Third International Symposium on Information Assurance and Security

On Detecting Packets Droppers in MANET: A Novel Low Cost Approach

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

Abstract

One of the commonest threats that mobile ad hoc networks are vulnerable to is data packet dropping, which is caused either by malicious or selfish nodes. Most of the existing solutions to solve such misbehaviour 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 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 the decision stage, in which direct neighbouring nodes decide whether the monitored node misbehave 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 battlefield 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, but none of these solutions eliminate the problem due to the complex nature of the network. As a result, users will be permanently worried 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 Misbehaviour Detection Protocol

0-7695-2876-7/07 $25.00 © 2007 IEEE
DOI 10.1109/IAS.2007.47

(SMDP) to monitor data forwarding, and detect packet dropping nodes in MANET. Our solution takes advantage of a cross-layer design, and exploits information related to the session layer that makes its control packet transmissions proportional to sessions, which reduces the communication overhead. At the end of a session, each forwarder node shows to its neighbours the number of packets it received from each other during the session, as well as the total sent, by sending a special packet we call Forwarding Approval Packet (FAP). Mechanisms to ensure authentication of such information and to prevent nodes from denying receptions of data packets are used. Nodes then collaboratively analyze the FAPs, and judge one another.

The rest of the paper is organised 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 [1] [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 introducing 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.

The power control technique has been used by many routing protocols proposed after the watchdog's proposal in the field of power consumption optimization, e.g. [11]. By using the power control technique, nodes in MANET can preserve their power, by only transmitting packets from one node to another using controlled power according to the distance separating them from each other. For example, in the watchdog, when node C is closer to node B than A, and when B transmits packets using controlled power according to the distance separating it from C, A could not overhear B's forwarding, and may accuse it wrongly.

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 Bayesian 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 further reduction using random acknowledgement approach [9]. A review of existing solutions is described in [10].

3. The New Proposed 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 each session. This has been done through cross-layer 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.

After the end of each session, each node included in a path used by 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, we denote by NPS, and the other one to its predecessor containing the number of packets it has received from it, denoted NPR.

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.

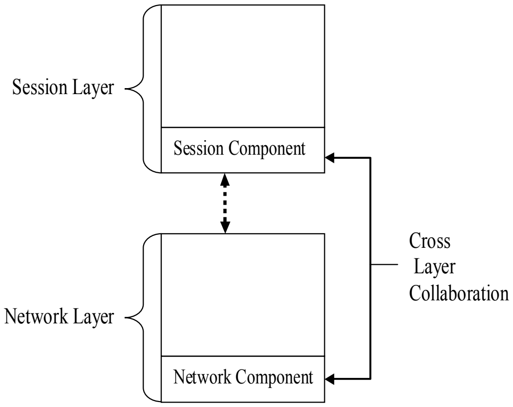

Figure1. Session-Based Protocol's Framework

NPR and NPS contain the sequence number of their sender, which is a number maintained by each node and monotonously increased (by 1) after including it in a packet. This prevents using an NPS or NPR more than once. After sending and receiving this information, each node builds and 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 node 'i' has sent/received to/from neighbor 'j'.

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

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

mj: The sequence number of node j.

SENT fields						RECEIVED fields				
Tij	id_{Tij}	mj	S_{Tij}	---	▨	Rij	id_{Rij}	mj	S_{Rij}	---

Figure2. The Forwarding Approval Packet (FAP)

After receiving a FAP broadcasted from its one hop neighbour, each node checks the authentication of each T_{ij} and R_{ij} in the FAP through digital signatures. It also checks that none of the sequence number has already been used. For this it keeps the last sequence number of each other node, so that the new received number should be greater than the previous one. Any failure in one of the previous verifications results in considering the appropriate number of packets to be zero, meaning do not accept such information.

If there are no packets dropped the following equation holds:

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

Thus far, nodes are assumed to not deny the sending and the reception of packets, and accordingly they correctly send the NPS and notably NPR packets, and include all the receptions in the FAPs as well. Now we deal with situations where selfish nodes lie. Assume that there is no more than one such a node in a neighborhood, and we do not consider collusions. Also, we suppose that when a selfish node starts misbehaving, it will do it continuously, and does not drop packets selectively. Finally, we point out that we are dealing with selfish nodes, but not with malicious attackers. If a well-behaving node does not receive NPR or NPS from a neighboring node, it simply lets the corresponding signature field empty in the FAP it sends. The neighbors receiving such a packet with an empty signature assume that either the node of the appropriate field or the FAP sender is misbehaving. They keep their Ids for further investigations. This will be enhanced in the following.

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. We first deal with the situations where nodes do not lie, and all the required signatures are put in the FAP. From equation (1) we consider the following:

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

If $R - T = 0$ then the node is forwarding packets correctly. Otherwise, ($R - T$) packets has been dropped and the node will be charged of dropping this number of packets. Since this dropping could be caused by collisions or nodes' mobility, the monitoring nodes will not directly accuse it 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.

