A new low cost Sessions-based Misbehaviour Detection Protocol (SMDP) for MANET

Tarag Fahad[1], Djamel Djenouri[2], Robert Askwith[1], Madjid Merabti[1]
[1] *School of Computing & Mathematical Sciences, Liverpool John Moores University, UK.*
E-mail: T.M.Fahad@2006.ljmu.ac.uk, {R.J.Askwith, M.Merabti}@ljmu.ac.uk
[2] *CERIST, Basic Software Laboratory, Algiers, Algeria*
E-mail: ddjenouri@mail.cerist.dz

Abstract

There is a strong motivation for a node to deny packet forwarding to others and being selfish in MANET. Recently, some solutions have been proposed, but almost all of these solutions rely on the watchdog technique, which suffers from many drawbacks, particularly when using the power control technique. To overcome this problem with a moderate communication overhead, this paper introduces a new approach for detecting misbehaving nodes that drop data packets in MANET. It consists of two stages: i) the monitoring stage in which each node monitors its direct neighbours with respect to forwarding data packets of a traffic session in the network, and ii) the decision stage, in which direct neighbouring nodes decide whether the monitored node misbehaves or not. Our new approach is able to detect the misbehaviour in case of power control employment, with a low communication overhead compared to the existing approaches.

1. Introduction

The past decade has shown an exceptional growth of wireless communications. One type in particular has huge interest from both research and commercial sides, which is called Mobile Ad hoc Network (MANET). It is a group of autonomous mobile nodes or devices connected through wireless links without the support of a communications infrastructure. The topology of the network changes dynamically as nodes move and the nodes reorganise themselves to enable communications with nodes beyond their immediate wireless communications range by relaying messages for one another, i.e. multihop.

Due to the infrastructure-less feature, all networking functions must be performed by the nodes themselves. Particularly, packets sent between distant nodes are expected to be relayed by intermediate nodes, which act as routers and provide the forwarding service. The forwarding service is closely related to the routing. It consists of correctly relaying the received packets from node to node until reaching their final destination, following routes selected and maintained by the routing protocol. These services (routing and data forwarding) together are at the core of the network layer.

The nature of MANET makes cooperation among nodes essential for the system to be operational. In some MANET applications, such as battlefields or rescue operations, all nodes belong to a single authority (in the application layer point of view) and have a common goal, e.g. soldiers in a military unit or rescuers in a rescue team. For this reason, nodes are cooperative by nature. However, in many commercial applications, such as networks of cars and provision of communication facilities in remote areas, nodes typically do not belong to a single authority and do not pursue a common goal. In such networks, forwarding packets for other nodes is not in the direct interest of anyone, so there is no good reason to trust nodes and assume that they always cooperate. Indeed, nodes try to preserve their resources, and particularly their batteries.

To take this constraint in charge many power-aware routing protocols have been proposed [1] but not all of these solutions eliminate the problem due to the complex nature of the network. As a result, users will be permanently anxious about their limited batteries, which may lead the nodes to behave selfishly. A selfish node regarding the packet forwarding process is the one that takes advantage of the distributed forwarding service and asks others to forward its own packets, but would not correctly participate in this service. This misbehaviour represents a potential danger that threatens the quality of service, as well as one the most important network security requirements, namely the availability. In this paper we introduce a new low cost session-based protocol (SMDP) to monitor data forwarding in MANET, and detect misbehaving nodes.

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

This paper is organized as follows. Section 2 discusses related work including some of the detection mechanisms proposed in literature so far. Section 3 describes the new protocol with an illustrative example. Section 4 provides an analysis of the solution. The protocol as it is described in sections 3 and 4 is only an outline solution to illustrate the basic idea of the work. In section 5 we provide an optimisation of the solution. Finally, section 6 concludes the paper.

