Random Feedbacks for Selfish Nodes Detection in Mobile Ad Hoc Networks

Djamel Djenouri[1], Nabil Ouali[2], Ahmed Mahmoudi[2], and Nadjib Badache[2]

[1] Basic Software Laboratory, CERIST Center of research, Algiers, Algeria
ddjenouri@mail.cerist.dz
[2] LSI, USTHB University, Algiers, Algeria
nabilouali@hotmail.com, mahmoudi_pfe@yahoo.fr, badache@lsi-usthb.dz

Abstract. A mobile ad hoc network (MANET) is a temporary infrastructureless network, formed by a set of mobile hosts that dynamically establish their own network *on the fly* without relying on any central administration. Mobile hosts used in MANET have to ensure the services ensured by the powerful fixed infrastructure in traditional networks, the packet forwarding is one of these services.
Resource limitation of MANET's nodes, particulary in energy supply, along with the multi-hop nature of these networks may cause a new problem that does not exist in traditional networks. To save its energy a node may behave ***selfishly (no-cooperatively)***, thus it misbehaves by not forwarding packets originated from other nodes, while using their resources to forward its own packets to remote recipients. Such a behavior hugely threatens the QoS (Quality of Service), and particulary the packet forwarding service availability. Some solutions for selfish nodes detection have been recently proposed, but almost all these solutions rely on the monitoring in the promiscuous mode technique of the watchdog [1], which suffers from many problems especially when using the power control technique. In this paper we propose a new approach to detect selfish nodes unwilling to participate in packet forwarding, that mitigates some watchdog's problems. We also assess the performance of our solution by simulation.

Keywords: mobile ad-hoc networks, security, selfishness, packet forwarding, GloMoSim.

1 Introduction

In some MANETs applications, such as in battlefield or rescue operations, all the nodes have a common goal and their applications belong to a single authority, thus they are *cooperative by nature*. However, in many civilian applications, such as networks of cars and provision of communication facilities in remote areas, nodes typically do not belong to a single authority and they do not pursue a common goal. In such self-organized networks forwarding packets for other nodes is not in the direct interest of any node, so there is no good reason to trust nodes and assume that they always cooperate. Indeed, each node tries to

T. Magedanz, E.R.M. Madeira, and P. Dini (Eds.): IPOM 2005, LNCS 3751, pp. 68–75, 2005.

save its resources, particularly its battery power which is a precious resource. Recent studies show that most of the nodes energy in MANET is likely to be devoted to forward packets in behalf of other nodes. For instance, Buttyan and Hubaux simulation studies [2] show that; when the average number of hops from a source to a destination is around 5 then almost 80% of the transmission energy will be devoted to packet forwarding.

Therefore, to save energy nodes may misbehave and tend to be *selfish*. A selfish node regarding the packet forwarding process is a node which takes advantage of the forwarding service and asks others to forward its own packets, but does not actually participate in this service. Some solutions have been Recently proposed. Almost all these solutions, however, rely on the watchdog [1] technique which suffers from many problems. It might cause false accusation, especially when using the power control technique employed by some new power-aware routing protocols following the watchdog's proposal.

Our purpose in this work is to propose a novel solution to mitigate some of these problems.

The remainder of this paper is organized as follows: In the next section we present related work, and then we present and discuss our solution in section 3. Section 4 is devoted to the simulation-based performance evaluation. Finally, section 5 concludes the paper and summarizes our perspectives.

2 Related Work

To the best of our knowledge, Marti et al. [1] are the first who dealt with the problem of nodes misbehavior on packet forwarding, they proposed the *watchdog* which they implemented with the dynamic source routing protocol (DSR) [3]. The watchdog lies on monitoring neighbors in the promiscuous mode, i.e each node in the source route monitors its successor after it sends it a packet to forward by overhearing the channel and checking whether the monitored relays the packet. The monitor accuses a monitored node as misbehaving when it detects that this latter drops more than a given number (threshold) of packets. This basic technique have been used by almost all the subsequent solutions. Nevertheless, it suffers from some problems, especially when using the power control technique, employed by some new power-aware routing protocols following the watchdog's proposal [4] [5] [6].

Assume three aligned nodes: A, B and C, such that A sends B a packet and monitors its forwarding to C, and lets assume that B uses the power control technique. When A is closer to B than C, B could circumvent the watchdog by using a transmission power strong enough to reach A, but less than the one required to reach C, which is power efficient for B. On the other hand, when C is closer to B than A, and B behaves correctly but uses the power control technique, A could not overhear B's forwarding to C, which might results in false detections when the number of packets falsely detected exceeds the configured threshold. Further, packet collisions either at C or A, during the monitoring, could cause problems. When B's forwarding causes a collision at C, the former could circumvent to A by not retransmitting the packet. On the other side, if B's forwarding results in a collision at A, A could falsely note a B's packet dropping.

