RESEARCH ARTICLE OPEN ACCESS

Selfish Node Detection in Mobile Ad-Hoc Networks through Audit Approach with Correlation Function

M.Madhumathi [1], R.Ashok Kumar [2]
Pg Scholar [1], Assistant Professor [2]
Department of Computer science & Engineering,
SNS College of Engineering, Coimbatore
Tamil Nadu - India

ABSTRACT

Mobile ad-hoc networks (MANETs) depends on network cooperation schemes to work properly. It assumes that mobile nodes voluntary cooperate in order to work properly. Nevertheless, if nodes have a selfish behaviour and are unwilling to cooperate, the overall network performance could be seriously degraded. The proposed system develops a homomorphic linear Authenticator based public auditing architecture that allows the detector to verify the truthfulness of the packet loss information reported by nodes. The high detection accuracy is achieved by exploiting the correlations between the positions of lost packets, as calculated from the auto-correlation function (ACF) of the packet-loss bitmap a bitmap describing the lost/received status of each packet in a sequence of consecutive packet transmissions. Therefore, by detecting the correlations between lost packets, one can decide whether the packet loss is purely due to regular link errors, or is a combined effect of link error and selfish node. As shown in the paper, this audit based approach reduces the time and better detection accuracy.

Keywords:- MANETs, Audit Approach, Homomorphic Linear Authenticator, Auto-Correlation Function

I. INTRODUCTION

MANETS are used in many contexts such as in mobile social networks, emergency deployment, intelligent transportation systems etc. Nodes in a MANET [1] freely move around while communicating with each other. These networks may perform in the presence of nodes with a selfish behaviour particularly when operating under energy constraints. In the transmission of the packets these selfish nodes will typically not cooperate, seriously affect the network performance. Selfish nodes are the nodes participating in the network which are hesitated to forward the packets in order to save the resources under the energy constraints. Nodes which all are less frequent may also fail to cooperate either intentionally (a malicious behaviour) or due to faulty software or hardware.

Node misbehaviour means deviation from the original routing and forwarding. The source node can relay packets to the destination node through other nodes in MANET. The selfish nodes [2] do not participate in the routing process, which intentionally delay and drop the packet. These misbehaviours of the selfish nodes will impact the efficiency, reliability, and the fairness. A selfish node does not perform the process related to packet forwarding function for data packets unrelated to itself. The selfish node utilizes its limited resources only for its own purpose because of the energy and storage constraints for each node in the MANET. It aims to save its resources to the maximum, so this type of misbehaving node discards all incoming packets except those which are destined to it. The selfish nodes neglect to share their resources, such as battery power, CPU time, and memory space to other nodes in MANET. This behaviour is observed in the data link/MAC layer, which is decisive, specifically when the mobile nodes possess small residual power.

The main objective of the proposed work is to detect the selfish node in MANET using the audit based detection technique. The proposed method consists of a packet dropping detection scheme and a selfish node mitigation scheme. The selfish node is required to generate a trust report during each neighbor, which reports its previous communication reports to the neighboring node. Based on that report, the neighboring node detects if the selfish node has dropped packets. The neighboring node gathers the trust report to detect misreporting and then it finds out which node has dropped packets. A selfish node may report a false record to hide the dropping from being detected.

II. RELATED WORKS

There are two main strategies to deal with selfish behaviour in cooperative networks. The first approach tries to motivate the nodes to actively participate in the forwarding activities. For example, in [3], [4] the authors presented a method using a virtual currency called Bnuglet. Zhong et al. [5] proposed SPRITE, a credit-based system to incentivate participation of selfish nodes in MANET communication. These incentivation methods present several problems, such as the need for some kind of implementation infrastructure to maintain the accounting and they usually rely on the use of some kind of tamper-proof hardware. The COMMIT Protocol [6] combines game-theoretic techniques to achieve truthfulness and an incentivation payment scheme to reduce the impact of selfish nodes on routing protocols. Regarding the detection and exclusion approach, there are several solutions for MANETs and DTNs. A first study about misbehaving nodes and how watchdogs can be used to detect them was introduced in [1]. The authors proposed a Watchdog and Pathrater over the DSR protocol to detect non-forwarding

nodes, maintaining a rating for every node. In [7] another scheme for detecting selfish nodes based on context aware information was proposed. In previous works it has been shown how some degree of cooperation can improve the detection of selfish or misbehaving nodes. The CONFIDENT protocol was proposed in [8], which combines a watchdog, reputation systems, Bayesian filters and information obtained from a node and its neighbours to securely detect misbehaving nodes. The system's response is to isolate those nodes from the network, punishing then indefinitely.

