Resource Controlled Replica Allocation Cluster for Secured Manet Against Selfish Nodes

Ms.P.Ramya[#1], Mr.S.Nandagopal M.E., (Ph.D).,[#2]
[#]M.E- CSE, Nandha College of Technology, Erode, India.
[#]Associate Professor, Nandha College of Technology, Erode, India.
[1]ramya.me2011@gmail.com
[2]asnandu@yahoo.com

***Abstract*- Mobile nodes in mobile ad hoc network have varied mobility and resource constraints that lead to network partitioning and performance degradation. Data replication techniques were used to minimize performance degradation that assumes all mobile nodes collaborate fully in sharing memory space. But certain nodes are selfish which cooperate partially or not at all. Selfish nodes reduce overall data accessibility.**

Existing work presented replica allocation model to evaluate impact of selfish nodes in MANET and selfish replica allocation. This develops selfish node detection algorithm considering partial selfishness and that cope with selfish replica allocation. However resource usage is increased due to selfish node identification. Dissimilarity in the replica allocation raises the complexity of node communication.

Proposal present Resource Controlled Replica Allocation Cluster (RCRAC) minimizes the resource consumption to secured ad hoc network against selfish nodes. Cluster the replica allocated nodes to improve the communication efficiency with minimal false detection of selfish nodes. This Control the resources of mobile node in evaluating security mode for ad hoc network. This also reduces improper communication between neighbor nodes.Simulations are conducted to demonstrate proposed approach to outperform traditional cooperative replica allocation techniques. Performance parameters are: Number of nodes, Selfish node density, communication overhead, average query delay, Energy consumption rate, Bandwidth capacity, Node Mobility.

Index terms: **Selfish nodes, replica allocation cluster, Selfish replica allocation.**

I. INTRODUCTION

MANETs have attracted high demand due to popularity of mobile devices and advances in wireless communication technologies. MANET is a peer-to-peer multihop mobile wireless network. Each node acts as a router and communicates with each other. MANET application areas are: battlefield or disaster area or mobile peer to peer file sharing system. Network partitions occur frequently due to mobility causing some data to be inaccessible to some other nodes.

Data are replicated at nodes other than original owners to increase data accessibility and to cope with frequent network partitions. Mobile nodes in mobile ad hoc network have varied mobility and resource constraints that lead to network partitioning and performance degradation. Data replication techniques were used to minimize performance degradation that assume all mobile nodes collaborate fully in sharing memory space but certain nodes are selfish that cooperate partially or not at all. Selfish nodes reduce overall data accessibility.

The existing system presented selfish node detection algorithm handling selfishness in the context of replica allocation in MANET. Selfish node did not share its own memory space to store replica to cooperate other nodes. The application area is peer- to-peer. Selfish replica allocation refers to node's non cooperative action when a node refuses to cooperate in sharing its memory space with other nodes. Partial selfish nodes are taken into account along with fully selfish nodes. We have to measure degree of selfishness to appropriately handle partially selfish nodes.

Partial selfishness used credit risk to detect selfish nodes. Node measure the degree of selfishness of another node to which it is connected by one or multiple hops in MANET. Replica allocation techniques with selfish node detection method was based on self-centered friendship tree (SCF-tree) and variation to achieve high data accessibility with low communication overhead. This reduces communication overhead and achieved good data accessibility.

But the existing system did not address the resource usage on securitizing MANET. The complexity of identifying selfish node increases. Node mobility needs more resource. There is redundant communication in the network. There is no indication of false alarm in

selfish replica allocation. To overcome all these drawbacks of existing selfish node detection algorithm, we propose a new scheme namely Resource Controlled Replica Allocation Cluster (RC-RAC) to control resource in detecting and selfish node in MANET.

To develop secured Mobile Ad Hoc Network (MANET) against selfish nodes, we present Resource Controlled Replica Allocation Cluster (RCRAC). We have to track resources of mobile nodes for both normal and selfish ones. Cluster the replica allocation of mobile nodes to use the network resources at an optimal point.

