Framework for monitoring and congestion control on Packet Dropping in Mobile Ad-hoc Network

Dr.Rohit Sharma¹, Himanshu Singh²

¹Assistant Professor, Computer Science Department, S.I.T.M., Barabanki, Uttar Pradesh, India
²Research Scholar, Computer Science Department, S.I.T.M., Barabanki, Uttar Pradesh, India

Abstract: Packet Dropper nodes is a severe problem in MANET and they not only are harmful for the sender and receiver but they also lead to congestion in mobile ad-hoc networks as the sender may get involved in sending packets again and again as there is no acknowledgement from the receiver. Mobile ad hoc network is wireless network of mobile nodes, with no centralized management and control. Network congestion leads to reduced throughput, routing and lifespan, etc. of network packets. Dropping control packets may be beneficial for selfish nodes and malicious ones as well. For example, simply by dropping RREQ (Route Request) packets a selfish node could exclude itself from routes and thereby avoid receiving data packets to forward. Similarly, a malicious could drop RERR (Route Error) packets to keep the use of failed routes, potentially resulting in a denial of service.

Keywords: — Manet, Packet dropper, RREQ, RERR.

I. INTRODUCTION

Secure routing in MANET is a topic that attracts more and more attention amongst researchers. When dealing with the selfish misbehaviour or the packet dropping attack (Byzantine black hole attack), most of the current sophisticated solutions focus on data packets. These solutions are not directly applicable to control packets. The number of control packets is too low compared with data packets, thus the solutions that rely on a long experience before making a judgment (such as the Bayesian-based methods[3]) are inappropriate, and a more realistic threshold is required. Further, some control packets are broadcast (RREQ), which might require a separate monitoring solution. In this paper we focus on these issues, and propose a comprehensive solution to monitor the forwarding of control packets, judge the monitored nodes, and isolate the detected misbehaving nodes. Regarding the monitoring, we propose separate solutions for directed and broadcast packets. For the first kind, we suggest the use of the two-hop ACK approach [8] and we propose a promiscuous mode based approach for the broadcast packets. As for the judgment, we propose a redemption method allowing nodes that are observed to forward packets to be redeemed. Finally, we propose to use a witness based isolation method to isolate the suspicious nodes.

II. RELATED WORK

Many secure routing protocols have been recently proposed for MANET. They aim at preventing the establishment of falsified routes. SAR [6] is a general proposal that can be implemented with a reactive routing protocol. It defines the trust degree that should be associated with each node, and ensures that a node is prevented from handling a RREQ (Route Request) unless it provides the required level. This way, data packets will be sent only through trusted nodes, with respect to the defined level. SAODV [6] is an implementation of SAR on AODV. One of the difficulties of this approach is the definition of the trust level. Further, assuming that nodes showing the required trust level are genuine is not always correct. SRP [7] is another secure routing protocol, based on DSR [4]. It prevents spoofing attacks, but it is vulnerable to the wormhole attack [2]. We also find this vulnerability in ARAN [9]. ARIADNE [12] is another DSR-based protocol that overcomes this attack. There are different implementations of this latter protocol; the first one is based on TESLA, the second uses MACs (Message Authentication Codes), and the most sophisticated uses digital signatures. However, it has been illustrated that this protocol is vulnerable to some fabrication attacks, which cause the construction of non-existent routes [2]. To mitigate this attack, ENDAIRA [11] has been proposed, which is very similar to the last version of ARIADNE. Its idea is simply to sign RREP (route reply) packets instead of RREQ ones. Note that all these secure routing protocols do not handle packet dropping misbehaviour, hence they are vulnerable to the black hole attack, and to the selfish behaviour.

