Session-Based Misbehaviour Detection Framework for Wireless Ad hoc Network

Tarag Fahad[1], Robert Askwith[1], Djamel Djenouri[2], Madjid Merabti [1]

[1] School of Computing & Mathematical Sciences, Liverpool John Moores University,
Byrom Street, Liverpool L3 3AF. UK.
T.M.Fahad@2006.ljmu.ac.uk, {R.J.Askwith, M.Merabti}@ljmu.ac.uk

[2] CERIST, Basic Software Laboratory, Rue des 3 freres Aissou, Ben-Aknoun BP 143.
Algiers, 16030, Algeria.
ddjenouri@mail.cerist.dz

Abstract: Data packet dropping misbehaviour is a serious threat in ad hoc networks. The motivation of such behaviour is either to take illegal advantage and preserve resources (selfish) or to intentionally cause harm (malicious) to some victims. 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 Sessions-based Misbehaviour Detection Framework (SMDF) for detecting misbehaving nodes that drop data packets in stationary wireless ad hoc network. It consists of three components, the detection component in which each node monitors its direct neighbours with respect to forwarding data packets of a traffic session in the network, and the Decision component, in which direct neighbouring nodes decide whether the monitored node misbehaved or not and finally the isolation component where the guilty node will be penalized. Simulations results show that SMDF is scalable and able to detect the misbehaviour with high accuracy at low communication overhead and low energy consumption compared to the existing approaches.

Keywords: Wireless Ad hoc Network Security, Wireless Sensors Network, Nodes Misbehaviour, Malicious Attacks, Misbehaviour Detection.

1. Introduction

Nowadays we witness an exceptional growth of wireless communications. One type in particular has huge interest from both research and commercial sides, which is called wireless Ad hoc Network. It is a group of autonomous mobile nodes or devices connected through wireless links without the support of a communications infrastructure. Nodes can be static e.g. sensors network or freely mobile.

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 wireless ad hoc network makes cooperation among nodes essential for the system to be operational. In some ad hoc network 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 of the most important network security requirements, namely the availability. In this paper we introduce a new low cost Session-based Misbehaviour Detection Framework (SMDF) to overcome nodes misbehaviour in terms of dropping data packets in stationary wireless ad hoc network. 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 a simulation study of the solution, followed by a summarized comparison with related work in section 5 and some discussions about the solution limitations in section 6. Finally, section 7 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 inherits 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 to the misbehaviour problems in wireless ad hoc network is a new Session-based Misbehaviour Detection framework (SMDF) [15]. It consists of three new components integrated together to detect and deal with nodes misbehaviour in ad hoc network. The first and most important component of the framework is the novel detection component. For this component we have developed a novel Session-based Misbehaviour Detection Protocol (SMDP) [13, 14]. The second component of the new framework is the decision component which will judge whether the nodes misbehave intentionally or not. The third and final component of our framework is the isolation component which will penalize nodes who are judged to have misbehaved. Figure 1 shows our framework SMDF and its components.

3.1 Detection Component

In our approach [13,14,23,24] 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, shown in figure 2. Cross-layer is a paradigm in wireless network architecture design that takes into accounts the dependencies and interactions among layers, and supports optimisation across traditional layer boundaries [16]. In our framework it means the exchange of information between the session layer and the network layer. As a result, our protocol has two components, a session component and a network component.