More recently, papers have focused on DTNs. In [9], the author introduces a model for DTN data relaying schemes under the impact of node selfishness. A similar approach is presented that shows the effect of socially selfish behaviour. Social selfishness is an extension of classical selfishness (also called individual selfishness). A social selfish node can cooperate with other nodes of the same group, and it does not cooperate with other nodes outside the group. Nevertheless, these approaches do not evaluate the effect of false positives, false negatives and malicious nodes. For example, the approach in [10] only transmits positive detections. The problem, as shown in the evaluation sections, is that if a false positive is generated it can spread this wrong information very quickly on the network, isolating nodes that are not selfish. Therefore, an approach that includes the diffusion of negative detections as well becomes necessary. Another problem is the impact of colluding or malicious nodes. Although a reputation system, as the one presented in [10], can be useful to mitigate the effect of malicious nodes, it clearly depends on how are combined local and global ratings, as shown in this paper. Another implementation issue is the high imposed overhead due to the flooding process in order to achieve a fast diffusion of the information. Since our approach is based on contacts, it has been proven that the overhead is greatly reduced.

III. PROBLEM DEFINITION

The main problem in replica allocation is selfishness of nodes. That selfish node did not share its own memory to help other nodes. But it enjoys all resources of other nodes, and limitedly share its own resources to others. Such selfish behaviour of node causes a serious problem in network in transmission of packets. Those selfish nodes did not consume their own services like battery and memory storage to transmit the data to others. Then entire network goes to retransmission stage, any how this is difficult. So that detecting particular selfish node is easy and allocated replica to those nodes.

IV. PROPOSED APPROACH

The proposed mechanism is based on detecting the correlations between the lost packets over each hop of the path. The basic idea is to model the packet loss process of a hop as a random process alternating between 0 (loss) and 1 (no loss). Specifically, consider that a sequence of M packets that are transmitted consecutively over a wireless channel. By observing whether the transmissions are successful or not, the receiver of the hop obtains a bitmap $(a_1, \ldots . a_M)$, where $a_j \in \{0,1\}$ for packets $j = 1, \ldots . , M)$. The correlation of the lost packet is calculated as the auto-correlation function of this bitmap. Under different selfish node conditions, i.e., link-error versus malicious dropping, the instantiations of the packet-loss random process should present distinct dropping patterns (represented by the correlation of the instance). This is true even when the packet loss rate is similar in each instantiation. To verify this property have simulated the auto-correlation functions of two packet loss processes, one caused by 10 percent link errors, and the other by 10 percent link errors plus 10 percent malicious uniformly-random packet dropping. It can be observed that significant gap exists between these two auto-correlation functions. Therefore, by comparing the auto-correlation function of the observed packet loss process with that of a normal wireless channel, one can accurately identify the cause of the packet drops.

A. Network Model

The number the nodes in *PSD* = {*n*1…… *nk*} in ascending order with *k* = {*PSDj*}. Node *ni* is *upstream* of *nj* if *i* < *j* and is *downstream* of *nj* if *i* > *j*. Also assume the source receives feedback from the destination when a significant performance drops in metrics of interest, such as throughput or delay occurs. Here assume that message integrity and authenticity can be verified using resource efficient cryptographic methods, i.e., nodes may use the Elliptic Curve Digital Signature Algorithm (ECDSA) that has been shown feasible for resource limited devices such as sensors. Finally, there are at least two independent paths to any destination, i.e., the network is two-connected. This assumption is essential for reaching every node in *PSD* through a disjoint path.

B. Adversarial Model

The adversarial assume the existence of multiple independently misbehaving nodes in PSD. Source or destination node in may be misbehaving, except the source and the destination which are assumed to be trusted. The goal of misbehaving nodes is to degrade throughput while remaining undetected. Misbehaving nodes are assumed to be aware of the mechanisms used for misbehavior detection.

C. Public Audit Request

The goal of the audit phase is to verify that the audited node ni forwards packets to the destination. When a node is audited, it has to provide proof of the packets it forwards. The proof is used by the source S to perform a simple membership test: Did node ni forward packets in set X to the next hop. The audit phase occurs in three steps: (a) sending an audit request, (b) constructing a behavioral proof, and (c) processing the behavioral proof.