II. LITERATURE REVIEW

An extensive analysis of user traffic on Gnutella shows a significant amount of free riding in the system. Gnutella boasts a number of features that make it attractive to certain users. For example, Gnutella provides for anonymity by masking the identity of the peer that generated a query. Additionally, Gnutella provides the mechanism by which ad-hoc networks can be formed without central control. Since there are no central servers in the Gnutella network, in order to join the system a user initially connects to one of several known hosts that are almost always available (although these generally do not provide shared files). These hosts then forward the IP and port address information to other Gnutella peers [1].

Energy-efficiency is a must for routing protocols in ad hoc networks. However, energy-efficiency is only desirable from a global point of view, but not from the point of view of an individual and selfish node: if a network node gets chosen as an intermediate node with the duty of forwarding packets very often, the knowledge that it is on the most energy-efficient route is all but comforting since the forwarding actions drain its battery; the reasonable thing to do for this node is to play dead as soon as it realizes that its battery level keeps decreasing, thus simply refusing to forward messages. This non-cooperative behavior is a very basic problem in any ad hoc network in which the nodes are owned by different profit-maximizing entities [2].

Non-cooperative actions of misbehavior are usually termed as selfishness, which is notably different from malicious behavior. Selfish nodes use the network for their own communication, but simply refuse to cooperate in forwarding packets for other nodes in order to save battery power. A selfish node would thus utilize the benefits provided by the resources of other nodes, but will not make available its own resources to help others. They have no intention of damaging the network. Malicious nodes injected by adversaries, on the other hand, will actively spend battery power to cause harm to the entire network [3]. Each mobile node has one or more wireless network interfaces, with all interfaces of the same type (on dl mobile nodes) linked together by a single physical channel. When a network interface transmits a packet, it passes the packet to the appropriate physical channel object. This object then computes the propagation delay from the sender to every other interface on the channel and schedules a "packet reception" event for each. This event notifies the receiving interface that the first bit of a new packet has arrived [4].

Cooperative caching, in which multiple nodes share and coordinate cached data, is widely used to improve Web performance in wired networks. The "Related Work in Cooperative Caching" sidebar provides additional information about recent research focusing on cooperative caching approaches for wired networks. However, resource constraints and node mobility have limited the application of these techniques in ad hoc networks [5]. Servers may behave selfishly — seeking to maximize their own benefit. For example, parties in different administrative domains utilize their local resources (servers) to better support clients in their own domains. They have obvious incentives to cache objects that maximize the benefit in their domains, possibly at the expense of globally optimum behavior. It has been an open question whether these caching scenarios and protocols maintain their desirable global properties (low total social cost, for example) in the face of selfish behavior [6].

Messages, that can be broadcast or unicast, are labeled by a unique identifier and can be used by the recipient to detect where the message comes from. This feature allows replies to broadcast messages to be unicast when needed. To reduce network congestion, all the packets exchanged on the network are characterized by a given Time-To-Live (TTL). On passing through a node, the TTL of a forwarded message is decreased by one; when the TTL reaches zero, the message is dropped. The limit of the TTL creates a horizon of visibility for each node on the network [7]. The DHT-based protocols introduce complexity in implementation. In order to maintain the correctness of each routing table, peers should communicate to each other by some stabilization protocols periodically. These protocols should be triggered more frequently for MANET due to mobility in underlying physical networks. An additional neighbor table involving peers with the nearest keys may also be needed to improve the robustness [8].

The free-riding phenomenon is by no means unique to P2P systems. However, the characteristics of P2P systems present interesting challenges and opportunities for the design of incentive-compatible systems. Some of these characteristics include: lack of central authority, highly dynamic memberships, availability of cheap identities (\pseudonyms"), hidden or untraceable actions, and collusive behavior [9]. At a relocation period, a mobile host may not connect to another mobile host which has an original or a replica of a data item that the host should allocate. In this case, the memory space for the replica is temporary filled with one of replicas that have been allocated since the previous relocation period but are not currently selected for allocation [10].