Now we treat the cases where a FAP's SENT field regarding some node *X* lacks a signature. Lack of a signature in a RECEIVED field is of no impact if the sender of the FAP has correctly forwarded packets and shows proofs (signatures in the SENT fields). The previous sums (*T* and *R*) are calculated as before, and if *R-T>0,* this number (*R-T*) of packets will be considered dropped. But in addition, the node will not be immediately considered forwarding the *T* packets. In fact, either *X* is denying the reception of packets, or the sender of the FAP has dropped packets and is lying. The two nodes IDs as well as the appropriate number of packets (claimed in the SENT field that lacks a signature) are safeguarded in what we call the suspicious set. Later, if one of these two nodes will be considered as a suspicious in another experience, it will be charged of dropping packets (both in the first and the second experiences), and the innocent's id will be released from the suspicious set. This will be clarified in the following example.

3.3 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. Node mobility is supposed to be low enough so that relative positions of nodes do not 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), and includes in total 60 packets. These packets are sent from n1 to n7, which forwards them to n13, and then 20 packets are routed through n14 and the remaining 40 through n19. 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 of this session 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 forwards them to the session final destination n21.

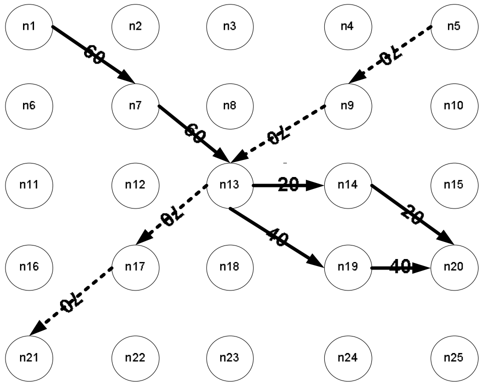

Figure3. MANET Two Sessions Example.

Suppose all nodes are well-behaved. After the end of the first session which starts at n1, each of the nodes n7, n13, n19, n14, sends a signed packet including the number of packets it has received, and another including the number of packets it has sent. n1 sends only the number of packets it has sent (it does not receive any packet as it is the originated source), while n20 sends only the number of packets received (as it is the final destination).

After receiving from its neighbours the number of packets it has received and sent, n13 will broadcast the following FAP:

40	n19	m19	S19	20	n14	m14	S14	▨
60	n7	m7	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:

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

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 build FAP 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. Note that thanks to the sequence number, fields used to construct the FAP cannot be reused. For instance, in a future session involving nodes 13 and n19, the former cannot drop packets and reuse the field (40, n19, m19, S19), as when neighbours receive such a field they remark that m19 has not increased, and consequently do not accept that n13 forwarded 40 packets to n19.

Now we consider the situation where node 13 is selfish. It drops packets received from n7, then it can either put a field with an empty signature, or simply deny the reception of packets from n7 (not sending FAP, neither NPR to n7). Note that it cannot claim forwarding packets to both n19 and n14 with empty signatures, as in this case it will be suspicious simultaneously with the two nodes, thus it will be immediately detected. Assume it claims forwarding the 60 packets to one of the nodes, such as n14. It then sends the following FAP:

60	n14	m14		▨	60	m7	n7	S7

When receiving such a packet, the neighbours will put nodes n14 and n13 in their suspicious set, along with the number 60. Next, when n13 drops packets of the second session, during which it receives packets from n9, either by sending a FAP with an empty signature regarding n17, or simply denying the reception from n9 and not sending neither the NPR to n7 nor the FAP. In the first case it will be suspicious with n17 then immediately detected by neighbours, after checking their suspicious sets. Node n17 will not be put in the suspicious sets in this case, and n14 will be removed from the sets. Whereas in the second case, it will be suspicious with n9 when this latter sends its FAP including n13 with an empty signature in the SENT field. n13 will be charged instead of n9, and n14 will be released. In the two cases, n13 will be charged of dropping 130 packets (the sum of the numbers of the two sessions 70+60). If in the earlier session n13 denies the reception of packets from n7, it will be simply suspicious with this latter (instead of n14), when it send its FAP including a SENT field regarding n13 with an empty signature. Identically to the previous scenario, n13 will be detected and n7 released at the end of the second session.

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 mentioned, 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) by 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 by 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 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 example, n13 in figure 3 can wait until both sessions end, then sends one aggregated FAP to its neighbours regarding the two sessions, instead of sending two FAPs separately. The aggregated packet is:

<table>
<tr><td>40</td><td>n19</td><td>m19</td><td>s19</td><td>20</td><td colspan="2">n14</td><td colspan="2">m14</td><td colspan="2">S14</td><td>70</td><td>n17</td></tr>
<tr><td>m17</td><td>S17</td><td></td><td>60</td><td>n7</td><td>m7</td><td colspan="2">S7</td><td colspan="2">70</td><td>n9</td><td>m9</td><td>S9</td></tr>
</table>

In this way we reduce the communication overhead. This optimization is beneficial for well-behaving nodes. A selfish node, however, has no interest of aggregating FAPs, since lying in such a packet will inevitably include two nodes, which allows to directly detect it.

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]). For future work, we plan to deal with the fixation of the accusation threshold, and to complete the solution with the reaction/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 22nd IEEE 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, 2006 **.**

[10]D. Djenouri, L. Khelladi, and A.N. Badache, "*A survey of security issues in mobile ad hoc and sensor networks",* Communications Surveys & Tutorials, IEEE, 2005, **7**(4): p. 2- 28.

[11] S. Doshi and T. Brown, *Minimum Energy Routing Schemes for a Wireless Ad Hoc Network*, *IEEE INFOCOM'02*, New York City, USA, 23–27 June 2002.