastructure and base station. According to [1] MANETs are applicable in many situations such as battlefield and disaster area. In ad hoc network, as all the nodes are having mobility, they move freely. This mobility causes frequent network partitions hence data accessibility in ad hoc networks is lower than the fixed networks. The nodes which are not willing to forward packets and share their memory space are called selfish nodes. The selfish node that does not allocate data items for the purpose of other nodes is called selfish replica allocation. Network partitions can occur frequently, since nodes move freely in a MANET, causing some data to be often inaccessible to some of the nodes. Hence, data accessibility is often an important performance metric in a MANET [2]. Data are usually replicated at nodes, other than the original owners, to increase data accessibility to cope with frequent network partitions. A considerable amount of research has recently been proposed for replica allocation in a MANET [2] [3] [10].The selfish nodes allocate data items that are highly accessed by it and do not consider other nodes during replica allocation. Selfish nodes reduce the data accessibility of other nodes in query processing. The selfish nodes do not satisfy neighbour nodes by giving required data to them. According to [1] nodes can be divided into three types they are,

- Non selfish nodes
- Fully selfish nodes
- Partially selfish nodes

Non selfish nodes allocate their memory space completely for the purpose of other nodes. Selfish nodes do not allocate their memory space for the purpose of other nodes. Partially selfish nodes allocate minimum portion of their memory space for the purpose of other nodes and remaining for the benefit of own node. Minimizing the effects of selfish nodes will be important to increase the data accessibility between the nodes. A node would like to enjoy the benefits provided by the resources of other nodes, but it may not make its own resource available to help others. Such selfish behavior can potentially lead to a wide range of problems for a MANET. For example, selfish nodes may not transmit data to others to conserve their own batteries [1] [5]. Existing research on selfish behaviours in a MANET mostly focus on network issues [11], [12], [13]. Fig. 1 illustrates an existing replica allocation scheme, DCG [4] [6], where nodes N1,N2, . . .,N6 maintain their memory space M1,M2, . . .,M6, respectively, with the access frequency information in Table 1.In Fig. 1, a straight line denotes a wireless link, a gray rectangle denotes an original data item, and a white rectangle denotes a replica allocated. Fig. 1, DCG seeks to minimize the duplication of data items in a group to achieve high data accessibility.

In Fig.1 where N3 behaves “selfishly” by maintaining M3’s, instead of M3, to prefer the locally frequently accessed data for low query delay. In the original case, D3, D9, and D2 were allocated to N3. However, due to the selfish behavior, D3, D5, and D2, the top three most locally frequently accessed items, are instead maintained in local storage. Thus, other nodes in the same group, i.e., N1, N2, and N4, are no longer able to access D9. This showcases degraded data accessibility. This paper, address the problem of selfishness in the context of replica allocation in a MANET, i.e., a selfish node may not share its own memory space to store replica for the benefit of other nodes

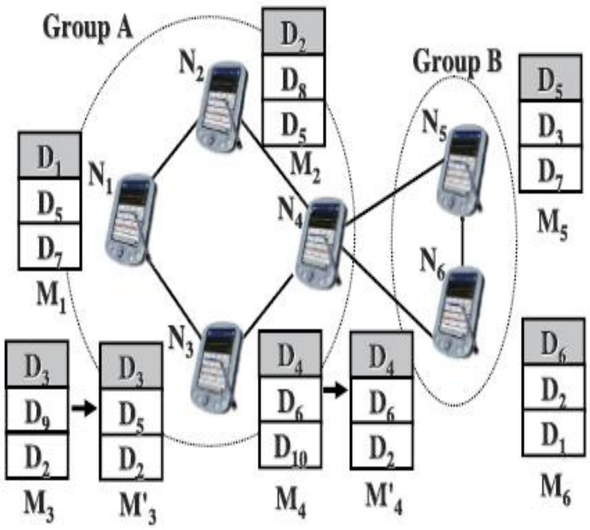

Fig.1 Example of Selfish Replica Allocation

Easily find such cases in a typical peer-to-peer application referring to such a problem as the selfish replica allocation. The selfish replica allocation problem based on the concept of a self-centred friendship tree (SCF-tree) and its variation to achieve high data accessibility with low communication cost in the presence of selfish nodes.