In [7], Yang et al. describe a unified network layer solution to protect both routing and data forwarding in the context of AODV. Michiardi and Molva [8] suggest a generic reputation-based mechanism, namely CORE (COllaborative REputation Mechanism to enforce node cooperation in MANETs), that can be easily integrated with any network function. CONFIDANT (Cooperation Of Nodes Fairness in Dynamic Ad hoc Networks) is another interesting reputation-based solution, proposed by Buchegger and Le Boudec [9] [10]. It relies on DSR [3] used as benchmark in the GloMosim-based simulation study performed by the authors to evaluate the new DSR fortified by CONFIDANT.

All these solutions, however, rely on the watchdog technique in their monitor component.

Buttyan and Hubaux [11] propose an efficient preventive economic-based approach stimulating nodes to cooperate, which is modeled and analyzed in [2]. The authors introduce what they called *virtual currency* or *nuglets*, along with mechanisms for charging/rewarding service usage/provision. The main idea of this technique is that nodes which use a service must pay for it (in nuglets) to nodes that provide the service. Other stimulating preventive approaches are based on game theory, such as [12].

These preventive solutions motivate nodes to cooperate, but do not aim at detecting the misbehaving nodes contrary to the previous solutions.

In [13], Papadimitratos and Haas present the SMTP protocol. It is a hybrid solution that mitigates the selfishness effects (packets lost) by dispersing packets, and detects the selfish misbehavior by employing end-to-end feedbacks. This kind of feedbacks allows the detection of the routes containing selfish nodes, but fails to detect these nodes. To overcome this problem, Kargl et al. [14] propose *iterative probing*, that detects links containing selfish nodes, but fails to detect appropriate nodes. To find the appropriate node on a link after an iterative probing, authors propose what they called *unambiguous probing*, which is based on the watchdog, thus suffers from its problems.

3 Novel Solution

3.1 Solution Overview

We define a new kind of feedbacks we call *two-hop ACK*, an ACK that travels two hops. Node C acknowledges packets sent from A by sending this latter via B a two-hop ACK.

Node B could, however, escape from the monitoring without being detected by sending A a *falsified* two-hop ACK. Note that performing in this way is power economic for B, since sending a short packet like an ACK consumes too less energy than sending a data packet. To avoid this vulnerability we use an asymmetric cryptography based strategy as follows:

Node A generates a random number and encrypts it with C's public key (PK) then appends it in the packet's header as well as A's address. When C receives the packet it gets the number back, decrypts it using its secret key (SK), encrypts it using A's PK, and puts it in a two-hop ACK which is sent

back to A via B. When A receives the ACK it decrypts the random number and checks if the number within the packet matches with the one it has generated, to validate B's forwarding regarding the packet in question. However, if B does not forward the packet A will not receive the two-hop ACK, and it will be able to detect this dropping after a time out. A will then accuse B as selfish when the number of packets dropped it detects exceeds a given threshold.

This encryption strategy needs a security association between each pair of nodes to ensure that nodes share their PK with each other. This requires a key distribution mechanisms which is out of the scope of this paper, but a mechanism like [15] or [16] can be used.

The watchdog's problems are mitigated with this approach, since B's forwarding validation at A is not only related to B's transmission, but to C's reception. Still, the problem with this first solution is that it requires a two-hop ACK for each packet, which might result in important overhead. To decrease this cost, we propose to *randomize* the ACK request. viz. A does not ask C an ACK for each packet, but when sending a packet to forward, it *randomly* decides whether it asks an ACK or not, with a probability p (probability of requesting an ACK). This *random* selection strategy prevents the monitored node from deducting which packets contain ACK requests. Note that getting such information allows a selfish to drop packets with no requests without being detected.

The probability p is either continuously decreased (resp increased) with α after each ACK request during a series of ACK requests[1] (resp sending a packet without requesting an ACK during a series of no-requests) till reaching 0 (resp 1), or switched after a series of requesting (res no-requests) to a non-request (res request). In these two latter cases of switching, p takes the value θ, the initial probability, which is continually updated as follows:

It is set to 1 upon a lack of a requested ACK (after the timeout), and decreased each time the requested ACK is received, till reaching the minimum value θ_0. This way, more trust is given to well-behaving nodes, and by setting θ to 1 the ACK request is enforced after a lack of ACK.

3.2 Discussions

Unlike the current detective solutions that are based on the promiscuous mode monitoring (the watchdog), ours relays on a new technique, namely the two-hop ACK. The monitoring (node A) validates the monitored (node B) forwarding when it receives an ACK from the successor of this latter (node C). This process can be generalized along the path for each consequent two hops until the destination, and efficient encryption/decryption operations have been added in order to authenticate the two-hop ACKs and secure the solution against spoofing attacks.