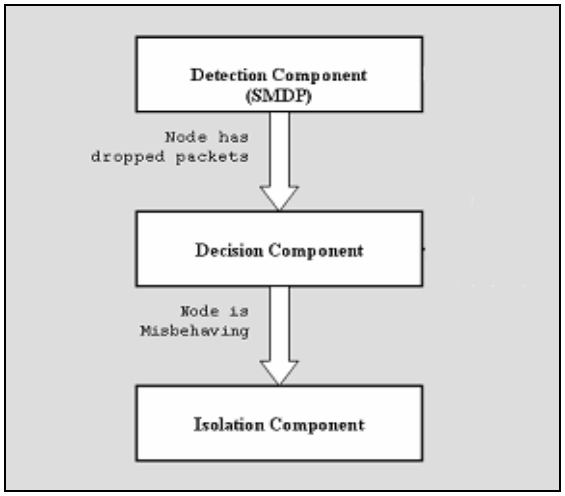

Figure1. Session-Based Protocol's Framework

The first one informs the second about the beginning and the end of sessions. All the other operations are performed by the network component. In our solution (SMDF) [13] we monitor nodes only after the end of the session contrary to all of the other existing approaches such as [5, 7, 1, 6, 17,18, 9] where monitoring happens immediately after the node sent packets to its successor to forward them further in the network. By using sessions based approach we will save a considerable amount of communication overhead and subsequently reduce the cost.

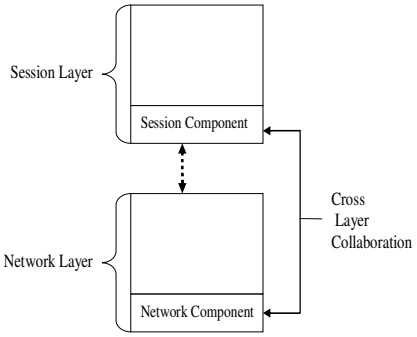

Figure2. Session-Based Protocol's Framework

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* (i.e. using asymmetric encryption) 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 NPS to its successor, and the final destination node will send only the number of packets it has received NPR from its predecessor. NPR and NPS contain the sequence number of their sender, which is a number maintained by each node and monotonically 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 4-3, 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.

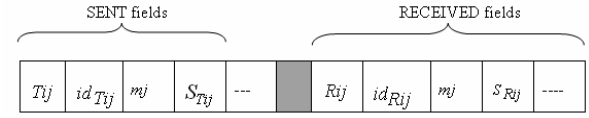

Figure3. The Forwarding Approval Packet (FAP)

Note that contrary to almost all the other solutions, our new framework can work independently of the routing protocol, as it does not need to know the two-hop neighbor to monitor its successor. It does it locally with its neighbours.

3.1.1 Detection Component Case Study 1 (well-behaved nodes)

To illustrate how our novel monitoring approach works consider the following case study in figure 4 where an ad-hoc network is 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 5-2, 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 is shown as dashed arrows in figure 5-2, 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.

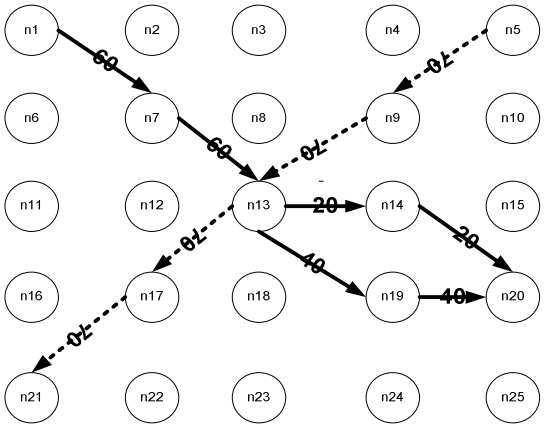

Figure 4: MANET Two Sessions Case Study

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

Tx	40	n13	m13	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.

n13 will also send the following signed packet to node n14:

Tx	20	n13	m13	S13

And finally it will send the following signed packet to its predecessor n7:

Rx	60	n13	m13	S13

Node n13 will also receive the following packet from n7:

Tx	60	n7	m7	S7

And the following packet from n14:

Rx	20	n14	m14	S14

And finally the following packet from n19:

Rx	40	n19	m19	S19

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 detect 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.

3.1.2 Detection Component Case Study 2: (Selfish and Liar Nodes)

Note that thanks to the sequence number, fields used to construct the FAP cannot be reused. For instance, in a future session involving nodes n13 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.

3.1.3 Optimised SMDF Using Sessions Aggregation

Our solutions SMDF 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 4 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:

40	n19	m19	s19	20	n14	m14	S14
70	n17	m17	S17	60	n7	m7	S7
70	n9	m9	S9				

In this way we reduce the communication overhead even further. This optimisation 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.

