ISSN 2321-807X
Volume 12 Number22
Journal of Advances in chemistry

PREVENTING SELFISHNESS DATA ACCESS IN WIRELESS NETWORK

Dr. M.Vinoth Kumar[1], Dr. G.Tholkappia Arasu[2]
[1.2]Professor
drvinojimail@gmail.com
[1]Department of CSE, A.V.S Engineering College, Salem, Tamil Nadu, India
[2]Principal, A.V.S College of Technology, Salem, Tamil Nadu, India

ABSTRACT

A mobile adhoc network (MANET) is a wireless network among mobile device. A mobile adhoc network (MANET) is a constantly self-configuring, infrastructure-less network of mobile devices connected wirelessly. Each device in a MANET is free to move independently in any direction, and will therefore modify its relations to further devices regularly .Selfish assign nodes from direct based on simulation result analysis on MANET. Mobile ad-hoc networks (MANETs) rely on collaboration of all contribute nodes. Thus they are vulnerable to self-interested nodes using the net without providing own resources, as well as hateful nodes attacking the network communications.

Keywords: Mobile Networks, Selfishness Node, MANET.

1. INTRODUCTION

The peer-to-peer (P2P) network and mobile ad hoc network (MANET) had obtained popularity in the field of research. Mobile adhoc networks have concerned a lot of consideration due to the popularity of mobile devices and the advances in wireless communication [2]. MANETs are used in many contexts such as in mobile social networks, emergency consumption, intellectual carrying systems etc. Nodes in a MANET liberally move around while communicating with each other. These networks may carry out in the occurrence of nodes with a self-centered behavior mainly when working under energy constraints.

A MANET is a peer-to-peer multihop [2] mobile wireless network that has neither a enduring infrastructure nor a central server. Each node in a MANET acts as a router, and communicates with each other. A large selection of MANET applications has been residential. For example, a MANET can be used in special situations, where installing infrastructure may be hard, or even infeasible, such as a battlefield or a disaster area. A mobile peer-to-peer file input system is another smart MANET application [9], [10] [11]. Network partitions can take place commonly, since nodes shift freely in a MANET, causing some data to be often difficult to get to to some of the nodes. Hence, data accessibility is often a major presentation metric in a MANET [7].

