ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

MANAGING SELFISH NODES THROUGH NODE COOPERATIVE INCENTIVES IN MANETS

V.Pavani[1], T.Rama Mohan[2], A.Vijay Kumar[3]

M.Tech. Student, Dept. of IT, GATES Institute of Technology, Gooty, Andra Pradesh, India [1]

Assistant professor, Dept. of IT, GATES Institute of Technology, Gooty, Andra Pradesh, India [2]

Associate Professor, Department of IT, NMREC, Hyderabad, Andra Pradesh, India [3]

ABSTRACT:Mobile ad hoc networks are low in resources. This constraint leads to problems such as performance degradation or network partition. In order to overcome this problem data is replicated in mobile nodes. The replicas ensure that the performance is not degraded. Towards this all nodes are expect to work together and share the available memory. However, chances are that some nodes may not cooperate and rather behave selfishly. The selfishness of some of the nodes may lead to the performance degradation in terms of accessing data. Many replica allocation techniques came into existence. This paper implements that for detecting selfish nodes and replication allocation using simulation tool NS2. The empirical results revealed that the proposed algorithm I effective in terms of communication cost, data accessibility and average query delay.

Keywords: Selfish replica allocation, mobile ad hoc networks, selfishness identification.

I.INTRODUCTION

Mobile devices have become popular as they can be used in computing. Mobile ad hoc networks are widely used as they are infrastructure less and can be established as and when required. This is made possible due to the innovations in communication technologies [1], [2], [3]. MANET is a multihop network sans a central server and infrastructure. However each node in the MANET works as a router and maintain communication with other nodes. There are many MANET applications in the world [4]. For example, MANET can be established in situations such as natural disasters, battle fields etc. Therefore MANET is suitable to address the problems in communication in emergency situations. Another important and interesting MANET application is a peer-to-peer mobile overlay network that helps in file sharing [5], [6]. As nodes in the MANET can move from one place to another that may cause network partitions. For this reason data present in one node may become inaccessible when it moves from one network partition to another network partition. The data accessibility is the measure used to know the level of accessibility in MANETs [7]. In order to improve the accessibility of data, the data present in the owner node is replicated in other nodes as well [7], [1], [8]. Selfish replica allocation is shown in fig. 1.

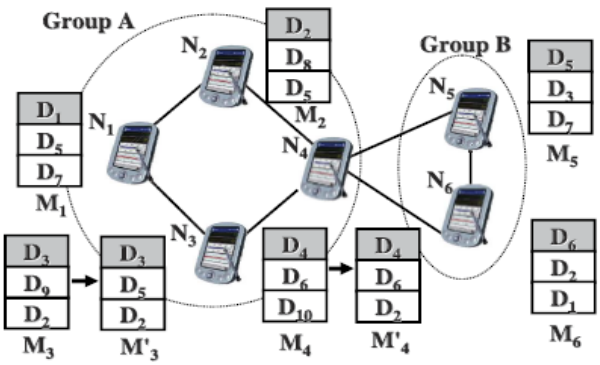

Fig. 1 – Selfish replica allocation

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

In figure 1 there are many nodes in MANET connected via wireless network. It shows an existing replication allocation scheme (DCG) presented in [7]. The straight line between nodes indicate wireless link; an item present in greycolored rectangle denotes original data; replication allocations are denoted by white boxes. The access frequencies of the data by nodes are presented in table 1.

Table 1 –Access frequency of nodes

Data	**Nodes**					
	N1	N2	N3	N4	N5	N6
D1	0.65	0.25	0.17	0.22	0.31	0.24
D2	0.44	0.62	0.41	0.40	0.42	0.46
D3	0.35	0.44	0.50	0.25	0.35	0.37
D4	0.31	0.15	0.10	0.60	0.09	0.10
D5	0.51	0.41	0.43	0.38	0.71	0.20
D6	0.08	0.07	0.05	0.15	0.20	0.62
D7	0.38	0.32	0.37	0.33	0.40	0.32
D8	0.22	0.33	0.21	0.23	0.24	0.17
D9	0.18	0.16	0.19	0.17	0.24	0.21
D10	0.09	0.08	0.06	0.11	0.12	0.09

In table 1, the greycolored items are frequently accessed items by node 3 and 4. The DCG employs a technique to minimize duplication allocation in any given group of nodes in the MANET. As seen in fig. 1, when node 3 behaves selfishly, the other nodes in the same group can't access data properly causing delay in data access.