Once misbehavior has been detected in PSD, the source S selects a node ni to be audited based on the search phase. The source constructs a routing path PSni such that PSni and PSD are disjoint to avoid the audit request being dropped by the misbehaving node. The source also selects an audit packet count, acount, denoting the duration of the audit in terms of number of packets. The value of account is user-definable and must be sufficiently large to differentiate misbehavior from normal packet loss rate.

Lastly, S selects an initial packet sequence number start, indicating the sequence number of the packet where the audit begins. The source signs the audit request to enable the verification of its authenticity and integrity.

When a node is audited, it constructs a behavioral proof of the set of all packets it forwards, from astart to astart + acount, denoted by X = {x1; x2…… xN}. Buffering packets themselves would require large amount of storage and significant overhead for transmission back to the source. On the other hand, request algorithm provide a compact representation of membership for a set X = {x1; x2……..xN} in an m-bit vector v with m ¿ N. For an empty set X, all m bits of v are initialized to zero.

When S receives the behavioral proof from ni, it verifies its authenticity and discards vi if the signature checks fails. If ni fails to respond to the audit request, S may re-transmit the request using alternative paths. After a certain number of reply failures, S assumes that the node ni is suspicious of misbehaving and continues with the algorithm execution. So far we have illustrated how the source S evaluates the behavior of node ni via auditing. We now show how S selects nodes for audit in order to identify misbehaving ones. We define the notion of a suspicious set V as the set of nodes ni 2 PSD which have not been shown honest.

Once the search process has converged on the misbehaving link, the two suspicious nodes ni; ni+1 are excluded in turn from the routing path to the destination D. The node preceding the first suspicious node will split the traffic between ni; ni+1 in turn uses node n3 to exclude in turn suspicious nodes n4 and n5. The source alerts D that two suspicious nodes are monitored via path exclusion. The destination creates two request algorithm, vDi,vDi+1 corresponding to the packets routed through suspicious nodes ni; ni+1, and send them to S. The source compares vi; vi+1 with its own vSi ; vSi+1, and identifies the misbehaving node.

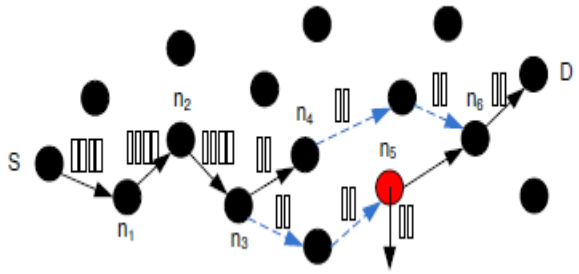

Fig. 1 Pubic audit request

Algorithm: public request audit Algorithm

1: Initialize: $V_l \leftarrow n_1, V_r \leftarrow n_{|p_{SD}|}, V_n = \{V_1 \dots \dots . V_r\}$
2: while $|V_n| > 2$ do
3: $audit(n_i) = V[rand]$
4: if $|X_i \cap X_s| \approx |X_s|$ then
5: $V_l \leftarrow n_i$
6: else
7: $V_r \leftarrow n_i$
8: end if
9: end while
10: return V_n

The proposed algorithm considers a sophisticated misbehaving node that changes its dropping pattern to avoid identification. Here describe this behavior by an example. misbe having node *n*1 drops packets. The source uses binary search to identify the misbehavior, choosing node *n*3 to audit. The audit reply of *n*3 fails the membership test, reducing the suspicious set to *V*1 = *fn*1*; : : : ; n*3*g*. The source then audits node *n*2,. Binary search is determine allowing *n*1 to predict the order that nodes are audited. Node *n*1 behaves honestly, thus *n*2's audit response passes the membership test. By changing its behavior, *n*1 removes himself from *V*.

D. Security Implementation

Prime fields are fields whose sets are prime. In other words, they have a prime number of members. Prime fields turn out to be of great use in asymmetric cryptography since exponentiation over a prime field is relatively easy, while its invserse, computing the logarithm, is difficult. The "Diffie-Hellman Method for Key Agreement" allow two hosts to create and share a secret key. Mathematically, a proof to this effect is neither known nor thought to be forthcoming. Before wide-scale implementation, it is thus of the utmost importance that an extensive investigation of the true complexity of the problem is done in order to obtain the highest degree of confidence in the security of discrete logarithm based cryptographic systems. Such an investigation is in progress by various researchers around the world.