3.2 SMDF Detection Component

After receiving a Forwarding Approval Packet FAP (described previously) broadcasted from its one hop neighbour, our detection component through our Session-based Misbehaviour Detection Protocol (SMDP) will start working. Each node checks the authentication of each T_{ij} and R_{ij} in the FAP using digital signature. 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 neighbourhood, and we do not consider collusions. 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 neighbouring node, it simply leaves the corresponding signature field empty in the FAP it sends. The neighbours 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.

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:

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

If $R-T=0$ then the node is forwarding packets correctly. Otherwise, ($R-T$) packets has been dropped.

Now we treat the cases where a FAP's SENT field regarding some node, for example *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 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.

We have used Bayesian approach for our new decision stage. Our new proposed Bayesian approach is similar to that used in [12,19] but using SMDP as the monitoring component. The monitor allows the neighbouring nodes to decide whether each monitored node in the session has forwarded packets correctly or not. Therefore, when a monitoring node notices that some packet has been dropped over a link it should not directly accuse the monitored as misbehaving, since this dropping could be caused by collisions or channel conditions. Therefore, a threshold of tolerance should be fixed.

In the Bayesian approach, well behaving of nodes improves their reputation, whereas intentional or unintentional packet dropping decreases it. Since misbehaving is usually exception rather than the norm, information exchange in our solution is limited to negative impressions. It is simpler and creates no overhead when nodes well-behave, as in [12].

Each node A thinks that each other node B misbehaves with a probability θ, which is a random variable estimated by a Beta distribution $Beta\ (a,b)$ described above. Initially with no prior information, θ is assumed uniform in [0,1], which is identical to $Beta(1,1)$. As observations (that follow a Bernoulli distribution with a parameter θ) are made, a and b are updated as follows:

$$a = a + (R-T), b = b + T$$

Where R is the number of packets received by the monitored node (as a router), and T is the number of packets forwarded by it during the session, as mentioned in our detection component. The previous sums (*T* and *R*) are calculated as before in our detection component, and if *R-T>0,* this number (*R-T*) of packets will be considered dropped.

After as many observations as the decision could be made (θ could be approximated by the mathematical expectation $E(Beta(a,b))$), the node will be judged. This is denoted by the decision (or stationary) point, while the number of observations is expressed by $a+b$. Upon reaching this point, B will be accused of misbehaving as soon as: $E(Beta(a,b)) > E_{\max}$.

Note that: $E(Beta(a,b)) = a/(a+b)$.

$E_{\max}$ could be fixed to 0.5 (i.e. 50% of misbehaviour. In mathematical estimation methods, the decision (stationary) point is the one upon which the difference between two subsequent observations could be negligible. One usual choice is that fulfilling the following condition:
$Var(Beta(a,b)) < \varepsilon$.

Such that Var is the mathematical variance and ε is a very small positive.

Note that:

$$Var(Beta(a,b)) = \frac{a \times b}{(a+b+1) \times (a+b)^2}$$

However, this choice is inappropriate here, since $Var(Beta)$ is not monotonic with $a+b$. We use the following variance like function, which is indeed decreasing with $a+b$:

$$Max\left(\frac{b}{(a+b)\times(a+b+1)}, \frac{a}{(a+b)\times(a+b+1)}\right)$$

When enough observations with regard to a given monitored node are collected such that the judgment point is reached, the monitoring node will accuse the monitored one as soon as the estimated probability ($E(Beta(a,b))$) exceeds the configured maximum tolerance threshold, i.e. $E(Beta(a,b)) > E_{\max}$.

$$E(Beta(a,b)) > E_{\max} \longleftrightarrow \frac{a}{a+b} > E_{\max} \longleftrightarrow a > \frac{b \times E_{\max}}{1-E_{\max}}$$

This latter $\left(\frac{b \times E_{\max}}{1-E_{\max}}\right)$ represents the tolerable number of packets a node is allowed to drop without being accused. This maximum tolerable threshold is proportional to b, the number of packets forwarded. The more a node forward packets, the more its tolerable threshold increases. Forwarding packets after unintentional or intentional droppings that do not result in an accusation would decrease *E*, which allows redemption. This redemption could not be possible when setting the tolerable threshold to a fixed number of packets. In our SMDF the redemption is just before decision. A node that forwards packets will need much more packets to be dropped before being accused compared to the one that does not forward, so it is like the forwarding redeem its dropping. However, there is no redemption after the decision.