The replication techniques are useful to reduce query delay, increase response time and improve overall data accessibility. As the nodes in the MANET have limited resources, the nodes may not participate in the replication process completely. They may exhibit selfish behavior in order to save and enjoy resources that cause problems in data accessibility [9], [10].

II.RELATED WORK

This section provides review of literature pertaining to MANETs and replication allocation strategies. Basically MANETs are of two types namely open and closed as described in [11], [12] and [13]. In case of closed MANET, the nodes behave as expected and all the nodes in the network have similar objectives. In case of open MANET, the nodes are guaranteed to have similar behavior. There are many techniques to handle misbehavior of nodes in the MANET [13]. This paper also considered open MANET. All the techniques that came into existence in order to handle selfish behavior of node are classified into two types. They are credit-payment based, reputation based, and game theory based. In case of reputation – based techniques, a node in MANET observes the behaviors of other nodes in order to make routing decisions [14], [15]. In case of credit payment techniques, credits concept is used to reward the nodes that

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

behave well in the MENT with respect to data forwarding [9], [16]. Whereas the game theory based techniques assume that the nods can have their own strategies to maximize profits [10], [17].

All the techniques discussed above make use of packet forwarding concept for communication. However, this paper focuses on selfish replica allocation. Many trust models were introduced by [18]. They see misbehavior in terms of refusing to forward data and dropping packets. These traditional techniques can't be directly applied to replication allocation. The reason is that they make binary decisions such as selfish or not and can't deal with partially selfish nodes.

In the prior work [7] effective techniques were proposed for replication allocation. Highest data accessibility was reported with such techniques like DCG. However, DCG can't handle selfish nodes in MANET. Both data accessibility and query delay problems were addressed by the work in [8]. It shows trade-off to exhibit balance in the technique.

Cooperative caching-based data accessing was introduced in [19]. The methods in it include Cache Data, Cache Path, and Hybrid. In this paper replication allocation technique [20] in MANET containing selfish nodes is implemented practically using NS2.

III.PROPOSED REPLICA ALLOCATION STRATEGY

This section provides information about the proposed replica allocation strategy and algorithms. There are three parts in the strategy. They are selfish node detection, SCF-tree building, and replica allocation. Credit risk scores concept is used to detect selfish nodes. Every node has the ability to detect selfish nodes. Algorithm for detecting selfish nodes is as given in listing 1.

```
Algorithm 1. Pseudo code to detect selfish nodes
00: At every relocation period
01: /* node Ni detects selfish nodes with this algorithm */
02: detection (){
03: for (each connected node Nk ){
04: if (nCRki<δ)Nk is marked as non-selfish;
05: else Nk is marked as selfish;}
06: waits until replica allocation is done;
07: for (each connected node Nk){
08: if (Ni has allocated replica to Nk){
09 NDki = the number of allocated replica;
10: SSki = the total size of allocated replica;}
11: else{
12: NDki = 1;
13: SSki = the size of a data item;
14: } } }
```

Listing 1 Algorithm to detect selfish nodes

Every node runs this algorithm in order to detect selfish nodes. At each relocation period, the node finds selfishness of each other connected node based on the credit risk concept. The algorithm to update selfish features is presented in listing 2.

```
Algorithm 2. Pseudo code to update selfish features
00: At every query processing time
01: /*When Ni issues a query */
02: update_SF(){
03: while (during the predefined time !){
04: if (an expected node Nk serves the query)
05: decrease Pki;
06: if (an unexpected node Nj serves the query){
07: NDji = NDji + 1;
08: SSji = SSji + (the size of a data item);
```