2. Related Work

There are two main approaches to dealing with node misbehaviour in MANET [2]. The first approach tries to give a motivation for participating in the network function. A typical system representing this approach is Nuglets [3]. The authors suggest to introduce a virtual currency called Nuglets that is earned by relaying foreign traffic and spent by sending its own traffic. The major weakness of this approach is the demand for trusted hardware to secure the currency. SPRITE [4] has a major advantage in comparison with [3], as it does not require any tamper-resistant hardware. In this solution, virtual money is considered as credits and is not held in packets. However, SPRITE relies on central authority to mange credits, which is not practical in MANET.

Most of the existing work in the field of node misbehaviour concentrates on the second approach which is detecting and excluding misbehaving nodes. Marti et al [5] propose a system which uses a watchdog that monitors the neighbouring nodes to check if they actually relay the data the way they should do. Then a component called pathrater will try to prevent paths which contain such misbehaving nodes. As indicated in the paper, the detection mechanism has a number of severe weaknesses such as its failure to correctly detect the misbehaviour in cases of collisions, partial collusion, and power control employment.

There are other reputation approaches that use the watchdog mechanism in their monitoring component. CORE [6] and CONFIDANT [7] are two examples of such observation approaches. In CORE [6] nodes' observations are propagated beyond the neighbourhood, but only the positive observations. Not propagating negative observations would prevent the vulnerability of propagating rumours aiming DoS attacks, but this way the experience of others gets unused. In contrast, CONFIDANT [7] propagates negative observations beyond the neighbourhood, while considering the rumours problem and taking measures to mitigate it at the trust manager component. Further, CONFIDANT with its modified Baysian approach for reputation, gives less and less importance to past observations, which allows redemption in contrast to CORE which gives more importance to past observations. Nonetheless, in both CORE and CONFIDANT the monitoring component inherit all of the watchdog problems. Moreover, the isolation is performed unilaterally by each node, which might result in false accusation. As when a node isolates unilaterally another and denies forwarding packets for it (punish it), other neighbours would consider its behaviour illegal.

The probing approach first proposed by Awerbuch et al. [8] could be viewed as a combination of route and node monitoring. It uses the end-to-end Acknowledgement (ACK) to monitor routes, and improves it by adding a dichotomic probing phase to detect the appropriate selfish nodes whenever a route becomes suspicious. Iterative probing [2] is more effective but allows to merely detect the link including the selfish node and has high overhead. On the other hand, unambiguous probing [2] deals with the node detection issue, by suggesting utilising the promiscuous monitoring at the predecessor of the suspicious link. This would have inevitably the watchdog's (promiscuous monitoring) problems.

Two-hop ACK [9] allows to detect the selfish nodes and not only unreliable routes, and it enables the usage of the power control technique with no detection problem, contrary to promiscuous monitoring solutions. However, the major drawback of this solution is the significant overhead it generates, although the authors provide some reduction using random acknowledgement approach [9].

3. Solution Overview

Our solution consists of two related stages: the monitoring stage, in which nodes monitor their direct neighbours when forwarding packets, and the decision stage, where nodes decide about the behaviour of each monitored node basing on the result of the previous stage.

3.1. Monitoring Stage

In our solution Sessions-based Misbehaviour Detection Protocol (SMDP) each node in the route session monitors all of its direct neighbours (i.e. neighbours within a one hop communication), and checks whether they correctly forward packets. We define a session as the continuous traffic sent from the source node to the final destination node. The routing protocol has to be aware of the beginning and the end of session. This has been done through cross layer

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

collaboration between the session layer and the network layer. Therefore, our protocol has two components, a session component and a network component. The first one informs the second about the beginning and the end of sessions. All the other operations are performed by the network component. The framework of the new protocol is shown in Figure1.