In [19], every node periodically broadcasts in its neighbourhood its view of θ regarding all the other nodes. Nodes use these information (known as second hand information) to update their own opinion on nodes' behaviour. To decide about the acceptance of the provided information, each node performs complicated tests on the trustworthiness of the provider. The problem with this proactive solution is that it causes an increase in the amount of overhead generated, even if nodes well-behave. Our approach is rather reactive, thus no such information are exchanged. Indeed, each node performs monitoring separately and informs the others as soon as a misbehaving node is approved, as we will see in the next section with more details.

3.3 SMDF Isolation Component

Our Isolation Component used the social sciences principle that a person that accuses another of misconduct must show proof. One possible way to prove the accusation is to get observers against the accused person. In order to mitigate false detections and false accusations vulnerability, we have used Observation-Based Protocol similar to that used in [12,22] but with the advantages of not using promiscuous mode. In this protocol, a node that detects and accuses another as misbehaving must approve its accusation before taking any measure against it. It should not isolate the assumed misbehaving unilaterally, because this could result in false detections against it. However, it could avoid routing its own packets through this node. Isolating a misbehaving node in wireless ad hoc network required two actions. First, not to route packets through it, to avoid losing them; second, do not forward packets for it, in order to punish it. For example, node A that judges some other node B as misbehaving should not isolate it unilaterally, but must ensure its isolation by all nodes. This is because when A unilaterally isolates B, the others could consider A as misbehaving when they realize that it does not forward packets for B. The way that our proposed Observation-Based Protocol work is describe as follow: Upon detection, the detector informs nodes in its neighbourhood about the dropper (the accused), and asks for observers by broadcasting an Observation REQuest (OREQ) packet. It also puts the detected node ID in a special set called a *suspicious set*. Each node receiving the OREQ investigates the issue as follows:

The packets recipient immediately sends a *signed* Observation REPly (OREP) packet to the accuser if the accused node's misbehaving expectation is close to $E_{\max}$, or the number of control packets considered dropped is close to the configured maximum threshold. Also if its suspicious set includes the accused node. Otherwise, when it has not enough experience with the accused node (B), 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 ACcusation REQuest ACREQ 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. 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 is not impersonating 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 Observation REPly- OREP packet. The signature of the packets prevents their spoofing, thus no node could testify using the ID of another.

The accuser node has to collect 'S' different signatures to approve its accusation. Theoretically, $S-1$ is the maximum number of misbehaving nodes that could exist at any time. In practice, however, it is hard to determine such a number, so it should be fixed to strike a balance between efficiency and robustness. Setting S to a high value increases the robustness of the protocol against false detections and rumours, but decreases its efficiency regarding true detections. On the other hand, a low value of S allows high detections, but opens the vulnerability of rumours and increases the unintentional false detections (false positives), since S nodes could collude to accuse maliciously (respectively wrongly) any node. Once the accuser collects S valid signatures, it broadcasts an Isolation Packet (ISOP) including all signatures through the network to isolate the guilty. This broadcast is not performed until a node is detected and approved as misbehaving. Apart from the monitoring stage, our solution requires no overhead as long as nodes well behave, as no opinions are exchanged periodically. This gives our solution the advantages of being a reactive one, unlike the other reputation-based solutions that were presented before.

4. Simulation Study

In this section we describe the performance evaluation of our SMDF and we discuss different simulation scenarios that show what happens when we modify the initial system state. We outline the metrics and parameters within our simulator that are the container for the initial data set for any scenario.

4.1 Simulations Parameters

To study the effect of node misbehaviour on wireless ad hoc network and to assess the performance of the proposed detection protocol, we have developed a GloMoSim-based [20] simulation study. We have simulated a network of 100 nodes, located in an area of $2500 \times 2000\ m^2$ where nodes are deployed randomly for 1800 seconds of simulation time. To generate traffic we have used five Constant Bit Rate (CBR) sessions between five pairs of remote nodes, each consists of continually sending a 512 byte 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.