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

```
09: } }
10: if (an expected node Nk does not serve the query){
11: increase Pki ;
12: NDki = NDki - 1;
13: SSki = SSki - (the size of a data item);
14: } }
```

Listing 2 Updating selfish feature

In listing 2, at each query processing time, the credibility of other nodes is computed and based on the selfishness of the nodes is updated. Building SCF tree is carried out using the algorithm 3 presented in listing 3.

```
Algorithm 3. Pseudo code to build SCF-tree
00:/* node Ni makes SCF-tree with a parameter of depth d*/
01: constructScfTree(){
02: append Ni to SCF-tree as  root node;
03: checkChildnodesðNiÞ;
04: return SCF-tree;}
05: Procedure checkChildnodesðNjÞ{
06: /*INajis a set of nodes that are adjacent nodes to Nj*/
07: for (each node Na 2 INaj){
08: if (distance between Na and the root > d )
09: continue;
10: else if ðNa is an ancestor of Nj in TSCFi)
11: continue;
12: else{ append Na to TSCFi as a child of Nj;
13: checkChildnodes(Na); } } }
```

Listing 3 Algorithm for building SCF tree

In listing 3, the algorithm is meant for building SCF tree which is further used in replication allocation effectively. Algorithm 4 is presented in listing 4 which is meant for replica allocation.

```
Algorithm 4. Pseudo code for replica allocation
00: /_Ni executes this algorithm at relocation period _/
01: replica_allocation(){
02: Li = make priority(TSCFi );
03: for (each data item 2 IDi){
04: if (Ms is not full)
05: allocate replica of the data to Ms ;
06: else{/_Ms is full _/
07: allocate replica of the data to  target node;
08: /_ the target node is selected from the Li
09: if (Mp is not full)
10: allocate replica of  data to Mp; } }
11: while (during a relocation period){
12: if (Nk requests for the allocation of Dq)
13: replica_allocation_for_others (Nk;Dq); } }
14: Procedure make_priorityðTSCFi ) {
15: for (all vertices in TSCFi ){
16: select a vertex in TSCFi in order of BFS;
17: append the selected vertex id to the Li }
18: return Li; }
19: Procedure replica_allocation_for_others(Nk;Dq){
```

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

```
20: if (Nk is in TSCFi and Ni does not hold Dq){
21: if (Mp is not full) allocate Dq to Mp ;
22: else{=-Mp is full */
23: if(Ni holds any replicas of  local interest in Mp)
24: replace the replica with Dq ;
25: else{
26: =-Nh is the node with the highest nCRhiamong the nodes which allocated replica to Mp
27: if (nCRhi>nCRki)
28: replace the replica requested by Nh with Dq;
29: } } } }
```

Listing 4 Algorithm for replication allocation

In listing 4, the replica allocation algorithm identifies all the data items that are to be replicated. At each node replicas are allocated in descending order by node's access frequency. First of all breadth first search is used to know priorities and then replicas are allocated based on the priorities.

IV.EXPERIMENTAL RESULTS

Experiments are made in a PC with 3GB of RAM and Core 2 Dual processor. The simulations are tested using NS2. MANET creation and selfish node identification and handling selfish nodes while allocating replicas are shown in simulations. The parameters considered for simulations are as shown in table 2

Table 2 – Simulation Parameters.