The watchdog [5] is the first solution dealing with the packet dropping problem. Its principle is that each node in the source route monitors its successor using the promiscuous mode. For this purpose, a source routing protocol should be used. This basic solution has the advantage of not requiring any overhead as long as nodes behave well, and it could be applied both to data and control packets. Nevertheless, it is inappropriate when using the power control technique, employed by some new power-aware routing protocols following the watchdog’s proposal [13, 10], such as. Moreover, it does not deal with the isolation step. When a misbehaving node is detected, packets will be sent around it, but no measures will be taken against it, which does not prevent nodes from misbehaving.
CORE [11] and CONFIDANT [14] are among the solutions that mitigate this problem, by defining some reputation and punishment strategies. But these solutions rely on the watchdog technique in their monitor component, thus inherit all its monitoring drawbacks. Moreover, they require periodic exchange of reputation information, which is costly and unnecessary so long as nodes behave well. Another monitoring approach is the employment of a kind of ACK packets known as two-hop ACK [8]. An interesting optimization of this approach is the random asking strategy [15]. All these ACK-based solutions focus on data packets, and are not directly applicable to control packets. In this paper we treat this problem.

III. SOLUTION OVERVIEW

We propose a general solution to monitor, detect, and isolate control packet droppers. We deal with both directed (unicast) and broadcast packets. For the monitoring we propose different approaches for each kind of packets. Regarding the directed packets we suggest the use of the two hop ACK approach [8]. As the number of these packets is too low compared to data ones, the random optimization approach [15] is not efficient. The two-hop ACK is not applicable to broadcast packets, as it becomes too much costly with this kind of packets. Therefore, we propose a promiscuous based solution to monitor control packets. Finally, we propose a redemption strategy for judgment and a reputation-based approach for isolation, applicable to directed packets as well as broadcast ones. However, the optimal values of thresholds used in judgment and isolation may change according to the kind of packets, as illustrated in the next section.

A. Directed Packets: The approach we suggest to use to monitor the forwarding of directed routing control packets (RREP, RERR) needs to be implemented with a source routing protocol. Each node A monitors its successor B in the source route and checks whether this latter forwards to C each packet it provides, such that C is B's successor in the source route and A could be either the source or any intermediate node. This process is repeated on each couple of hops until reaching the final destination. The solution uses a special kind of feedbacks called two-hop ACK [8] that travel two hops. Node C acknowledges packets sent from A by sending this latter via B a two-hop ACK. To ensure authentication of two-hop ACK packets an asymmetric cryptography-based strategy is used. Node A generates a random number and encrypts it with C's public key (PK), then appends it in the packet's header. When C receives the packet it retrieves the number, decrypts it using its secret key (SK), encrypts it using A's PK, and puts it in a two-hop ACK it sends back to A via B. In the first hop (C, B) the ACK is not transmitted in a separate packet, but piggybacked to the ordinary MAC ACK. This inclusion and employment of the MAC ACK reduces the number of two-hop ACK packets as much as half compared with a separate transmission on each hop. When A receives the ACK it decrypts the random number and checks whether it matches with the one it has generated, in order 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 dropping after a timeout. This strategy requires a key distribution mechanism enabling a security association between each pair of nodes. To ensure this distribution, a mechanism like the chain of trust can be used. Note that the same keys could be employed for other security purposes at the other layers. As soon as the monitor node detects that the number of packets dropped by the monitored node exceeds a defined threshold, it considers this latter as misbehaving and proceeds to its isolation.

Contrary to the watchdog, largely used by the current detective solutions, this approach functions well regardless the power control employment[8]. In [15] this solution has been improved by the random asking strategy, in which a monitor node does not continuously ask ACKs but it does so randomly with a coefficient that depends on the behaviour of the monitored node. This strategy is efficient with data packets, and decreases considerably the overhead. However, it is inappropriate with control packets, whose number is too low, and with which we should be more severe. Remember that dropping RREPs (respectively RREQs) prevents a selfish node from being included in routes, while dropping RERRs allows a malicious node to launch a DoS attack by preventing the destruction of broken routes. Also, note that the overhead is not an important issue for this kind of packets, since their number is low.

B. Broadcast Packets: For RREQs packets (which are broadcast) each node monitors every RREQ it forwards or launches as a source. The monitoring starts from the reception of the RREQ (or its launch if the node is the source) and ends after a timeout from its retransmission. For each RREQ, the transmitter monitors all its neighbours. It
should either receive (or overhear) the RREQ or a RREP from every neighbour, except the node from which it received the RREQ if the node is not the source. If no one of these packets is received from a neighbour B, then the monitor notices a packet dropping for B. When a node observes that another node B drops more than the configured threshold number of packets it judges B as misbehaving, and tries to isolate it as we will see later.