Table 1 shows the important simulation parameters that have been used in our simulation. These parameters are typical for wireless ad hoc network simulations (see e.g.[21]) and are used for all following simulations. For the results of the simulation to be meaningful, it is important that the model on which is based the simulator matches as closely as possible the reality. Various examinations, such as [22], show significant divergences between different simulators that demonstrate an identical protocol. Therefore, the results obtained from the simulations should be evaluated appropriately.

Parameter	Value
Number of Nodes	100
Area X (m)	2500
Area Y (m)	2000
Traffic Model	Constant Bit Rate (CBR)
Sending Rate (Packets/S)	1.0
Packet Size (Byte)	512
Simulation Time (S)	1800
Node Placement	Random

Table 1: Simulation parameters

4.2 Simulation Metrics

We evaluate our proposed SMDF using the following six metrics:

- **Packet delivery ratio:** This is the percentage of sent data packets actually received by the intended destinations over the submitted packets.
- **Overhead:** This is the amount of control-related transmissions (control packets including FAPs) measured in bytes, and generated during each session in the network. We count the amount of the actual control packets in bytes instead of the number of packets, because it reflects the real amount of overhead, as you might have small number of packets that generates huge amounts of bytes and vice versa.
- **True Positive Detection Rate:** The rate of true dropping detection, when nodes correctly detected dropping packets.
- **False Positive Detection Rate:** The rate of false dropping detection, when nodes wrongly accused of misbehaviour, when in fact they are not.
- **Power Consumption Rate:** The average amount of power consumed by the nodes during the simulation.
- **Scalability:** How scalable the new protocol is if the network number of nodes increased.

4.3 Simulation Results

In this section we present our simulation results. We evaluate the proposed mechanism using simulation

techniques by determining the utilization level of network resources achieved using them, and by comparing our results based on simulation models to the best possible deterministic schemes available on the literature.

4.3.1 Evaluation of the Effect of Node Misbehaviour on Wireless Ad hoc Network Packet Delivery Ratio

In order to study how node misbehaviour affects a wireless ad hoc network performance, we have done a number of simulations where we modelled a varying number of selfish nodes. In order to compare the affect of node misbehaviour in the network, we first run the simulation without selfish nodes (i.e. all of the nodes in the sessions are behaving correctly and forwarding packets as required from them without any dropping). Next, we run the simulation and in this case we have injected the network with selfish nodes that misbehave by not forwarding packets they received from other nodes. We have varied the number of selfish nodes from 1 to 20 nodes of the total number of nodes, which is 100. Figure 5 shows the results of these simulations. It is obvious that this number has a significant effect on the rate of packets that are successfully delivered in the network. In this simulation we have used DSR routing protocol, the selfish node has not been detected by DSR and no countermeasures are taken.

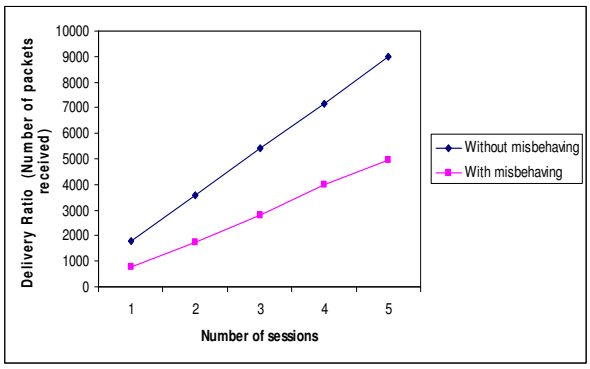

Figure 5: Node Misbehaviour Effects on wireless ad hoc network's Packet Delivery Ratio

4.3.2 Evaluation of the Effect of Packet Dropping Attack in Wireless Ad hoc Network

In this simulation we are targeting to simulate and study the affect of packet dropping attack in wireless ad hoc network. Unlike the previous simulation in figure 5 where the dropping rate fixed to 50%, in this simulation the dropping rate varies form 0% to 100%. We simulated 20 nodes launching this attack by different rate of dropping as shown in figure 6. It can be seen from figure 5 that when the dropping rate of the attacker is low, the throughput (i.e. number of packet received) is high. As the dropping rate

increased the throughput is severely affected until it reaches 0 as the attacking nodes increased their dropping rate to 100%. This clearly shows the affect of such attack on the performance of wireless ad hoc network and wireless sensor network. The result shows that the more number of such malicious nodes inside the network, the more the harmful impact on the reliability, and that the decline is dramatic.