Parameter(unit)	Value(default)
No.of nodes	40
No.of data items	40
Radius of communication range	1~19(7)
Size of network	50*50
Size of memory space	2~40(10)
Percentage of selfish nodes	0~100(70)
Maximum velocity of nodes	1
Relocation period	64~8,192(256)
Zipf parameter	0.8

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

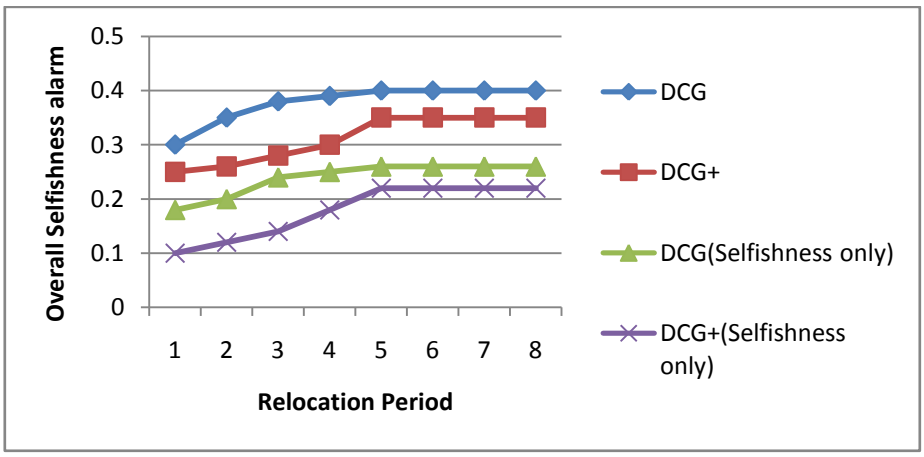

Fig. 2 Relocation period vs. overall selfishness alarm

In fig. 2, the horizontal axis represents relocation period while the vertical axis shows overall selfishness alarm. The results reveal that overall selfishness alarm of DCG+ shows very less selfishness alarm

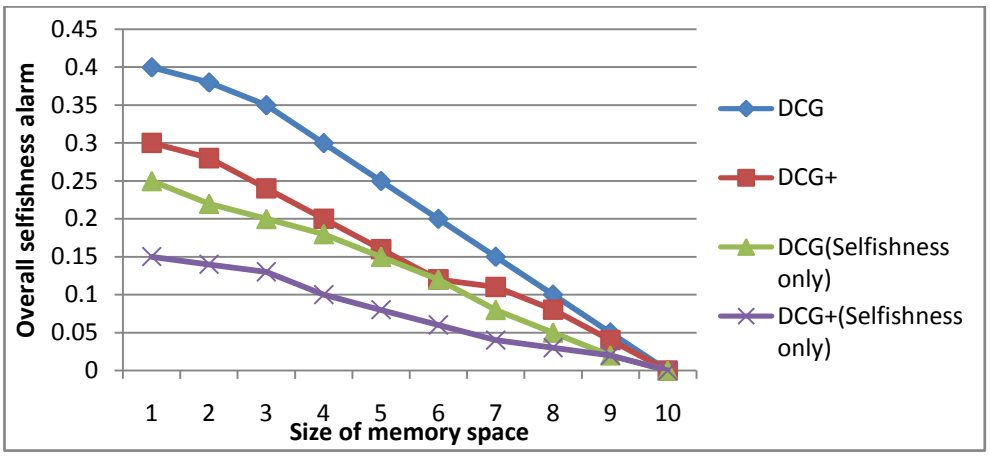

Fig. 3 Size of memory space vs. overall selfishness alarm

In fig. 3 the horizontal axis represents size of memory space while the vertical axis shows overall selfishness alarm. The results reveal that overall selfishness alarm of DCG+ shows very less selfishness alarm.