C. Redemption: To get over false detections that may occur due to nodes mobility and channel conditions, we propose a redemption strategy for both kinds of packets. The aim is to allow a well-behaving node improving its reputation and tolerance threshold after it has been observed to drop packets due to mobility or collisions. This can be achieved by decreasing the number of packets considered dropped each time it is perceived to correctly forward packets. The pace of decreasing is not inevitably 1, but should be < 1 to prevent nodes from abusing this redemption. That is, forwarding one packet does not decrease the number of packets considered dropped by one. If the pace is m/n (such that m, n ∈ N, m < n), then forwarding n packets decreases the number by m.

D. Isolation: After judging a node as misbehaving, the detector attempts to isolate it. Isolating a misbehaving node means: i) do not route packets through it, to avoid losing them, and ii) do not forward packets for it, to punish it. A node A that judges some other node B as misbehaving should not punish it unilaterally, but must ensure that this will be done by all nodes. This is because when A unilaterally punishes B, the others could consider A as misbehaving when they realize that it does not forward packets for B. In social life, a person that accuses another must show proof. One possible way to prove the accusation is to get witnesses against the accused person. Similarly, to isolate a detected node we suggest the use of a testimony-based protocol, already used with data packets. Upon a detection, the detector informs nodes in its neighbourhood about the dropper (the accused), and asks for witnesses by broadcasting a WREQ (Witness REQuest) packet. It also puts the detected node ID in a special set we call a suspicious set. Each node receiving the WREQ investigates the issue as follows:

i) Directed packets: The receiver of WREQ immediately sends a signed WREP (Witness REPly) packet to the accuser if its suspicious set includes the accused node (denoted by B). Otherwise, if it has not enough experience with the accused node, and if B is its neighbour then it asks the successor of this latter whether it has received packets forwarded from it, by sending an ACREQ (ACCusation REQuest) packet, using a route that does not include B. But first, in order to avoid false accusations, the investigator should ensure that the accuser has really sent a packet to B to be forwarded to the appropriate successor. One possible way to do this is to check whether such a packet has been recently overheard, using the promiscuous mode. The node also should check whether B has sent the accuser an ACK just after overhearing the data, to ensure that the former has really received the packet and that the latter does not impress it. If B's successor has not recently received any packet forwarded from B, it sends a signed ACREP (ACcusation REPly) packet to the investigator, then this latter testifies for the accusation and sends the accuser a signed WREP (Witness REPly) packet.

ii) Broadcast packets (RREQ): In this case the node, if it is a neighbour of B, merely checks whether it has recently received (respectively overheard) either any RREQ forwarded from this node, or a RREP originated from it. To do this, each node keeps the RREQs and RREPs it receives in a buffer for a short time. If neither RREQ nor RREP have been received then it testifies for the accusation and sends the accuser a signed WREP (Witness REPly) packet. But it must first ensure that the accuser node has really recently sent out a RREQ, by checking its buffer. When the detector collects k validation from its neighbours, with at least one provided by direct experience (without asking the successor of B), it broadcasts in the accuser a signed WREP (Witness REPly) packet. But it must first ensure that the accuser node has really recently sent out a RREQ, by checking its buffer. When the detector collects k validation from its neighbours, with at least one provided by direct experience (without asking the successor of B), it broadcasts in the network an accusation packet (AC) containing signatures of all the validating nodes. The requirement of at least one direct witness aims at mitigating wrong accusations caused by false testimonies each node receiving such a valid accusation isolates the guilty.

IV. CONCLUSION

In this paper we proposed a general solution to packet dropping misbehaviour in mobile ad hoc
networks. The solution allows to monitor, detect and isolate the droppers. Usually, the MANET nodes’ mobility causes degradation in efficiency of protocols, but in our case we remarked that it helps improving the true positives of our solutions. The Redemption and Isolation are used to reduce routing overhead and increase the security of routing, resulting in a protocol with performance comparable to that of traditional MANET routing protocols and secure enough for use in high risk environments.

REFERENCES