II. RELATED WORK

In the work reported by M.siva et al., [6] in mobile ad hoc networks (MANETs), mobile nodes move freely from one place to another place and link/node failures are common. Due to link/node failures leads to frequent network partitions. When network a network partition occurs, mobile nodes in one partition are not able to access data contained in other partition. For this reason the performance of the data access significantly decreased. To avoid such problems data replication scheme proposed to improve the performance of the data access, and hence data availability and query traffic increased and query delay should decrease, also, query rate query delay decreased and hit ratio increased.

In the work reported by M.Shorfuzzaman et,al, [14] the algorithm exploits data access histories to identify popular files and determines optimal replication locations to improve access performance by minimizing replication overhead (access and update) assuming a given traffic pattern. The problem is formulated using dynamic programming. It offers shorter execution time and reduced bandwidth consumption compared to other dynamic replica placement methods.

D.Nukarpu et.al, [15] has proposed data replication algorithm that not only has a provable theoretical performance guarantee, but also can be implemented in a distributed and practical manner. Specifically, the author design a polynomial time centralized replication algorithm that reduces the total data file access delay by at least half of that reduced by the optimal replication solution. Based on this centralized algorithm, the author also design distributed caching algorithm, which can be easily adopted in a distributed environment such as Data grids.

In the proposal work by Y.Fang et.al, [16] research is based on the sub dividable area. The author proposes a new mechanism calls junction replication Method (JRM) in unstructured decentralized P2P network which reflects the most usual P2p network in the real environment. Set a new limit, if a normal node's request over that limit, the author also sends the replica to it directly. In this way, the author can avoid nodes become overloaded with low node in nodes.

Q.Gu et.al., [17] addressed the problem of replication data on multiple nodes can improve availability and response time. Yet determining when and where to replicate data in order to meet performance goals with many users and files, dynamic network characteristics, and changing user behaviour is difficult. Also it is a challenging problem to find the best – fit replica efficiency according to location and performance of physical storage systems at which the replicas dwell.

In the work reported by A.Christopher et.al, [18] the selfish caching issue in MANET is formulated as a non-cooperative problem. For solving this problem, we propose an efficient storage space allocation approach for data replication problem while nodes are in selfish behaviours. In this approach each node independently decides how allocate its available storage space. It is based on which each node determines the quality of service that the device will offer to each one of their requesters according to their reputations and demands. It dynamically adapts the capacity that they dedicate for uploading and downloading in order to improve their utility.

In the proposal reported by Min Mengl et.al, [19] they allocate the data replicas according to time and space. In temporal method, we store the original data, the median data in a specific time period and the replica with the second highest frequency among all the other data on the mobile hosts to improve data accessibility. In spatial method, we store the original data, the median data among all the neighbours and the replica with the second highest frequency. The advantages of storing the median data are for fault tolerance to eliminate the outliers so that we can get more accurate information of a specific time period or a region, to save memory spaces and energy of the mobile hosts.

Rashid Azeem et.al, reported the [20] Mobility and Power constrain of the Server and Client influence Data accessibility in MANETs, Data in MANET are replicated. Numeral of Data Replication techniques has been projected for MANET Databases. This scheme recognizes topics drawn in MANETs Data Replication and tries to categorize presented MANET Data Replication techniques found on problems they deal with limitation and presentation. Additionally, this paper also suggest criterion for choosing suitable Data Replication techniques for a variety of applications requirements in conclusion, the document end with a argument on future research directions.