III. RESOURCE CONTROLLED REPLICA ALLOCATION CLUSTER

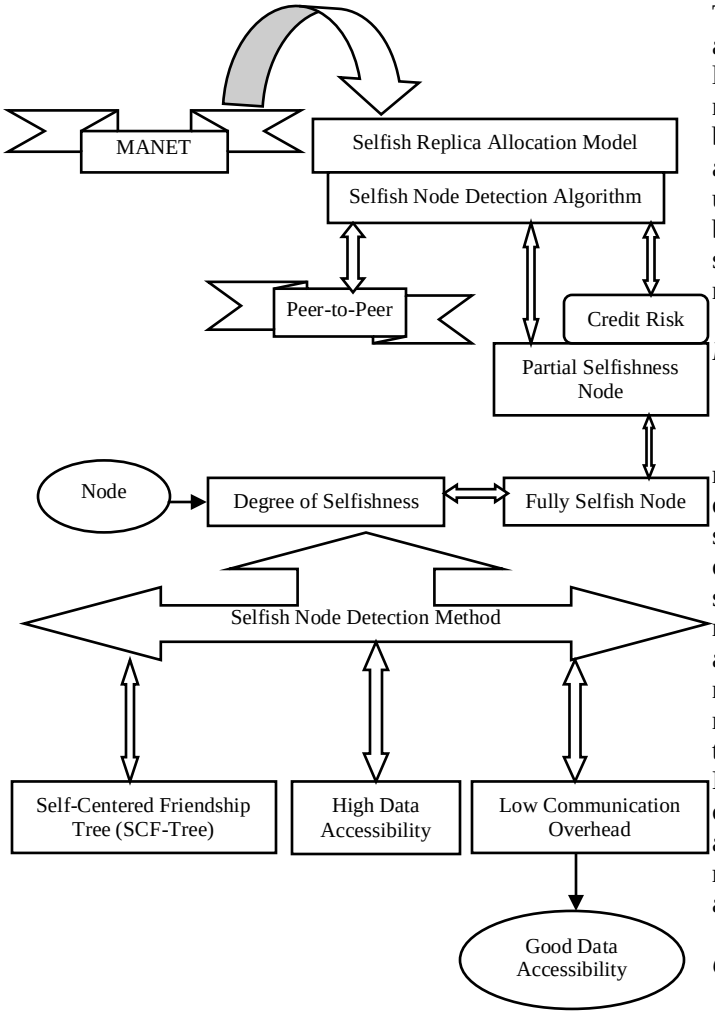

Fig.1 Architecture Diagram

The phases involved in the proposed scheme are:

- MANET and Mode Behavior Model
- Detecting Selfish Node and Self Centered Tree (SCF)
- Replica Allocation
- Resource Controlled SCF
- Clustered Replica Allocation

A. MANET and Node Behavior Model

In MANET each node has limited local memory space acts as data provider of several data items and data consumer. Each node holds replicas of data items to maintain replicas in local memory space. Replicas are relocated in a specific period. No central server determines allocation of replica. Three types of node behavioral states for selfish replica allocation. They are as follows, Type-1 node is a Non selfish node, and it holds replicas allocated by other nodes within limits of memory space. Type-2 node is a fully selfish nodes, it do not hold replicas allocated by other nodes but allocate replicas to other nodes for their accessibility. Type-3 node is a partially selfish nodes, it use their memory space partially for allocated replicas by other nodes. Memory space is divided logically into selfish and public area, nodes allocate replicas to other nodes for their accessibility

B. Detecting Selfish Nodes and Self Centered Tree (SCF)

Each node detects selfish nodes based on credit risk (CR) scores. CR score is updated accordingly during query processing phase to measure degree of selfishness. Node identifies, another node is believable or not replica is paid back or served upon request to share a memory space. Selfish features lead to selfish replica allocation problem determines expected value and expected risk. Selfish features are categorized into node specific and query processing-specific. Each node makes its own (partial) topology to builds its own SCF-tree. SCF represents relationships among nodes in MANET for replica allocation. SCF minimize communication overhead achieved high data accessibility. Each node detects selfishness for made replica allocation at its own discretion without forming any group or engaging in lengthy negotiations.