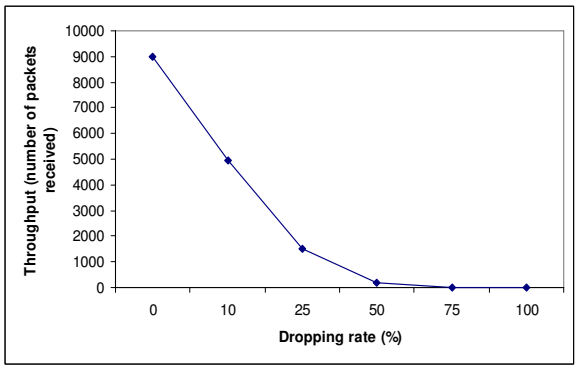

Figure 6: The Effect of Packet Dropping Attack on Throughput in wireless ad hoc network

4.3.3 Comparison with Existing Approaches for True Positive Detection Rate

Having seen true detection positive rate results for our SMDF, we are now comparing it with other mechanisms. Figure 7 depicts a comparison between our proposed SMDF and the Watchdog [5] and Random Two Hop ACK [9]. We have used the same simulation parameters mentioned before and run the simulations using each protocol separately (i.e. WD then Random Two Hop ACK). In addition, as in our SMDF simulation case we have set the POWER COTROL parameter to YES in order to see how the other two protocol perfume. The result shows that our proposed SMDF outperformed Watchdog, which suffers from a sharp fluctuation between (98% - 100%), whereas SMDF remains constant at 100%. On the other hand, SMDF has as same true detection rated as the random Two Hop ACK.

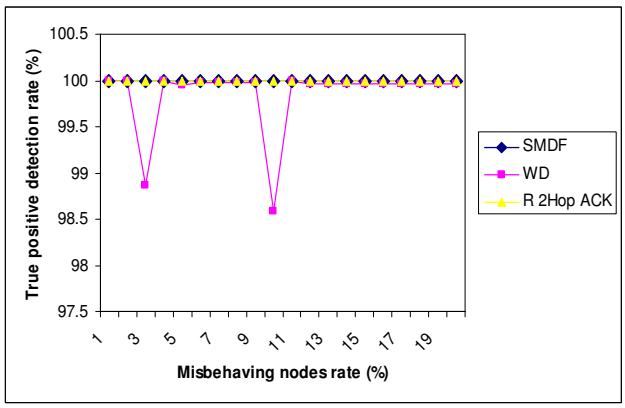

Figure 7: Comparison of True positive detection vs. Misbehaving rate

4.3.4 Comparison with Existing Approaches for False Positive Detection Rate

As we have done in the true detection positive rate comparison, we have compared our SMDF False Positive Detection Rate result with the Watchdog and the Random Two Hop ACK. Figure 8 shows clearly the considerable advantages of SMDF over both the Watchdog and the Random Two Hop ACK in keeping the false detection rate steady at 0% level. As the highest false detection rate was produced by the Watchdog which was between (35% - 75%), the Random Two Hop ACK performed better than the Watchdog as it fluctuates around 20%.