Getting rid of the promiscuous mode makes our solution independent of transmission powers, and resolves the watchdog problems related to the employment of the power-control technique. Further, it resolves the receiver collision

[1] A series of ACK requests is a series of packets, for which A asks C ACKs

problem presented previously. When a collision appears at C, B should retransmit the packet, otherwise A will not validate its forwarding. This because B's forwarding will not be validated at A until C really receives the packet and sends back the two-hop ACK, unlike the watchdog where the validation is only related to B's first transmission.

Moreover, The two-hop ACK technique allows to detect the appropriate selfish node, unlike the end-to-end ACKs [13] and the iterative probing [14].

As illustrated, authentication of the two-hop ACK packet is ensured by employing encreption/decreption operations on a random number, generated by the monitor and piggybacked to the monitored packet. These operations have minor impact on computation overhead, since they are applied merely on the random number and not on the whole packet holding it. Note that we avoided the use of digital signatures in order to avoid useless packet's hash computation.

Instead of requiring a two-hop ACK for each packet, we proposed to *randomize* the ACK request, where the monitor *randomly* decides whether it asks an ACK or not, with a certain probability which is continuously updated in such a way to give more trust (less ACK requirements) to well-behaving nodes, and enforce requirements after an ACK reception failure. This *random* selection strategy prevents the monitored node from deducting which packets contain ACK requests, thus from dropping two-hop ACK free packets[2].

Still, like the watchdog and all the currently proposed solutions, the challenging problem of cooperative misbehavior remains untreated with our proposal. That is, if B and C collude, such as B does not correctly monitor C (i.e it does not report back when C drops packets), C will not be detected. This problem represents an open research topic.

4 Simulation Study

To asses the proposed protocol performance we have driven a GloMoSim-based [17] simulation study, that we will present hereafter.

We have simulated a network of 50 nodes, located in an area of $1500 \times 1000 m^2$, where they move following the random way-point [18] model with an average speed of 1m/s, for 900 seconds (the simulation time). To generate traffic, we used three CBR sessions between three pairs of remote nodes, each consists of continually sending a 512 bytes data packet each second. On each hop, each data packet is transmitted using a controlled power, according to the distance between the transmitter and the receiver.

We compare two versions of our protocol, 2HopACK and Random 2HopACK, as well as the watchdog (WD), with regard to the selfish detection rate, the false detection rate (rate of false accusations as selfish) and the number of two-hops ACKs (which represents the overhead). We measured these metrics vs the selfish nodes rate, which represents the rate of nodes that behave selfishly and drop packets they are asked to relay. Each point of the plots presented hereafter has been obtained by averaging five measurements with different seeds. Note that

[2] Packets that do not require a two-hop ACK

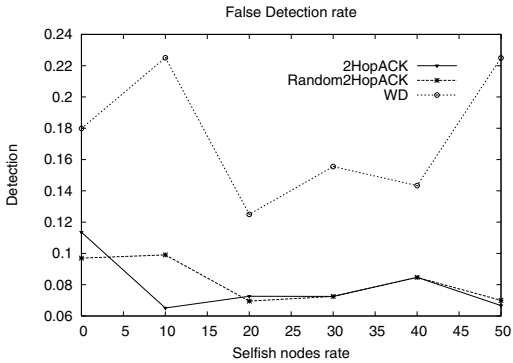

Fig. 1. False detection vs. selfish rate

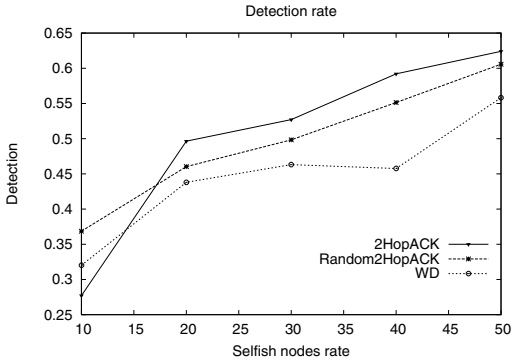

Fig. 2. True detection vs. selfish rate

we implemented our protocol with DSR for this simulation, like the watchdog. However, it can be implemented with any source routing protocol. Also note that WD requires no kind of ACK, so the last metric (number of two-hop ACK) concerns merely our protocol's versions.

The first version of our protocol requires an ACK for each packet, while the second one uses the efficient technique of randomizing the ACK request, which reduces the overhead especially when the selfish nodes rate is low, as shown in figure 1. However, the cost of this overhead decreasing is a minor loss in detection efficiency, as shown in figure 2. But we can clearly see in the same figure that both versions have better detection than WD. Figure 3 illustrates how our protocol (the two versions) decreases hugely the false detection rate compared with WD.

