Volume 3, Issue 4, April-2016, pp. 165-168 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Handling Selfishness in Replica Allocation over a Mobile Ad-Hoc Network

Laxmi R. Shinde, Trupti A. Jadhav, Prajakta R. Baviskar

Abstract:- MANET is a collection of mobile devices that can communicate with each other without the use centralized administration. One of the interesting application of MANET is File Sharing. File Sharing in MANET is similar to that of the regular file sharing, what makes the difference is it allow user to access the data or memory of that nodes only which are connected to it. This File sharing many a times leads to Network Partitioning, i.e dividing a network into two different networks .Due to which the nodes may act selfishly. The selfishness of some of the nodes may lead in reduction of performance in terms of accessing data. The proposed system will use the SCF-tree technique for building a tree of Node which will share their data in terms of Replica, and as a result it detects the selfish node in the network. The replica insures that performance is not degraded.

Keywords – Replica allocation, Selfish node, Network partitioning.

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 [10], [11], [7]. A MANET is a peer-to-peer multi hop mobile wireless network that has neither a fixed infrastructure nor a central server. Each node in a MANET acts as a router, and communicates with each other. A large variety of MANET applications have been developed. For example, a MANET can be used in special situations, where installing infrastructure may be difficult, or even infeasible, such as a battlefield or a disaster area. A mobile peer-to-peer file sharing system is another interesting MANET application [12].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 [9]. 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 [9] [10]. In general, replication can simultaneously improve data accessibility and reduce query delay, i.e., query response time, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replicas and the original data. For example, the response time of a query can be substantially reduced, if the query accesses a data item that has a locally stored replica. However, there is often a trade-off between data accessibility and query delay, since most nodes in a MANET have only limited memory space. For example, a node may hold a part of the frequently accessed data items locally to reduce its own query delay. However, if there is only limited memory space and many of the nodes hold the same replica locally, then some data items would be replaced and missing. Thus, the overall data accessibility would be decreased. Hence, to maximize data accessibility, a node should not hold the same replica that is also held by many other nodes. However, this will increase its own query

delay.A node may act selfishly, i.e., using its limited resource only for its own benefit, since each node in a MANET has resource constraints, such as battery and storage limitations. 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. Existing research on selfish behaviors in a MANET mostly focus on network issues [2], [8], [6]. For example, selfish nodes may not transmit data to others to conserve their own batteries. Although network issues are important in a MANET, replica allocation is also crucial, since the ultimate goal of using a MANET is to provide data services to users.

2. RELATED WORK

Takahiro Hara stated the three replica allocation methods that are used to improve data accessibility by replicating data items on mobile hosts. In these three methods, the access frequency from mobile hosts to each data item and the status of the network connection is taken into consideration. Jaydip Sen1 and Kaustav Goswami2 presented an Algorithm for detection of selfish nodes in a WMN(Wireless Mesh network).Wireless mesh networks (WMNs) are evolving as a key technology for next-generation wireless networks showing raid progress and numerous applications .However the throughput of a WMN may be severely degraded due to presence of some selfish routers therefore this paper introduces the use of statistical theory of inference for reliable clustering of the nodes and is based on local observations by the nodes. Jim Solomon Raja.D, Immanuel John Raja.J gave a new mechanism that minimizes the problem of selfish nodes with the help of Credit risk and Brain trapping function Model. Including Degree of selfishness in allocating replicas will considerably reduce communication cost and produce high data accessibility. A collaborative monitoring mechanism is also used to manage false alarms. Simulation results shows that the proposed system provides better detection efficiency, low false positive and delay constraint. M. Manjula M.C.A, P. Elango MCA, examined the impact of selfish nodes in a mobile ad hoc network from the perspective of replica allocation. This work was motivated by the fact that a selfish replica allocation could lead to overall poor MANET data accessibility. The strategies are inspired by the real-world observations in economics in terms of credit risk and in human friendship management in terms of choosing ones friends completely at ones own circumspection. The notion of credit risk from economics is to detect nodes that behave selfishly. Each and every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness.