C. Replica Allocation

Based on SCF-tree each node allocates replica in a fully distributed manner. Node allocates replica at

every relocation period. Each node asks non selfish nodes within its SCF-tree to hold replica when it is unable to hold replica in its local memory space. Each node determines replica allocation individually without any communication with other nodes. Every node has its own SCF-tree and it perform replica allocation at its discretion. SCF-tree based replica allocation with degree of selfishness (SCF-DS) degree of selfishness in allocating replicas and less selfish nodes is visited first at the same SCF-tree level for more frequently accessed data items reside on less selfish nodes. SCF-tree based replica allocation with closer node (SCF-CN), allocates more replicas to the closer nodes in the SCF-tree more replicas allocated to node with lower depth within SCF-tree. Extended SCF-tree based replica allocation (eSCF), includes selfish nodes and non selfish nodes, marks detected selfish nodes within its eSCFtree, allocates replicas to the non selfish nodes in its eSCF-tree first and after first round, allocates replicas to all nodes within its eSCF-tree in round-robin manner minimal query delay.

D. Resource Controlled SCF

Resource utilized by mobile nodes on communication is measured. Effect of identifying selfish node replica allocation based on resources energy and bandwidth requirements. Replica allocation clusters are formed to evolve similarities of replica allocation in nodes.

E. Clustered Replica Allocation

Cluster index indicate the replica allocation nodes nature and capability. Energy consumption levels of replica allocation nodes indicate the nodes selfishness. Selfish nodes bandwidth requirements are higher compared to normal nodes. Clustered replica allocation nodes identify false information of selfish nodes. False alarms are generated to track selfish node mobility location inform corresponding neighbor about replica allocation status.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS RESOURCE CONTROLLED REPLICA ALLOCATION CLUSTER

In this section we evaluate performance of Resource Controlled Replica Allocation Cluster for Secured Mobile Ad Hoc Network against Selfish Nodes through NS2 simulation. One of the major contributions of this work is the replica allocation. To confirm the analytical results, we implemented Resource Controlled Replica Allocation in the MANET simulator ns-2 and evaluated the performance of technique.

The performance of Resource Controlled Replica Allocation is evaluated by the following metrics.

1. Resource Control Rate
2. Energy Consumption
3. False Alarm Rate

TABLE I
RESOURCE CONTROL RATE

Number of Mobility	Existing Selfish Node Detection Algorithm	Proposed Resource Controlled Replica Allocation Cluster
20	7	5
40	6.5	4.5
60	5.3	4
80	3	2.2
100	2	1

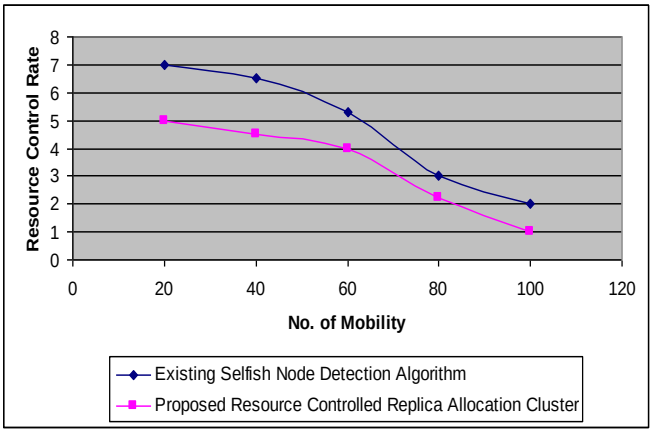

Fig.2 Resource Control Rate

Figure 2 demonstrates the resource control rate. X axis represents the number of mobility whereas Y axis denotes the resource control rate using both the Selfish Node Detection Algorithm and our proposed Resource Controlled Replica Allocation Cluster. When the number of mobility increased, resource control rate gets decreases accordingly. The rate of resource control is illustrated using the existing the Selfish Node Detection Algorithm and proposed Resource Controlled Replica Allocation Cluster. Figure 2 shows better performance of Proposed Resource Controlled Replica Allocation Cluster in terms of mobility than existing Selfish Node Detection Algorithm and proposed

Resource Controlled Replica Allocation Cluster. Resource Controlled Replica Allocation Cluster achieves 5 to 15% less resource control rate variation when compared with existing system.