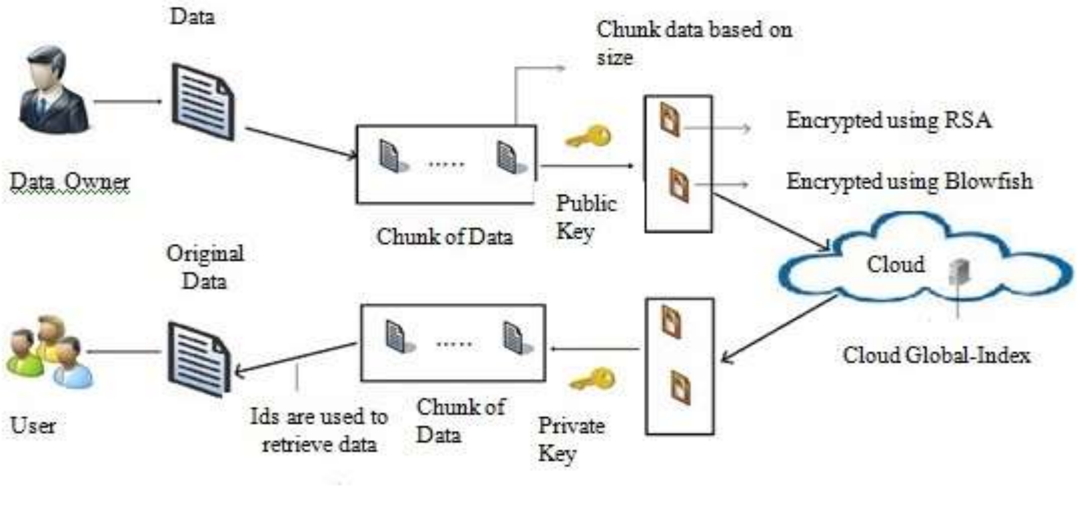

Fig.1 : A Model of Selfish Replica Allocation

In this paper, we assume that each node has limited local memory space and acts as a data provider of several data items and a data user. Each node holds replicas of data items, and maintains the replicas in local memory space. The replicas are relocated in a specific period. Any node freely joins and organizes an open MANET. We model a MANET in an undirected graph G ¼ ðIN; ILÞ that consists of a finite set of nodes, IN, and a finite set of communication links, IL, where each element is a tuple ðN$_j$; N$_k$Þ of nodes in the network. To focus on the selfish replica allocation problem, we do not consider selfishness in data forwarding throughout this paper. We generate the next hypothesis, related to those in [10] [12]. Each node in a MANET has a unique identifier. All nodes that are placed in a MANET [3] are indicated by N ¼ fN$_1$; N$_2$; . . .; Nmg, where m is the full amount number of nodes.

2. PROPOSED STRATEGY

Our approach consists of three parts:

- Identify Selfish Nodes
- Make the SCF-tree
- Distribute Replica

At a accurate period, or rearrangement period [6][13], each node executes the next actions: Each node notice the selfish nodes based on credit risk scores. Each node makes its own (partial) topology graph and builds its own SCF-tree by exclusive of selfish nodes. Based on SCF-tree, each node allocates replica in a fully dispersed manner. The CR [5] score is updated consequently during the query handing out phase [1]. We borrow the notion of credit risk from economics to effectively determine the "degree of selfishness". In economics, credit risk is the measured risk of loss due to a debtor's failure to pay of a loan. A bank examines the credit risk of an applicant prior to favorable the loan. The exact credit threat of the applicant indicates if he/she is creditworthy. We take a similar approach.

2.1 Detecting Selfish Node

$$Credit\ Risk = \frac{expected\ risk}{expected\ value}$$

In our strategy, each node estimate a CR score for each of the nodes to which it is connected. Each node shall approximate the "degree of selfishness" for all of its connected nodes based on the score. We first illustrate selfish features that may direct to the selfish replica allocation difficulty to make a decision both predictable value and predictable risk. Selfish features are separated into two group: node specific and query processing-specific. Node-specific features can be explained by allowing for the following case: A selfish node may split part of its individual memory space, or a little amount of data items, like the type-3 node. In this case, the amount of shared memory space and / or the number of public data items can be used to characterize the degree of selfishness. In our approach, the size of Nk's shared memory space, denoted as SSki , and the number of Nk's shared data items, denoted as NDki , observed by a node Ni, are used as node-specific features. Note that both SSki and NDki are Ni's predictable values, since Nk, which may be selfish or not, does not necessarily let Ni know the number of shared data items or size of the shared memory space.

The node-specific features can be used to represent the expected value of a node. For example, when node Ni observes that node Nk shares large SSki and NDki , node Nk may be treated as a valuable node by node Ni. As the query processing-specific feature, we employ the proportion of selfishness alarm of Nk on Ni, denoted as Pki , which is the ratio of Ni's data request being not served by the expected node Nk due to Nk's 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 gets larger, node Ni will treat Nk as a risky node because a large Pki means that Nk cannot serve Ni's requests due to selfishness in its memory procedure. To capably make out the predictable node (s), Ni be supposed to know the (expected) rank of other nodes' memory space. Our SCF-tree-based replica allocation techniques, fortunately, hold this statement.

2.2 Building SCF-Tree

The SCF-tree [4] based replica allocation techniques are encouraged by human friendship administration in the real world, where each person makes his / her own friends shaping a web and control friendship by himself / herself. He / she does not have to talk about these with others to carry on the friendship. The selection is solely at his / her caution. The main purpose of our novel replica allocation techniques is to diminish traffic operating cost, while achieving high data convenience. If the novel replica allocation techniques can allocate replica without discussion with other nodes, as in a human friendship management, traffic overhead will decrease.

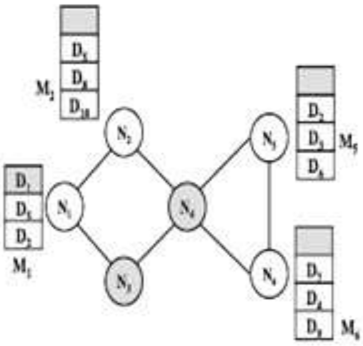

Fig. 2 : Building the SCF-tree

Prior to building the SCF-tree, each node makes its own partial topology graph G_i ¼ ðIN_i ; IL_iÞ, which is a component of the graph G. G_i consists of a finite set of the nodes connected to N_i and a finite set of the links,

Where

N_i 2 IN_i ; IN_i _ IN, and IL_i_ IL.