3. PROPOSED WORK

In a proposed system at a specific period, or relocation period, each node executes the following procedures:-
-Each node detects the selfish nodes based on credit risk scores.
-Each node makes its own (partial) topology graph and builds its own SCF-tree by excluding selfish nodes.
-Based on SCF-tree, each node allocates replica in a fully distributed manner.

3.1. ALGORITHM TO DETECT SELFISH NODE:

00: At every relocation period
01: /* Ni detects selfish nodes with this algorithm */
02: detection(){
03: for (each connected node N_k){
04: if($nCR_i^k < \alpha$)N_k is marked as non-selfish;
05: else N_k is marked as selfish;}
06: wait until replica allocation is done;
07: for (each connected node N_k){
08: if(N_i has allocated replica to N_k){
09 ND_i^k =the number of allocated replica;
10: SS_i^k = the total size of allocated replica;}
11: else{
12: ND_i^k = 1;
13: SS_i^k = the size of a data item;
14: } } }

3.2. ALGORITHM TO UPDATE SELFISH FEATURE

00: At every query processing time
01: /* When Ni issues a query */
02: update_SF () {
03: while (during the predefined time ω){
04: if (an expected node Nk serves the query)

05: decrease P^{k}_{i} ;
06: if (an unexpected node N_k serves the query){
07: ND^{j}_{i}= ND^{j}_{i} + 1;
08: SS_i= SS_i+ (the size of a data item);
09: } }
10: if (an expected node N_k does not serve the query){
11: increase P^{k}_{i};
12: ND^{k}_{i} = ND^{k}_{i}- 1;
13: SS^{k}_{i} = SS^{k}_{i} (the size of a data item);
14: } }

3.3. ALGORITHM TO BUILD SCF-TREE

00:/* Ni makes SCF-tree with a parameter , depth d*/
01: constructScfTree(){
02: append Ni to SCF-tree as the root node;
03: checkChildnodes (N_i);
04: return SCF-tree;}
05: Procedure checkChildnodes (N_j){
06: /* $IN_{aj}$is a set of nodes that are adjacent nodes to Nj */
07: for (each node N_a ЄIN_{aj}){
08: if (distance between Na and the root >d)
09: continue;
10: else if (N_a is an ancestor of Njin T_i^{scf})
11: continue;
12: else{ append N_a to in T_i^{scf}as a child of N_j;
13: checkChildnodes(N_a); } } }

3.4. ALGORITHM TO ALLOCATE REPLICA

00: /* N_i executes this algorithm at relocation period */
01: replica_allocation(){
02: Li = make_priority(T_i^{SCF}) ;
03: for (each data item Є 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 the target node;
08: /* the target node is selected from Li */
09: if (Mp is not full)
10: allocate replica of the 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_priority (T_i^{SCF}) {
15: for (all vertices in T_i^{SCF}){
16: select a vertex in T_i^{SCF} in order of BFS;
17: append the selected vertex id to Li; }
18: return Li ; }
19: Procedure replica_allocation_for_others(Nk ,Dq){
20: if (Nk is in $T_i^{SCF}$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 replica of local interest in Mp)
24: replace the replica with Dq ;
25: else{
26: /* Nh is the node with the highest nCR^{h}_{i} among the nodes which allocated replica to Mp */
27: if (nCR^{h}_{i}>nCR^{k}_{i})
28: replace the replica requested by Nh with Dq;
29: } } } }

4. CONCLUSION

MANETS are used in various contexts like mobile social networks, emergency deployment; intelligent transportation systems etc. According to the viewpoint of network, problem of selfish nodes from the replica allocation has been addressed. This term is stated as selfish replica allocation. The fact that a selfish replica allocation could lead to overall poor data accessibility in a MANET, so the proposed solution describes a selfish node detection method and 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 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.

5. FUTURE SCOPE

A selfish node is one that tries to utilize the network using its limited resource only for its own benefit, since each node in a MANET has resource constraints, such as battery and storage limitations, it 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. The research is currently going on the impact of different mobility patterns. The proposed system improves the data accessibility, reduces communication cost, and average query delay and also to reduce the detection time of the selfish nodes.

REFERENCES

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

[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] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

[13] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," Proc. ACM MobiCom, pp. 255-265, 2000.

[14] 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.

