Random Two-hop ACK to Detect Uncooperative Nodes in MANETs

Djamel Djenouri§, Nabil Ouali♯, Ahmed Mahmoudi♯
§: Basic Software Laboratory, CERIST Center of Research, Algiers, Algeria.
E-mail: ddjenouri@mail.cerist.dz
♯: Computer Science Department, USTHB University, Algiers, Algeria.

ABSTRACT

The resource limitation of nodes used in self-organized mobile ad hoc networks (MANETs), particulary in energy supply, and 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*** or ***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. To mitigate some of these problems we propose hereafter a novel approach, then we assess its performance by simulation.

1. RELATED WORK

To the best of our knowledge, Marti et al. [1] are the first who dealt with this problem, they proposed the *watchdog* which they implemented with the dynamic source routing protocol (DSR) [2]. A technique based 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. However, 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 [3].

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 overhears B's forwarding to C, which might result in false detections when the number of packet 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 [4] Yang et al. describe a unified network layer solution to protect both routing and data forwarding in the context of AODV. Michiardi and Molva [5] suggest a generic reputation-based mechanism (CORE), supposed to be easily integrated with any network function. Another reputation-based solution is CONFIDANT, proposed by Buchegger and Le-Boudec [6].

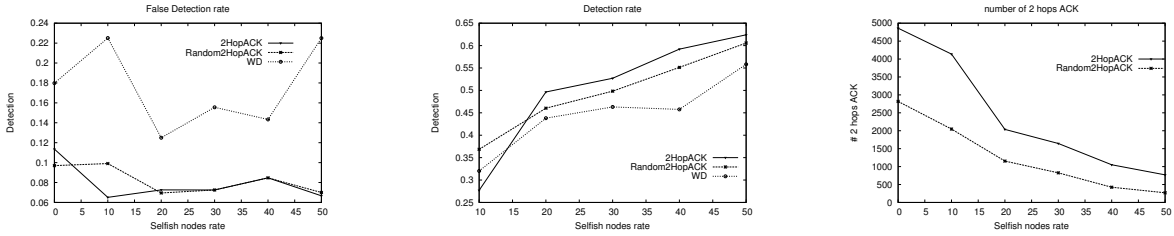
Still, all these solutions (and some others whose citations are omitted due to space limitation) rely on the watchdog technique for monitoring. They consequently inherit all the watchdog's problems cited above.

2. 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 appropriate packet. However, if B does not forward the packet A will not receive the two-hop ACK, and it will be able to detect this misbehavior after a time out. This 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 [7] 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. Nevertheless, the problem with this first



(a) (b) (c)

Figure 1: Simulation results

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 ask. viz. A does not asks 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 and ACK requests. Note that getting such information allows a selfish to drop packets with no request without being detected.
The probability p is either continuously decreased (resp increased) with α upon 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. SIMULATION RESULTS AND FUTURE WORK

To asses the proposed protocol performance we have driven a GloMoSim-based [8] simulation study. We have simulated a network of 50 nodes, located in an area of $1500 \times 1000m^2$, where they move following the random way-point model with an average speed of 1m/s, for 900 seconds (the simulation time). 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). In figure 1 these metrics are plotted vs the selfish node rate, the rate of nodes that behave selfishly. Note that WD is not involved in the last metric since it does not use two-hop ACKs, and requires no communication overhead for monitoring.
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(c). However, the cost of this overhead decreasing is a loss in detection efficiency, as shown in figure 1(b). But we can also see in the same figure that both versions have better detection than WD. Figure 1(c) illustrates how our protocol (the two versions) decreases the false detection rate compared with WD.
In this work we have focused on the selfish nodes detection problem, and we proposed an approach that mitigates some watchdog's drawbacks. 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.

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

4. REFERENCES

[1] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *ACM Mobile Computing and Networking, MOBICOM 2000*, 2000, pp. 255–65.

[2] B. David and A. David, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing, Chapter 5*, pp. 153–181, 1996.

[3] D. Djenouri and N. Badache, "New power-aware routing for mobile ad hoc networks," *Accepted in the International Journal of Ad Hoc and Ubiquitous Computing (Inderscience)*, 2005 (to appear).

[4] X. M. H. Yang and S. Lu, "Self-organized network layer security in mobile ad hoc networks," in *ACM MOBICOM Wireless Security Workshop (WiSe'02), Georgia, Atlanta, USA*, September 2002.

[5] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Communication and Multimedia Security 2002 Conference, Portoroz, Slovenia*, September 26-27 2002.

[6] S. Buchegger and J.-Y. Le-Boudec, "A robust reputation system for p2p and mobile ad-hoc networks," in *Second Workshop on the Economics of Peer-to-Peer Systems*, June 2004.

[7] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Transactions on Mobile Computing, Vol.2, No.1*, pp. 52–64, January 2003.

[8] X. Zeng, R. Bagrodia, and M. Gerla, "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*, May 1998.