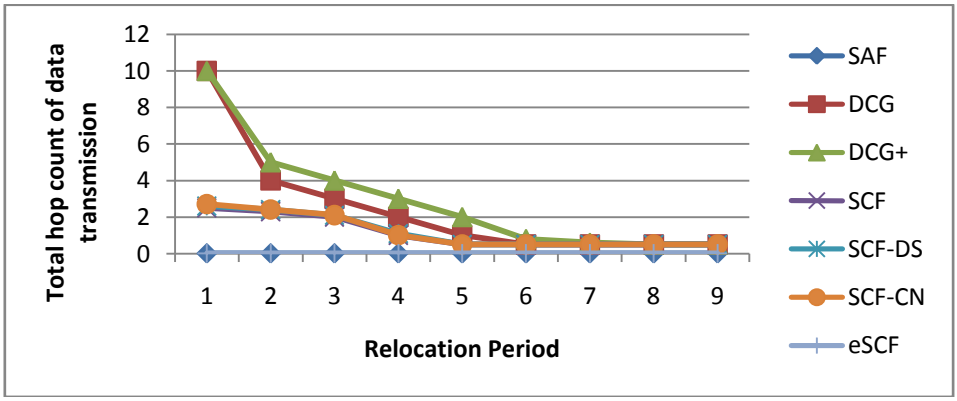

Fig. 4 Varying relocation period vs. communication cost

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

In fig. 4, the horizontal axis represents relocation period while the vertical axis shows total hop count of data.

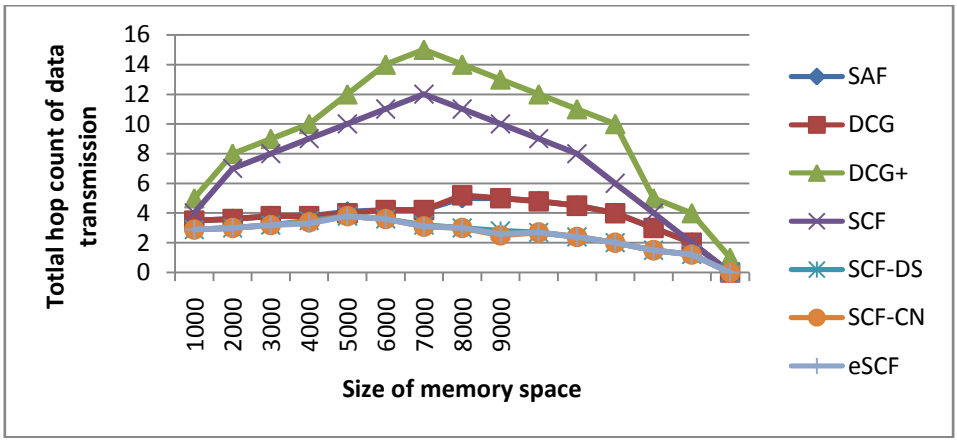

Fig. 5 Varying size of memory space vs. communication cost

In fig. 5, the horizontal axis represents size of memory space while the vertical axis shows total hop count of data. The results reveal that the communication cost increase as size of memory space increases. However after some size, it starts reducing. The reason for this behavior is that after certain size of memory, each node contains replicas of many items. Therefore relocation of replicas occur rarely. Nevertheless, communication cost of SAF is very less when compared with other techniques.

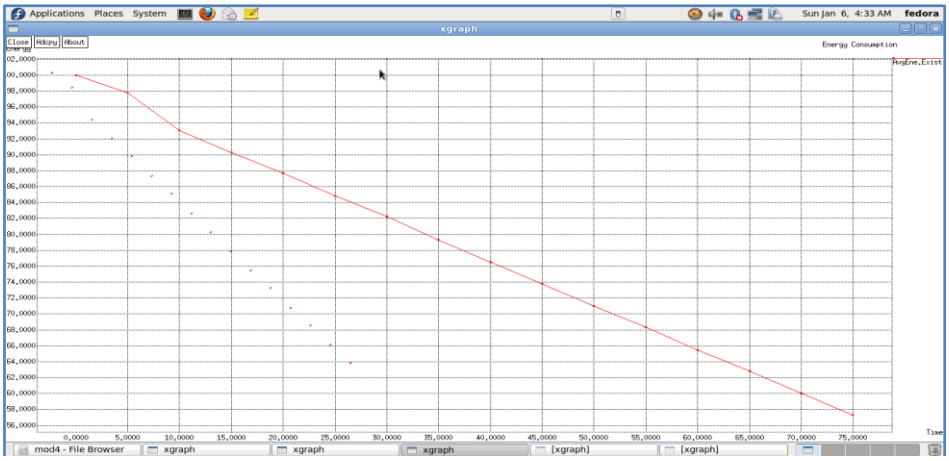

Fig. 6 Time vs. Energy Consumption

From fig. 6, it is evident that the graph plots 2 values. The X axis shows the time in milliseconds while the vertical access represents energy consumption. As the time increases, the energy consumption decreases. It does mean that once handling selfish nodes is started, the energy consumption started decreasing.