TABLE II
ENERGY CONSUMPTION

Number of Mobility	Existing Selfish Node Detection Algorithm	Proposed Resource Controlled Replica Allocation Cluster
20	2	1
40	4	2
60	6	3
80	8	4
100	9	5

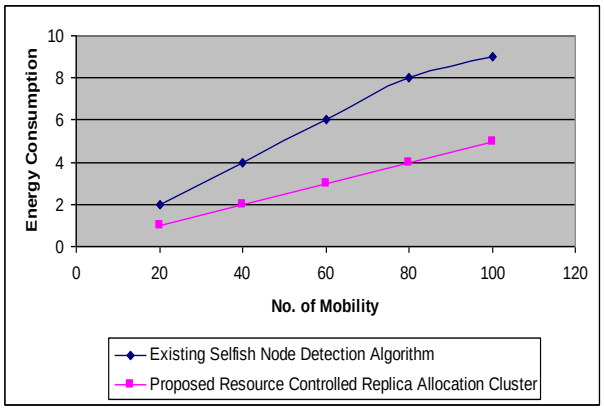

Fig.3 Energy Consumption

Figure 3 demonstrates the energy consumption. X axis represents number of mobility whereas Y axis denotes the energy consumption using both the Selfish Node Detection Algorithm and our proposed Resource Controlled Replica Allocation Cluster. When the number of mobility increased energy consumption also gets increased. Figure 3 shows the effectiveness of energy consumption over different number of mobility than existing Selfish Node Detection Algorithm and proposed Resource Controlled Replica Allocation Cluster. Resource Controlled Replica Allocation Cluster achieves 20% to 30% more energy consumption when compared with existing schemes.

Figure 4 demonstrates the false alarm rate. X axis represents number of mobility whereas Y axis denotes the false alarm rate using both the Selfish Node Detection Algorithm and our proposed Resource Controlled Replica Allocation Cluster.

TABLE III
FALSE ALARM RATE

Number of Mobility	Existing Selfish Node Detection Algorithm	Proposed Resource Controlled Replica Allocation Cluster
20	2.3	1.3
40	3.5	2
60	4.6	3.2
80	5.6	4
100	6.7	5.1

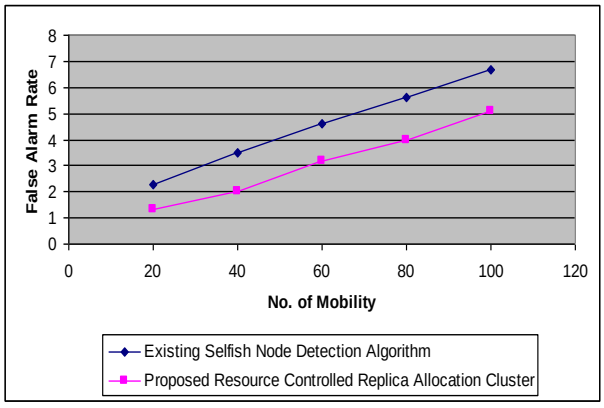

Fig 4: False Alarm Rate

When the number of mobility increased false alarm rate also gets increased. Figure 4 shows the effectiveness of false alarm rate over different number of mobility's than existing Selfish Node Detection Algorithm and proposed Resource Controlled Replica Allocation Cluster. Resource Controlled Replica Allocation Cluster achieve 30% to 50% more false alarm rate when compared with existing schemes.

V. CONCLUSION

We have proposed Resource Controlled Replica Allocation Cluster (RC-RAC) scheme to control resource in detecting and selfish node in MANET. This is also 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.

This proposed control resource usage in the ad hoc network. The Energy consumption and bandwidth requirements are reduced in identifying selfish node replica allocation. This scheme also reduces complexity of communication trustiness between mobile nodes. This has the indication of false alarms generated by selfish mobile nodes and so eventually, overall network resource utilization gets minimized

VI. REFERENCES

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

[2] 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.

[3] 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.

[4] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," Proc. ACM MobiCom, pp. 85-97, 1998.

[5] G. Cao, L. Yin, and C.R. Das, "Cooperative Cache-Based Data Access in Ad Hoc Networks," Computer, vol. 37, no. 2, pp. 32-39, Feb. 2004.

[6] 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.

[7] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Managing and Sharing Servents' Reputations in P2P Systems," IEEE Trans. Knowledge and Data Eng., vol. 15, no. 4, pp. 840-854, July/Aug. 2003.

[8] 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.

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

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