In the proposal work reported by P.Mukilan et.al, [21] the main aim of the research work is to develop the data replication algorithm based on data availability, data consistency and load balancing in order to provide the minimum energy consumption and high data availability rate in the network. Due to the presence of the network partition, mobile nodes in one partition are not able to access the data hosted by nodes in the other partition. So the performance of data access is degraded. Existing methods aims at balancing trade-off between energy consumption, data availability and delay.

K.Santhi et.al, [22] has proposed the security of the unattended replica nodes attack is extremely critical, because they allow the attacker to leverage, or else the attacker may be able to capture the nodes, and then use them to injected fake data, data loss in network communication. In this proposed work a fast and effective mobile replica node detected using the Sequential Probability Ratio Test (SPRT) scheme in Game Theory. This work is to track the problem of replica node attacks in Mobile Ad-hoc Network from intruders and provide the best security. This paper aims to detect the intruders and increase the both speed and predictive accuracy.

In the proposal work reported by Chi-Anh La et.al, [23] we design a practical, distributed solution to content replication that is suitable for dynamic environments and achieves load balancing. Simulation results show that our mechanism, which uses local measurements only, approximates well an optimal solution while being robust against network and demand dynamics. Also, our scheme outperforms alternative approaches in terms of both content access delay and access congestion.

Jun-Won Ho et.al, has proposed [24] several replica node detection schemes have been proposed in the literature to defend against such attacks in static sensor networks. However, these schemes rely on fixed sensor locations and hence do not work in mobile sensor networks, where sensors are expected to move. The work proposed is a fast and effective mobile replica node detection scheme using the Sequential Probability Ratio Test. To the best of our knowledge, this is the first work to tackle the problem of replica node attacks in mobile sensor networks. We show analytically and through simulation experiments that our scheme detects mobile replicas in an efficient and robust manner at the cost of reasonable overheads.

III. PROPOSED SYSTEM

A. Overview

Proposing a selfish node detection method and novel replica allocation techniques to handle the selfish replica allocation appropriately. The proposed strategies are inspired by the real-world observations in economics in terms of credit risk and in human friendship management in terms of choosing one's friends completely at one's own discretion. We applied the notion of credit risk from economics to detect selfish nodes. Every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness. Since traditional replica allocation techniques failed to consider selfish nodes, we also proposed novel replica allocation techniques.