5 Conclusion and Future Work

In this work we have focused on the selfish nodes detection problem, and we proposed an approach that mitigates some watchdog's drawbacks. Our solution is based on the two-hop ACK, it allows to detect the selfish node, unlike the

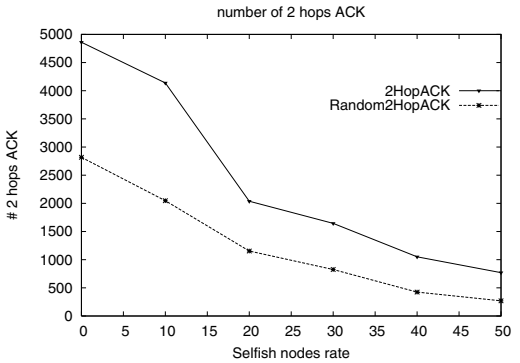

Fig. 3. Number of two-hop ACK vs. selfish rate

end to end ACK that just detects the route containing such a node, or iterative probing which just detects the appropriate link. To reduce the overhead while keeping efficiency, we have suggested to ask two-hop ACKs at random points, instead of of asking them for each data packets.

We have assessed the performance of our solution by a simulation study, whose results show how our approach outperforms the watchdog, especially regarding false detections. The simulation results also show how the random requesting strategy reduces the overhead.

As perspective, we plan to complete the proposal by defining actions that have to be taken when a node is accused as a selfish, and particulary by proposing a mechanism allowing nodes to exchange their knowledge regarding nodes that behave selfishly.

References

1. Marti, S., Giuli, T., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: ACM Mobile Computing and Networking, MOBICOM 2000. (2000) 255–65
2. Buttyan, L., Hubaux, J.P.: Stimulating cooperation in self-organizing mobile ad hoc networks. ACM/Kluwer Mobile Networks and Applications, Vol 8, N 5 (2003)
3. David, B., David, A.: Dynamic source routing in ad hoc wireless networks. Mobile Computing, Chapter 5 (1996) 153–181
4. Djenouri, D., Badache, N.: New power-aware routing for mobile ad hoc networks. Accepted in the International Journal of Ad Hoc and Ubiquitous Computing (Inderscience) (2005 (to appear))
5. Doshi, S., Brown, T.: Minimum energy routing schemes for a wireless ad hoc network. In: IEEE INFOCOM 2002. (2002)
6. Djenouri, D., Badache, N.: Simulation performance evaluation of an energy efficient routing protocol for mobile ad hoc networks. In: IEEE International Conference on Pervasive Services (ICPS'04), American University of Beirut (AUB), Lebanon (2004)

7. Yang, H., Meng, X., Lu, S.: Self-organized network layer security in mobile ad hoc networks. In: ACM MOBICOM Wireless Security Workshop (WiSe'02), Georgia, Atlanta, USA. (2002)
8. Michiardi, P., Molva, R.: Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Communication and Multimedia Security 2002 Conference, Portoroz, Slovenia. (2002)
9. Buchegger, S., Boudec, J.Y.L.: Performance analysis of the confidant, protocol cooperation of nodes fairness in dynamic ad hoc networks. In: Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02), Lausanne, Switzerland. (2002) 80–91
10. Buchegger, S., Le-Boudec, J.Y.: A robust reputation system for p2p and mobile ad-hoc networks. In: Second Workshop on the Economics of Peer-to-Peer Systems. (2004)
11. Buttyan, L., Hubaux, J.P.: Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. Technical report No. DSC/2001/001, Swiss Federal Institution of Technology, Lausanne, Switzerland (2001)
12. Srinivasan, V., Nuggehalli, P., F.Chiasserini, C., R.Rao, R.: Cooperation in wireless ad hoc networks. In: IEEE INFOCOM'03, San Francisco, California, USA. (2003)
13. Papadimitratos, P., Haas, Z.J.: Secure data transmission in mobile ad hoc networks. In: ACM MOBICOM Wireless Security Workshop (WiSe'03), San Diego, California, USA. (2003)
14. Kargl, F., Klenk, A., Weber, M., Schlott, S.: Advanced detection of selfish or malicious nodes in ad hoc networks. In: 1st European Workshop on Security in Ad-Hoc and Sensor Networks, ESAS 2004. (2004)
15. Capkun, S., Buttyan, L., Hubaux, J.P.: Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing, Vol.2, No.1 (2003) 52–64
16. Weimerskirch, A., Westhoff, D.: Zero common-knowledge authentication for pervasive networks. In: Selected Areas in Cryptography. (2003) 73–87
17. Zeng, X., Bagrodia, R., Gerla, M.: Glomosim: A library for the parallel simulation of large-scale wireless networks. In: proceeding of the 12th Workshop on Parallel and distributed Simulation. PADS'98. (1998)
18. Djenouri, D., Derhab, A., Badache, N.: Ad hoc networks routing protocols and mobility. International Arab jornal of Information Technology (2005 (to appear))