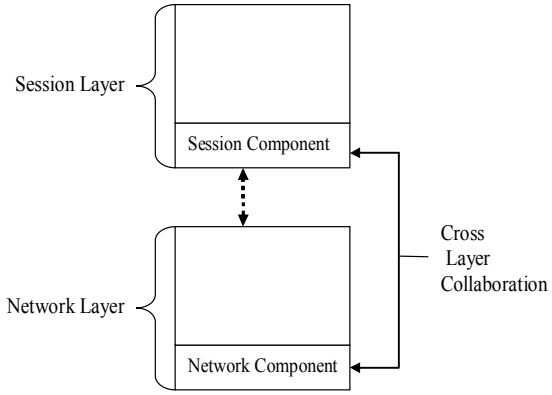

Figure1. Session-Based Protocol's Framework

After the end of each session, each node included in the session (apart from the originated source node and the final destination node) sends two cryptographically *signed* packets. One to its successor containing the number of packets it has sent to it, and the other one to its predecessor containing the number of packets it has received from it. The source node will send only the number of packets it has sent to its successor, and the final destination node will send only the number of packets it has received from its predecessor. After sending and receiving this information, each node broadcasts to all of its one hop neighbours a Forwarding Approval Packet (FAP) shown in figure 2, which is divided into sent/received fields. Each field involves one neighbour participating in the session, and contains the following attributes:

Tij / Rij : Number of packets 'i' has sent/received to/from neighbour 'j'.

id_{Tij} / id_{Rij} : Node Identification number (id) of the sender/receiver node.

S_{Tij} / S_{Rij} : A node signature for authentication.

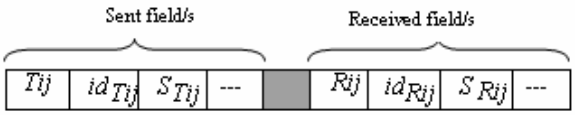

Figure2. The Forwarding Approval Packet (FAP)

After receiving the FAP broadcast from its one hop neighbours, each node checks the authentication of each T_{ij} and R_{ij} in the FAP.

Then for a set of nodes I that surround a single neighbour j, if there are no packets dropped the following holds:

$$\sum_{i \in I} T_{ij} = \sum_{i \in I} R_{ij} \qquad (1)$$

3.2. Decision Stage

The new monitoring method allows the neighbouring nodes to judge whether each monitored node in the session has forwarded packets correctly or not. From equation (1) we consider the following:

$$\sum_{i \in I} T_{ij} = T \ \& \ \sum_{i \in I} R_{ij} = R$$

If $R - T = 0$ then the node is forwarding packets correctly. Otherwise, ($R - T$) packets has been dropped and the node will be considered as a suspicious node but not misbehaving. Since this dropping could be caused by collisions or nodes' mobility, the monitoring neighbouring nodes will not directly accuse the monitored node as misbehaving. Therefore, a threshold of tolerance is crucial and should be fixed carefully to allow a fair decision. If we have t as a threshold, then when $R - T > t$, the node will be considered as a misbehaving.

3.4. Example

Consider the following example in figure 3 where an ad-hoc network shown as a set of 25 nodes (5x5 nodes) in a squared grid surface. Nodes mobility is supposed to be low enough at this stage so that relative position of nodes doesn't vary during the sessions.

There are two sessions running, the first one shown as a solid arrow in figure 3, starts at n1 (session source) and ends at n20 (session final destination). The total number of packets sent from n1 to be delivered to n20 is 60. The 60 packets are sent form n1 to n7 where n7 will forward them to n13, and then 20 packets go through n14 and the remaining 40 go through n19.

Thus, the total number should arrive to the final destination n20 is 60 packets. The second session shown as a dashed arrows in figure 3, starts form n5 (session source) and ends at node n21 (session final destination). The total number of packets sent from n5 to be delivered to n21 is 70. Node 5 sends the 70 packets to node 9 to forward them to node n13, then from n13 to n17 and finally the latter forward them to the session final destination n21. In this example we are applying our solution to detect misbehaviour in the first session only.