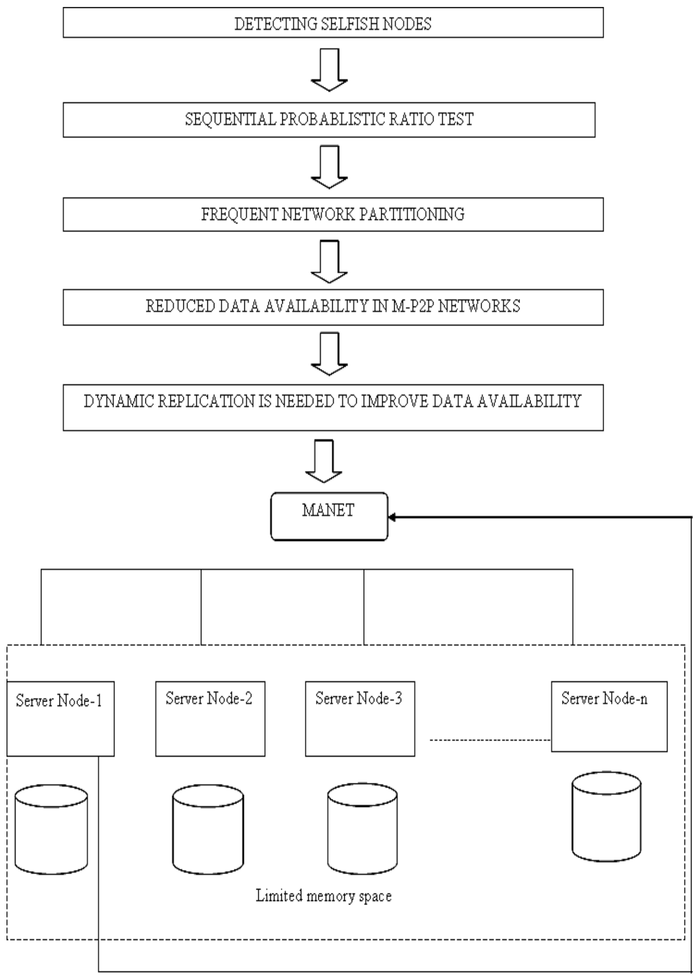

Fig. 2 Proposed Design

The selfish node is detected by the self replica allocation. They are based on the concept of a self-centred friendship tree (SCF-tree) and its variation to achieve high data accessibility with low communication cost in the presence of selfish nodes. The SCF-tree is inspired by our human friendship management in the real world. In the real world, a friendship, which is a form of social bond, is made individually. Another proposed technique is Sequential Probability Ratio Test.

The technical contributions of this paper can be summarized as follows.

- *Recognizing the selfish replica allocation problem:* We view a selfish node in a MANET from the perspective of data replication, and recognize that selfish replica allocation can lead to degraded data accessibility in a MANET.
- *Detecting the fully or the partially selfish nodes effectively:* We devise a selfish node detection method that can measure the degree of selfishness.
- *Allocating replica effectively:* We propose a set of replica allocation techniques that use the self-centred friendship tree to reduce communication cost, while achieving good data accessibility.
- *Verifying the proposed strategy:* The simulation results verify the efficacy of our proposed strategy.

4. Allocating the Replica

After building the SCF-tree, a node allocates replica at every relocation period. Each node asks non selfish nodes within its SCF-tree to hold replica when it cannot hold replica in its local memory space. Since the SCF-tree based replica allocation is performed in a fully distributed manner, each node determines replica allocation individually without any communication with other nodes.

The advantages of above methods are, easily detects the selfish node without collision. SCF-tree based replica allocation is performed in a fully distributed manner. Cooperative replica allocation techniques in terms of data accessibility, communication cost, and query delay so it is efficient.

B. Modules

The above proposed work has been classified into various modules they are;

- Node Creation and node configuration
- Selfish Node Detection
- Building SCF-Tree
- Allocating the Replica
- Detection Of Replica Allocation Using Sequential Probability Ratio Test

1. Node Creation and Node Configuration

This module is developed to create a node and more than 10 nodes placed at a particular distance. Wireless node placed at the intermediate area. Each node knows its location relative to the sink. The access point has to receive the transmitted packets and then send acknowledgement to the transmitter.

Non selfish nodes allocate their memory space completely for the purpose of other nodes. Selfish nodes do not allocate their memory space for the purpose of other nodes. Partially selfish nodes allocate minimum portion of their memory space for the purpose of other nodes and remaining for the benefit of own node. Minimizing the effects of selfish nodes will be important to increase the data accessibility between the nodes. The packets are sending and received through the source to destination. It is based on the scheme of packets delivered for ACK packet drop on the nodes. Mainly used to create the source and destination node of the network and transmit the data processing to their whole network.

2. Selfish Node Detection

The notion of credit risk can be described by the following equation: Credit Risk = expected risk / expected value In this strategy, each node calculates a CR score for each of the nodes to which it is connected. Each node shall estimate the “degree of selfishness” for all of its connected nodes based on the score. The Selfish features are divided into two categories: node specific and query processing-specific. The Node specific features can be used to represent the number of shared items & shared memory space for the node. The size of N_k’s shared memory space, denoted as SSi^K , and the number of N_k’s shared data items, denoted as $ND^k i$, observed by a node Ni ,are used as node-specific features. The node-specific features can be used to represent the expected value of a node.