Since the SCF-tree consists of only non-selfish nodes, we need to measure the degree of selfishness to apply real-world friendship management to replica allocation in a MANET. We use the value of nCR^k_i for this purpose. Before constructing or updating the SCF-tree, node N_i eliminates selfish nodes from IN_i. Thus, N_i changes G_i into its own partial graph G^{ns}_i. More formally, we define G^{ns}_i as the undirected graph G^{ns}_i ¼ ðIN^{ns}_i ; IL^{ns}_iÞ, which consists of a finite set of non-selfish nodes detected by N_i, IN^{ns}_i, and a finite set of communication links among nodes N 2 IN^{ns}_i, IL^{ns}_i. IL^{ns}_i is derived by a smoothing out operation in graph theory.

For instance, if there exists a path where Nj; N_k

$h^{N_j\,;\,N_a;\,N_b;\,...\,;\,N_l;\,N_k}j\,a\,b\,l_s\,ki$ in $G_{i_{ns}'}$ 2 IN_i^{ns} and link removes every

N_a; N_b; . . . ; N_l 2 IN_i ¼ IN_i N_i_ IN_i ,

Including the selfish nodes and then replaces ðN_j; N_kÞ with a new edge (the new edge is added since we do not consider selfishness in data forwarding).

Based on G^{ns}_i, N_i builds its own SCF-tree, denoted as T_i^{SCF} . Each node has a parameter d, the depth of SCF-tree. When N_i builds its own SCF-tree, N_i first appends the nodes that are connected to N_i by one hop to N_i's child nodes. Then, N_i checks recursively the child nodes of the appended nodes, until the depth of the SCF-tree is equal to d. Figure 2 illustrates the network topology and some SCF-trees of N_1 and N_2 in Figure 1. In this example, we assume that all nodes are non-selfish nodes for simplicity.

2.3 Distribute 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.

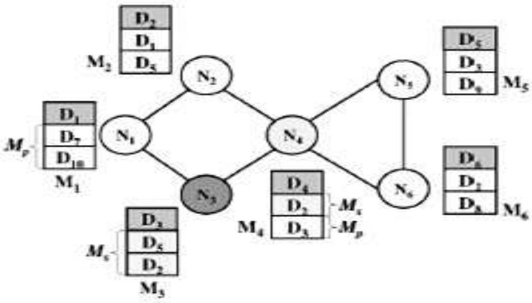

Fig. 3 : Distributing Replica

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.

3. EXPERIMENTAL RESULTS

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	0.45	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

From the above table, for each data D, different levels of access frequency is tabulated for each node N.

4. SIMULATION RESULTS

The simulateion result of data in each node is pointed below. Every node in a MANET calculates credit risk information on other connected nodes autonomously to compute the amount of selfishness. Since traditional replica allocation techniques failed to suppose selfish nodes, we also projected novel replica allocation techniques