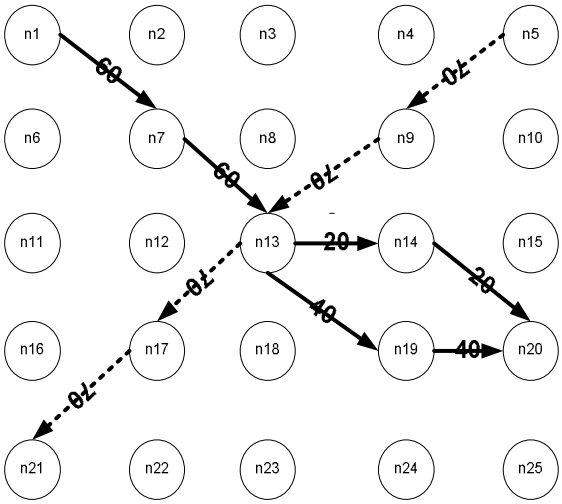

Figure3. MANET two sessions example

After the end of the first session which started at n1, node n13 will send the following signed packet to node 19:

Tx	40	n13	S13

Such that, Tx is the type of the packet (Tx stands for a packet that includes the number of packet sent and Rx for a packet that includes the number of packets received), 40 is the number of packets sent from node n13 to n19, n13 is the ID of the sender, and finally S13 is a signature of node n13 applied on the packet.
Node n13 will also send the following signed packet to node n14:

Tx	20	n13	S13

And finally it will send the flowing singed packet to its predecessor n7:

Rx	60	n13	S13

Node n13 will also receive the following packet from n7:

Tx	60	n7	S7

And the following packet from n14:

Rx	20	n14	S14

And finally the following packet from n19:

Rx	40	n19	S19

At this stage, n13 will broadcast the following FAP to all of its direct neighbours:

40	n19	S19	20	n14	S14		60	n7	S7

When receiving this packet, neighbouring nodes will check first the authentication of each T_{ij} and R_{ij} in the FAP. Then they will calculate the following:

$$\sum_{i \in I} T_{ij} = 40+20 = 60, \quad \sum_{i \in I} R_{ij} = 60$$

Based on the above calculations it is obvious that $\sum_{i \in I} T_{ij} = \sum_{i \in I} R_{ij}$, in all of the three direct neighbours.

Based on this, neighbouring nodes of node 13 will decide that this latter is forwarding packets correctly without any dropping. On the other hand, the same nodes i.e. n7, n19 and n14 will build FPA packets using the packets sent from n13 and their neighbours as well, then broadcast them. Subsequently, they will be evaluated by their neighbours in the same way that n13 has been evaluated.

4. Analysis

The overhead of SMDP is proportional to sessions, independently of the number of packets transferred in the sessions.

First, we consider the previous example in figure3, then we will generalise the results to infer the communication complexity. In the first session, i.e. the one starting at n1 and ends at n20, there are 6 nodes participating in this session. Let this number be denoted by *h*. At the end of the session, each of the nodes n1, n7, n13, n14 and n19 will send a packet to its

successor containing the number of packets it has sent. This makes a total of 5 packets, which is *h-1*.
Also, each of the nodes n7, n13, n14, n19 and n20 will send a packet containing the number of packets it has received from its predecessor. Overall, 5 packets of such a kind will be sent. That is, *h-1*. After receiving the Rx and Tx packets (explained in section 3.4), each of the intermediate nodes (n7, n13, n14 and n19) builds and broadcasts a FAP packet to its direct neighbours, resulting in 4 transmissions, which is h-2.