For instance, when node N_i observes that node N_k shares large SSi^K and $ND^k i$, node N_k may be treated as a valuable node by node Ni. As the query processing-specific feature, utilize the ratio of selfishness alarm of N_k on N_i, denoted as Pi^k , which is the ratio of N_i’s data request being not served by the expected node N_k due to N_k selfishness in its memory space (i.e., no target data item in its memory space).Thus, the query processing-specific feature can represent the expected risk of a node. For instance, when Pi^k gets larger, node Ni will treat N_k as a risky node because a large Pi^k means that N_k’ cannot serve Ni’s requests due to selfishness in its memory usage. The value of the Cri^k is the credit risk of node Ni. § is the threshold value of node N_i. α is the system parameter, where $0 \leq \alpha \leq 1$.

Algorithm 1

Pseudo code for selfish node detection

```
detection()
{
   for (each connected node Nk )
{
    if (nCRk i < §)
    Nk is marked as non-selfish;
    else Nk is marked as selfish;
}
wait until replica allocation is done;
for (each connected node Nk)
{
  if (Ni has allocated replica to Nk)
```

```
{
  ND^k i= the number of allocated replica;
SS^k i = the total size of allocated replica;
}
else
{
  ND^k i =1;
  SS^k i = the size of a data item;
}
}
}
```

3. Building SCF-Tree

The SCF-tree based replica allocation techniques are inspired by human friendship management in the real world, where each person makes his/her own friends forming a web and manages friendship by himself/herself. He/she does not have to discuss these with others to maintain the friendship.

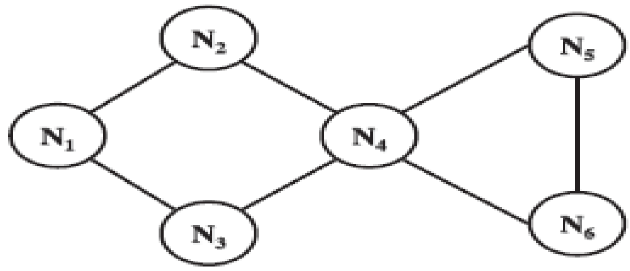

Fig.4 Sample Topology G

The main objective of the novel replica allocation techniques is to reduce traffic overhead, while achieving high data accessibility. Before constructing the SCF-tree, each node makes its own partial topology graph $G_i = (IN_i, IL_i)$, which is a component of the graph G. Gi consists of a finite set of the nodes connected to Ni and a finite set of the links, where Ni € IN_i , IN_i _ IN, and IL_i _ IL. Based on Gi^{ns}, Ni builds its own SCF-tree, denoted as Ti^{SCF}. Algorithm 2 describes how to construct the SCF-tree. Each node has a parameter d, the depth of SCF-tree. When Ni builds its own SCF-tree, Ni first appends the nodes that are connected to Ni by one hop to Ni's child nodes. Then, Ni checks recursively the child nodes of the appended nodes, until the depth of the SCF-tree is equal to d. Fig.4 illustrates the network topology.

Algorithm 2

Pseudo code to build SCF-tree

```
constructScfTree()
{
   append Ni to SCF-tree as the root node;
   checkChildnodes(Ni );
   return SCF-tree;
}
Procedure checkChildnodes(Nj)
{
   for (each node Na € INaj)
 {
   if (distance between Na and the root > d) continue;
   else if (Na is an ancestor of Nj in   TiSCF ) continue;
   else
 {
   append Na to Ti^SCF as a child of Nj;
   checkChildnodes(Na);
}
}
}
```