V. CONCLUSION

In this paper, as against to network centric solution, we considered selfish nodes in the MANET and addressed the problem of selfish behavior in replica allocation. The selfish replica allocation is the problem which leads to reduced data accessibility. We have implemented the algorithm proposed by Choi et al. using NS2. The algorithm covers both identification of selfish nodes and also handling them for good replication allocation. The simulation results reveal that the algorithm is capable of reducing delay in response, improve data accessibility and overall performance of the network.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

REFERENCES

[1] T. Hara and S.K. Madria, "Data Replication for Improving Data Accessibility in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 11, pp. 1515-1532, Nov. 2006.
[2] T. Hara and S.K. Madria, "Consistency Management Strategies for Data Replication in Mobile Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 8, no. 7, pp. 950-967, July 2009.
[3] S.-Y. Wu and Y.-T. Chang, "A User-Centered Approach to Active Replica Management in Mobile Environments," IEEE Trans. Mobile Computing, vol. 5, no. 11, pp. 1606-1619, Nov. 2006.
[4] P. Padmanabhan, L. Gruenwald, A. Vallur, and M. Atiquzzaman, "A Survey of Data Replication Techniques for Mobile Ad Hoc Network Databases," The Int'l J. Very Large Data Bases, vol. 17, no. 5, pp. 1143-1164, 2008.
[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. Li, W.-C. Lee, and A. Sivasubramaniam, "Efficient Peer-to-Peer Information Sharing over Mobile Ad Hoc Networks," Proc. World Wide Web (WWW) Workshop Emerging Applications for Wireless and Mobile Access, pp. 2-6, 2004.
[7] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," Proc. IEEE INFOCOM, pp. 1568- 1576, 2001.
[8] L. Yin and G. Cao, "Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks," Proc. IEEE Int'lSymp. Reliable Distributed Systems, pp. 289-298, 2004.
[9] 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.
[10] 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.
[11] 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.
[12] H. Miranda and L. Rodrigues, "Friends and Foes: Preventing Selfishness in Open Mobile Ad hoc Networks," Proc. IEEE Int'l Conf. Distributed Computing Systems Workshops, pp. 440-445, 2003.
[13] Y. Yoo and D.P. Agrawal, "Why Does It Pay to be Selfish in a MANET," IEEE Wireless Comm., vol. 13, no. 6, pp. 87-97, Dec. 2006.
[14] Y. Liu and Y. Yang, "Reputation Propagation and Agreement in Mobile Ad-Hoc Networks," Proc. IEEE Wireless Comm. and Networking Conf., pp. 1510-1515, 2003.
[15] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," Proc. ACM MobiCom, pp. 255-265, 2000.
[16] W. Wang, X.-Y. Li, and Y. Wang, "Truthful Multicast Routing in Selfish Wireless Networks," Proc. ACM MobiCom, pp. 245-259, 2004.
[17] V. Srinivasan, P. Nuggehalli, C. Chiasserini, and R. Rao, "Cooperation in Wireless Ad Hoc Networks," Proc. IEEE Infocom, pp. 808-817, 2003.
[18] H. Li and M. Singhal, "Trust Management in Distributed Systems," Computer, vol. 40, no. 2, pp. 45-53, Feb. 2007.
[19] 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.
[20] Jae-Ho Choi, Kyu-Sun Shim, SangKeun Lee, and Kun-Lung Wu, Fellow, IEEE, "Handling Selfishness in Replica Allocationover a Mobile Ad Hoc Network", IEEE Transactions on mobile computing, vol. 11, no. 2, Feb. 2012.