Generally speaking, we have:
- *h-1* packets containing the number of packets sent, i.e. all the nodes, except the destination, send one packet including such an information.
- *h-1* packets containing the number of packets received, all the nodes, except the source, send one packet including such an information.
- *h-2* FAP packets. That is, every intermediate node (neither the source nor the destination) broadcasts such a packet. Overall, we have *3h-4* transmissions which is in term of complexity:

$$\approx O(3(h-1))$$

As we have seen, SMDP is operational when employing the power control technique, contrary to the watchdog-based solutions. Compared with the random two-hop ACK [9], which is also operational with the power control technique, our solution is low cost. The communication complexity of that solution is:

$$O((h-1)nP_{trust}).$$

Such that n is the number of packets, and P_{trust} is an intrinsic parameter of the solution. The mathematical study performed in [9] illustrates that the best value of this latter is 0.5. Thus, our solution outperforms this one (in term of overhead reduction) upon 6 packets/session. Thus, the reduction factor of the communication overhead is n/6.

As for probing, the communication complexity of that solution is:

$$O((h-1)n).$$

Our solution outperforms it upon 3 packets/session. The reduction factor of the communication overhead is n/3.

5. Optimisation

Our solutions can be optimised even further to reduce the communication overhead. This can be done by using aggregating sessions. When using this approach, nodes that are involved in more than one session could wait a certain time until all sessions end before sending the FAP to their direct neighbours. For examples n13 in figure 3 could wait until both sessions end, then send one aggregated FAP to its neighbours regarding the two sessions, instead of sending two FAPs separately. The aggregated packet is:

40	n19	S19	20	n14	S14	70	n17	S17
60	n7	S7	70	n9	S9			

In this way, we could reduce the communication overhead. This will result in increasing efficiency and applicability to MANET.

6. Conclusion

This paper introduces a new low cost approach for monitoring node misbehaviour in MANET. Unlike other monitoring approaches such as [5] [6] [7], our approach is able to detect the misbehaviour in cases of power control employment. It is also cost effective as it reduces the communication overhead, by using only one hop communication (no flooding), and sending control packets only at the end of sessions, instead of doing so for each packet, contrary the current solutions that exist in literature ([5], [6], [7] [8] and [9]). In our perspective, we plan to deal with the fixation of the accusation threshold, and to complete the solution with the isolation stage. Also, we plan to extend the solution to support nodes' mobility during the session, and to evaluate the performance of our approach by simulation.

7. References

[1] H Yang, H Y. Luo, F Ye, S W. Lu, and L Zhang. *Security in mobile ad hoc networks: Challenges and solutions.* IEEE Wireless Communications, 2004. **11**(1): p. 38-47.

[2] Frank Kargl, Andreas Klenk, Stefan Schlott, and Michael Weber. *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. Heidelberg, Germany.

[3] Buttyan, L. and J.-P. Hubaux, *Stimulating Cooperation in Self- Organizing Mobile Ad Hoc Networks*. ACM/Kluwer Mobile Networks and Applications, 2003. **8**(5).

[4] S. Zhong, J. Chen, and Y. R. Yang, "SPRITE: *A simple, cheat-proof, credit-based system for mobile ad-hoc networks*. in The 22th IEEE Annual Joint Conference on Computer Communications and Networking INFOCOM'03, San Franciso, CA, USA, April 2003.

[5] S. Marti, T. Giuli, K. Lai, and M. Baker, *Mitigating routing misbehaviour in mobile ad hoc networks.* Mobile Computing and Networking, 2000: p. 255-265.

[6] Michiardi, P. and R. Molva. *CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks*. in *Communication and Multimedia Security*. 2002. Portoroz, Slovenia: Kluwer Academic.

[7] Buchegger, S. and J.-Y.L. Boudec. *Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness in Distributed Ad-hoc Networks.* in *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*. 2002. Lausanne.

[8] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. *An on-demand secure routing protocol resilient to byzantine failures*," in ACM Workshop on Wireless Security (WiSe), Atlanta, Georgia, USA, September 2002.

[9] D.Djenouri and N.Badache. *Cross-layer Approach to Detect Data Packet Droppers in Mobile Ad-hoc Networks*. In Proceeding of the first International Workshop On Self-organized systems IWSOS'06, Passau, Germany, ser. LNCS no 4124, pp 163-176, Springer-Verlag GmbH Publisher, September 2006.