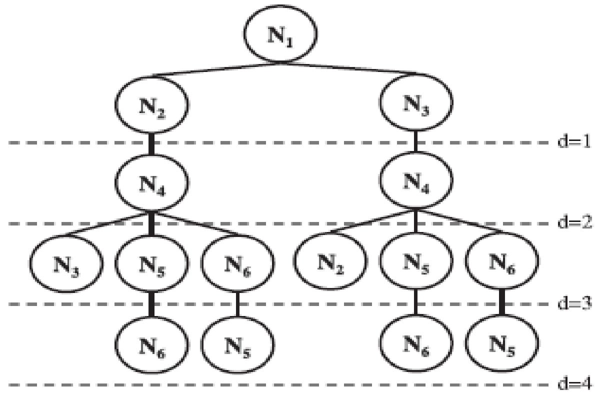

Fig.5 SCF-Tree of N_1

4. Allocating the Replica

After building the SCF-tree, a node allocates replica at every relocation period. Each node asks non-selfish nodes within its SCF-tree to hold replica when it cannot hold replica in its local memory space. Since the SCF-tree based replica allocation is performed in a fully distributed manner, each node determines replica allocation individually without any communication with other nodes. Since every node has its own SCF-tree, it can perform replica allocation at its discretion.

Algorithm 3

Pseudo code for replica allocation

```
/*Ni executes this algorithm at relocation period */
replica_allocation()
{
  L_i =make_priority(Ti^SCF);
  for (each data item belongs to ID_i)
{
   if (M_s is not full)
   allocate replica of the data to Ms ;
   else{/*Ms is full */
   allocate replica of the data to the target   node;
   /*the target node is selected from Li*/
   if (M_p is not full)
   allocate replica of the data to M_p;
} }
```

while (during a relocation period)
{
if (N_k requests for the allocation of D_q)
replica_allocation_for_others (N_k;D_q);
} }
Procedure make_priority (T_i^{SCF})
{
for (all vertices in (T_i^{SCF})
{
select a vertex in T_i^{SCF} in order of BFS;
append the selected vertex $_i$d to L_i;
}
return Li; }
Procedure replica_allocation_for_others(N_k;D_q)
{
if (N_k is in T_i^{SCF} and N_i does not hold D_q)
{
if (M_q is not full) allocate D_q to M_p ;
else{/*Mp is full*/
if(N_i holds any replica of local interest in M_p)
replace the replica with D_q ;
else{
/*N_h is the node with the highest $_nCR_i{}^h$
among the nodes which allocated replica to M_p*/
if ($_nCR_i{}^h > {}_nCR_i{}^k$)
replace the replica requested by N_h with D_q;
} } } }

5. Detection Of Replica Allocation Using Sequential Probability Ratio Test

The new proposed solution which will overcome the above scheme is Sequential probability Ratio Test. To design an effective, fast, and robust replica detection scheme specifically for mobile sensor networks. For the effective scheme a novel mobile replica detection scheme based on the SPRT. By using the fact that an uncompromised mobile node should never move at speeds in excess of the system-configured maximum speed. The main advantage of the scheme is to quickly detect captured nodes with the aid of the SPRT.

The SPRT can be thought of as one dimensional random walk with the lower and upper limits [5]. A random walk starts from a point between two limits and moves toward the lower or upper limit in accordance with each observation. If the walk reaches (or exceeds) the lower or upper limit, it terminates and the null or alternate hypothesis is selected, respectively. The lower and upper limits can be configured to be associated with speeds less than and in excess of V-max, respectively [5].To apply the SPRT to the mobile replica detection problem as follows: Each time a mobile sensor node moves to a new location, each of its neighbours asks for a signed claim containing its location and time information and decides probabilistically whether to forward the received claim to the base station.

The base station computes the speed from every two consecutive claims of a mobile node and performs the SPRT by considering speed as an observed sample. The basic mission of the sensor network is already completely undermined if the base station is compromised. Each time mobile node's speed exceeds (respectively, remains below) V max, it will expedite the random walk to hit or cross the upper (respectively, lower) limit and thus lead to the base station accepting the alternate (respectively, null) hypothesis that the mobile node has been (respectively, not been) replicated. Once the base station decides that a mobile node has been replicated, it revokes the replica nodes from the network.