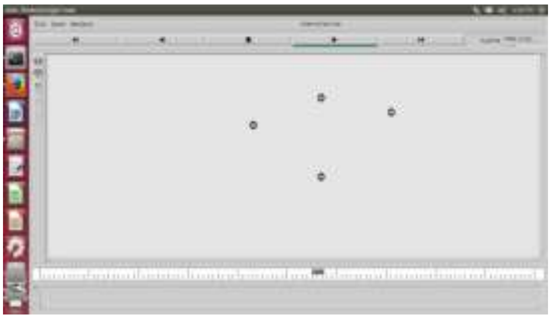

Fig. 4 : Value without Data Range

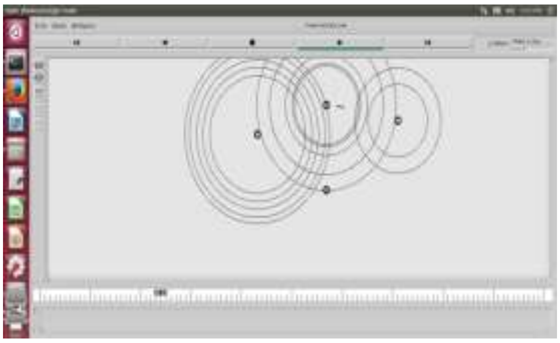

Fig. 5 : Different Nodes with Data Range

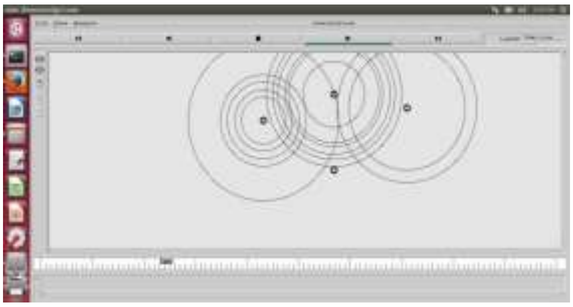

Fig. 6 : Different Nodes Grouping Data Range

5. CONCLUSION

This work was required by the reality that a selfish replica allocation strength through to on the whole poor data convenience in a MANET. We have designed a selfish node detection method and novel replica allocation techniques to hold the selfish replica allocation appropriately. The future strategies are inspired by the real-world clarification in economics in conditions of acknowledgment threat and in human friendship association in terms of choosing one's friends entirely at one's own discretion. We applied the view of credit risk from economics to perceive selfish nodes. Extensive simulation demonstrates that the proposed strategies outperform obtainable representative supportive replica allocation techniques in terms of data ease of access, message cost, and query delay. We are currently working on the impact of diverse mobility patterns. We plan to identify and hold false alarms in selfish replica allocation.

REFERENCES

1. Laura Giarré, Giovanni Neglia, and Ilenia Tinnirello, 2009. Medium Access in WiFi Networks: Strategies of Selfish Nodes, IEEE Signal Processing Magazine, 124-127.
2. Murthy, S. and J.J. Garcia-Luna-Aceves, 1996. An Efficient Routing Protocol for Wireless Networks, ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, 183-97.
3. Charles E. Perkins, editor.Ad Hoc Networking, Addison-Wesley, 2001.
4. Baumeister, R F and Leary,M R, 1995.The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motivation, Psychological Bull., Vol.117, No.3, 497-529.
5. Broch J, Maltz D A, Johnson D B, Y and Jetcheva J,1998. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Proc. ACM MobiCom, 85-97.
6. Cao G, Yin L, and Das C R, 2004. Cooperative Cache-Based Data Access in Ad Hoc Networks, Computer, Vol.37, No.2, 32-39.
7. Chun B G, Chaudhuri K, Wee H, Barreno M, Papadimitriou C H, and Kubiatowicz J, 2004. Selfish Caching in Distributed Systems: A Game-Theoretic Analysis, Proc. ACM Symp. Principles of Distributed Computing, 21-30.
8. Damiani E, di Vimercati S D C, Paraboschi S, and Samarati P, 2003. Managing and Sharing Servents Reputations in P2P Systems, IEEE Trans. Knowledge and Data Eng., Vol.15, No.4, 840-854.
9. Hara T and Madria S K, 2009. Consistency Management Strategies for Data Replication in Mobile Ad Hoc Networks, IEEE Trans. Mobile Computing, Vol.8, No.7, 950-967.
10. Khan S U and Ahmad I, 2009. A Pure Nash Equilibrium-Based Game Theoretical Method for Data Replication across Multiple Servers, IEEE Trans. Knowledge and Data Eng., Vol.21, No.4, 537-553.
11. Shanmuga Priya K P and Seethalakshmi V, 2013. Replica Allocation In Mobile Ad Hoc Network For Improving Data Accessibility Using SCF-Tree, International Journal of Modern Engineering Research, Vol.3, No.2, 915-919.
12. Lilu Odedra, Ashish Revar and Munindra H. Lunagaria, 2016. Comparative Analysis of Prevention and Detection Policies for Selfish Behaviour in MANET, International Journal of Advanced Research in Computer and Communication Engineering, Vol.5, No.2, 59-67.
13. Jananandhini E, 2014. Monitoring Selfish Nodes in MANET during Replica Allocation, IOSR Journal of Computer Engineering, Vol.16, No.5, 71-77.