KeyGen: Given the domain parameters (a,b,p,G,n,E) of an elliptic curve E over finite field Fp where p is a large prime that satisfy . Where G is the base point of order n, note that n*G = ∞, the private key x is randomly selected from [1, n-1], the public key is Y=xG, another point on the curve.

Encryption: Given the plaintext m and Y, output C
1. k ∈ [1, n – 1]
2. M = map (m)= mG
3. C= (R, S) = (kG, kY+mG)

Homomorphic operation: Given C1, C2... Cn, output C'
C'= (k1G, k1Y+m1G)+(k2G, k2Y+m2G)+…+(knG, knY+mnG)
C'= ((k1+k2+..kn)G, (m1+m2+mn)G+(k1+k2+..kn)Y)

Decryption: Given C' and the private key x, output m
1. M = S – xR
2. m =rmap(M)

The map function satisfies the desired additive homomorphic property. However, the reverse mapping function is the shortcoming of this scheme, the reverse function maps a given point M into a plaintext m, and thus, the ECDLP (defined above) on M must be resolved.

V. SIMULATION RESULTS AND ANALYSIS

During the simulation, each node starts its journey from a random spot to a random chosen destination. Once the destination is reached, the node takes a rest period of time in second and another random destination is chosen after that pause time. This process repeats throughout the simulation, causing continuous changes in the topology of the underlying network. **PDR** is the ratio of the number of data packets received by the destination node to the number of data packets sent by the source mobile node. It can be evaluated in terms of percentage (%). This parameter is also called "success rate of the protocols", and is described as follows:

$$\text{PDR} = \left(\frac{\text{Send Packet no}}{\text{Receive packet no}}\right) \times 100$$

Throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node.

$$\mathbf{X} = \frac{\mathbf{C}}{\mathbf{T}}$$

Where X is the throughput, C is the number of requests that are accomplished by the system, and T denotes the total time of systemobservation.

Average end-to-end delay Average end-to-end delay signifies how long it will take a packet to travel from source to destination node. It includes delays due to route discovery, queuing, propagation delay and transfer time.

$$D_{end-end} = N(d_{trans} + d_{prop} + d_{proc}$$

Where dend-end= end-to-end delay, dtrans= transmission delay,dprop= propagation delay,dproc= processing delay,dqueue= Queuing delay and N= number of links. This metric is useful in understanding the delay caused while discovering path from source to destination.

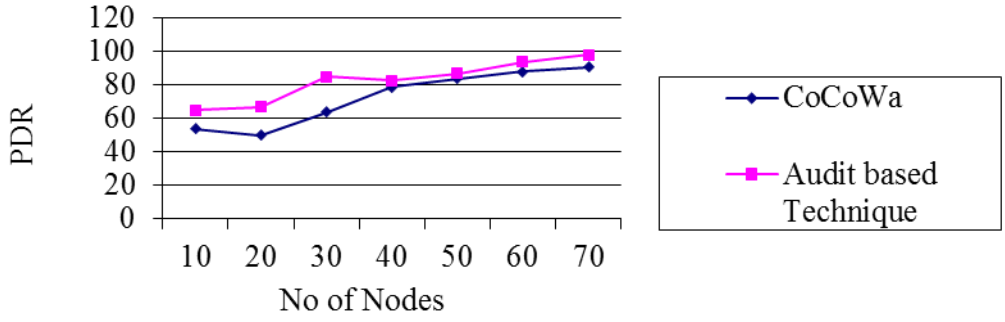

Fig. 2 Compare PDR existing with proposed

Shows packet delivery ratio against the number of nodes. It shows that the protocol has a better Audit method compare to cocowa.