Algorithm 4
SPRT for replica detection
INITIALIZATION: $n = 0$, $!n = 0$
INPUT: location information L and time information T
OUTPUT: accept the hypothesis H_0 or H_1
cur loc = L
cur time = T
if $n > 0$ then
compute ω
0(n) and ω 1(n)
compute speed o from cur loc and prev loc, cur time
and prev time
if $o > Vmax$ then
$!n = !n + 1$
end if
if $!n >= ω$
1(n) then
accept the hypothesis H_1 and terminate the test
end if
if $!n <= ω$
0(n) then
initialize n and $!n$ to 0 and accept the hypothesis H_0
return;
end if
end if
$n = n + 1$
prev loc = cur loc
prev time = cur time

6. Advantages Of Using SPRT

- If the replicated node is moving much faster than any of the benign nodes, and thus the replica nodes measured speeds will often be over the system-configured maximum speed.
- Accordingly, if we observe that a mobile node's measured speed is over the system-configured maximum speed, it is then highly likely that at least two nodes with the same identity are present in the network.
- To minimize these false positives and false negatives, we apply the SPRT, a hypothesis testing method that can make decisions quickly and accurately.

- We perform the SPRT on every mobile node using a null hypothesis that the mobile node has not been replicated and an alternate hypothesis that it has been replicated.

- In using the SPRT, the occurrence of a speed that is less than or exceeds the system-configured maximum speed will lead to acceptance of the null or alternate hypotheses, respectively.

- Once the alternate hypothesis is accepted, the replica nodes will be revoked from the network. The main attack against the SPRT based scheme is when replica nodes fail to provide signed location and time information for speed measurement.

IV. CONCLUSIONS

A selfish node detection method and SCF tree replica allocation techniques to handle the selfish replica allocation. The traditional selfish node detection replica allocation techniques such as DCG, SAF and DAFN are failed to consider selfish nodes in terms of replica allocation. So we proposed a SCF-tree and a fast detection of SPRT. The programming language used to implement this process is java. To design an effective, fast, and robust replica detection scheme specifically for mobile sensor networks. By using the fact that an uncompromised mobile node should never move at speeds in excess of the system-configured maximum speed. The main advantage of the scheme is to quickly detect captured nodes with the aid of the SPRT.

REFERENCES

[1] Jae-Ho Choi Kyu-Sun Shim, Sang Keun Lee, and Kun-Lung Wu, Fellow, "Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network" IEEE Transaction on Mobile Computing, Voume 11, No.2, February 2012.

[2] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," IEEE International conference On Computer Communications (INFOCOM) , pp. 1568- 1576, 2001.

[3] T. Hara and S.K. Madria, "Data Replication for improving data accessibility in mobile ad-hoc networks" IEEE Transaction on Mobile Computing, Volume 5, No. 11, pp. 1515-1532, November 2006.

[4] S.U. Khan and I. Ahmad, "A Pure Nash Equilibrium-Based Game Theoretical Method for Data Replication across Multiple Servers," IEEE Transaction on Knowledge and Data Engineering., Volume. 21, No. 4, pp. 537-553, April. 2009.

[5] C.Balakrishnan "Mobile Replica Attack Detection Using Sequential Hypothesis Testing In Wireless Sensor Networks" International Journal of Communication Engineering Applications-IJCEAICICES-2012.

[6] M.siva Lakshmi, R.Raja Sekhar " ODRS: Optimal Data Replication Scheme for Time Efficiency In MANETs" IOSR Journal of Computer Engineering IOSRJCE) ISSN: 2278-0661 Volume 5, Issue 1 (Sep-Oct. 2012), PP 21-27

[7] Y. Yoo and D.P. Agrawal, "Why Does It Pay to be Selfish in a MANET," IEEE Wireless Communication., Volume. 13, No. 6, pp. 87-97, December. 2006.

[8] T. Hara and S.K. Madria, "Consistency Management Strategies for Data Replication in Mobile Ad Hoc Networks," IEEE Transaction on Mobile Computing, Voume l. 8, No. 7, pp. 950-967, July 2009.

[9] S.-Y. Wu and Y.-T. Chang, "A User-Centered Approach to Active Replica Management in Mobile Environments," IEEE Transaction on Mobile Computing, Volume 5, No. 11, pp. 1606-1619, November 2006.

[10] L. Yin and G. Cao, "Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks," Proceedings IEEE International Symposium. Reliable Distributed Systems, pp. 289-298, 2004.

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

[12] D. Hales, "From Selfish Nodes to Cooperative Networks -Emergent Link-Based Incentives in Peer-to-Peer Networks," IEEE International Conf. Peer-to-Peer Computing, pp. 151-158, 2004.

[13] Y. Liu and Y. Yang, "Reputation Propagation and Agreement in Mobile Ad-Hoc Networks," IEEE Wireless Communication. And Networking Conference. pp.1510-1515, 2003.

[14] M.Shorfuzzaman, P. Graham, R. Eskicioglu, Distributed popularity based replica placement in data grid environment, 2010 International conference on parallel and distributed computing applications and technologies pages 66-77, issue: 8-11 Dec. 2010.

[15] D.Nukarpu, B. Tang, L. wang, "Data replication in data Intensive scientific applications with performance guarantee", IEEE transactions on parallel and distributed systems, 2010.

[16] Y. Fang, L. Huo "Research of replication in unstructured p2p network" 2009 5th International Conference on wireless communications and networking, volume 33 issue 1.

[17] Q. Gu, B. Chen "Dynamic replica placement and location strategies for data grid", 2008 International Conference on Computer science and software engineering. 2008 IEEE, Issue: 1.1109/CSSE.2008

[18] Mr.A.Christopher paul, Mr.C.Chandru Vignesh .Dr.S.Karthik "Reputation-Based Storage Space Allocation for Balancing the Tradeoffs between Data Availability and Query Delay in MANET" International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 8, August 2012 ISSN: 2277 128X.

[19] Min Meng1, Xiaoling Wu1, Jie Yang1, Byeong-Soo Jeong2, Sungyoung Lee "Adjusted Replica Allocation in Ad Hoc Networks for Improving Data Accessibility"

[20] Rashid Azeem and Muhammad Ishfaq Ahmad Khan "Techniques About Data Replication For Mobile Ad-Hoc Network Databases International Journal Of Multidisciplinary Sciences And Engineering, Vol. 3, No. 5, May 2012 [Issn: 2045-7057].

[21] P. Mukilan, A. Wahi, "CDRA: Consistency based Data Replication Algorithm for MANET"International Journal of Computer Applications (0975 – 8887) Volume 51– No.14, August 2012.

[22] K.Santhi M.C.A., M.Phi l P.Bakeyalakshmi "Principal Component Analysis in Routing to Identify the Intrusion by Sequential Hypothesis

Testing"International Journal of Engineering and Science ISBN: 2319-6483, ISSN: 2278-4721, Vol. 1, Issue 8 (November 2012), PP 13-19.

[23] Chi-Anh La, Student Member, IEEE, Pietro Michiardi, Member, IEEE, Claudio Casetti, Member, IEEE, Carla-Fabiana Chiasserini, Senior Member, IEEE, and Marco Fiore, Member, IEEE"Content Replication in Mobile Networks"Journal On Selected Areas In Communications, Vol. X, No. X, Month 2010.

[24] Jun-Won Ho, Matthew Wright, Member, IEEE, and Sajal K. Das, Senior Member, IEEE "Fast Detection of Mobile Replica Node Attacks in Wireless Sensor Networks Using Sequential Hypothesis Testing" IEEE Transactions On Mobile Computing, Vol. 10, No. 6, June 2011.