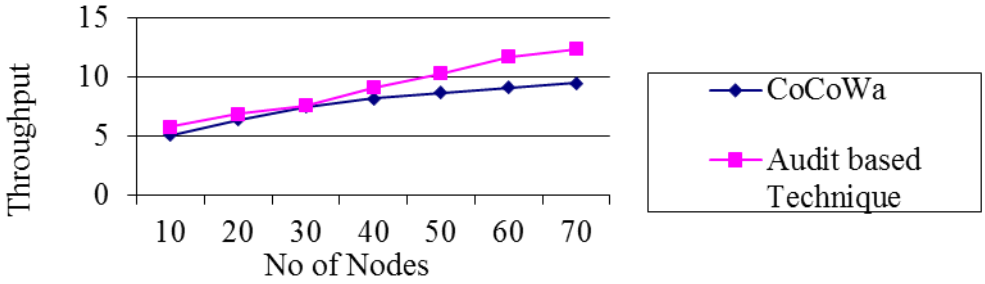

Fig. 3 Compare throughput existing with proposed

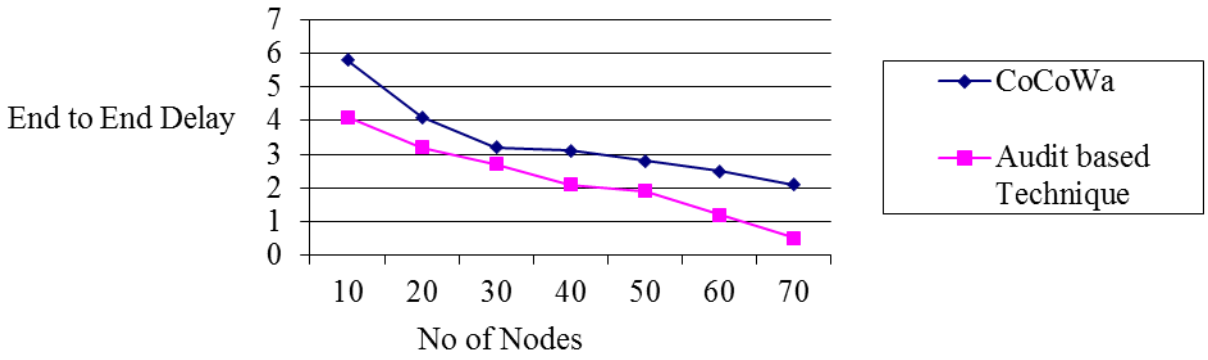

Fig. 4 Compare end to end delay existing with proposed

VI. CONCLUSIONS

The proposed a public audit request scheme algorithm which relies on the select audit of a subset of nodes along the source/destination path to identify selfish nodes. Each audited node uses search to construct a storage and communication efficient behavioral proof of the packets it forwarded. Here showed that proposed algorithm significantly reduces the communication over- head associated with the misbehavior identification process compared to Privacy-Preserving schemes. This reduction in resource expenditure comes at the expense of a logarithmic increase in the identification delay, due to the reactive nature of our scheme.

REFERENCES

[1] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "*Mitigating routing misbehavior in mobile ad hoc networks*," in Proc. ACM Mobicom Annu. Int. Conf. Mobile Comput. Netw., 2000, pp. 255–265.

[2] Ryu BG, Choi JH, Lee S*: Impact of node distance on selfish replica allocation in a mobile ad-hoc network*. Ad Hoc Netw. 2013, 11: 2187-2202. 10.1016/j.adhoc.2013.05.001.

[3] L. Buttyan and J.-P. Hubaux, "*Enforcing service availability in mobile ad-hoc WANs*," in Proc. 1st Annu. Workshop Mobile Ad Hoc Netw. Comput., 2000, pp. 87–96.

[4] L. Buttyan and J.-P. Hubaux, *"Stimulating cooperation in self organizing mobile ad hoc networks*," Mobile Netw. Appl., vol. 8, pp. 579–592, 2003.

[5] S. Zhong, J. Chen, and Y. Yang, "*Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks*," in Proc. IEEE Conf. Comput. Commun., Mar. 2003, vol. 3, pp. 1987–1997.

[6] S. Eidenbenz, G. Resta, and P. Santi, "*The COMMIT protocol for truthful and cost-efficient routing in ad hoc networks with selfish nodes*," IEEE Trans. Mobile Comput., vol. 7, no. 1, pp. 19–33, Jan. 2008.

[7] K. Paul and D. Westhoff, "*Context aware detection of selfish nodes in DSR based ad-hoc networks*," in Proc. IEEE Global Telecommun. Conf., 2002, pp. 178–182.

[8] S. Buchegger and J.-Y. Le Boudee, "*Self-policing mobile ad hoc networks by reputation systems*," IEEE Commun. Mag., vol. 43, no. 7, pp. 101–107, Jul. 2005.

[9] M. Karaliopoulos, "*Assessing the vulnerability of DTN data relaying schemes to node selfishness*," IEEE Commun. Lett., vol. 13, no. 12, pp. 923–925, Dec. 2009.

[10] P. Michiardi and R. Molva, "*CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks*," in Proc. 6th Joint Working Conf. Commun. Multimedia Secur., 2002, pp. 107–121.