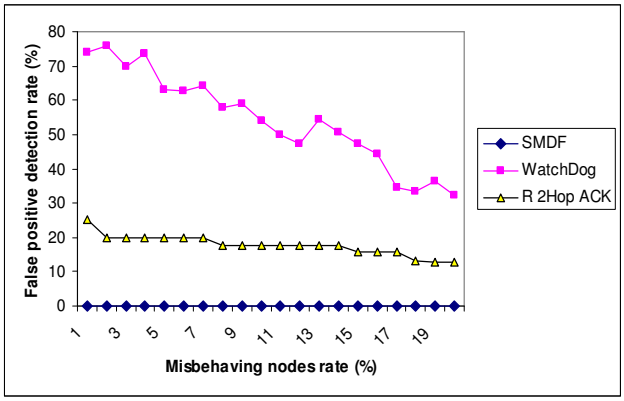

Figure 8: Comparison of False positive detection vs. Misbehaving rate

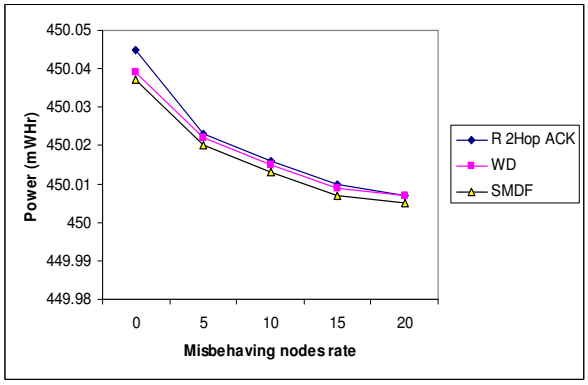

Figure 9: Comparison of power consumption of SMDF with existing approaches

4.3.5 Comparison of SMDF vs. Optimised SMDF with Sessions Aggregation and Two-Hop ACK

In this scenario we compare the overhead produced by SMDF in table 2 to the optimised SMDF and Two-Hop ACK. The optimised SMDF uses sessions aggregation approach presented earlier. 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. The results (reported in table 2) show a significant reduction in the amount of communication overhead produced by the aggregated SMDF in comparison with that in non-aggregated SMDF. Both versions clearly outperform Random Two-Hop ACK.

	Overhead Amount (Byte Per Session)				
Sessions	**1**	**2**	**3**	**4**	**5**
SMDF	724	1088	1512	2012	2028
Optimised SMDF	724	1022	1096	1436	1450
R 2 Hop ACK	11260	13300	20520	42800	43860

Table 2: Overhead Comparison between SMDF and R 2Hop ACK

4.3.6 Comparison with Existing Approaches for Power Consumption

Having seen the SMDF power consumption results in previous section, we now compare them with the Watchdog and Random Two Hop ACK results. The comparison results in figure 9 shows that our SMDF clearly outperforms both the Watchdog and Random Two Hop ACK in saving energy with less power consumption. There is a very small difference between the Watchdog and Random Two Hop ACK, with slight advantages to Watchdog. This could be due the huge amount of overhead that the Random Two Hop ACK generates.

4.3.7 Comparison with Existing Approaches for Scalability

Our proposed protocol SMDF has already been evaluated using 100 nodes, which is higher than 50 nodes average used in many other existing mechanisms evaluated using simulation. Since the scalability property is one of the desired characteristics especially in wireless sensor network, we have increased the number of nodes to 500 and the terrain to $3500x3000\,m^2$ and measure the true/false positives. The main difference between small and large networks is the average path lengths (e.g. 3-4 hops in small network vs. 8-13 hops in large network). We have increased the network sessions from 5 to 50 sessions to reflect the increase in the number of nodes. It can be seen from figure 10 that SMDF is scalable and still has the highest true positive detection rate at 100%. Figure 11 shows that SMDF has the lowest ever false positive detection rate at 0% compared with the WD and Random 2 Hop ACK.

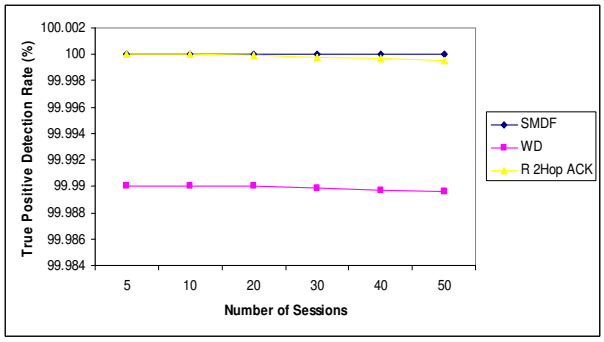

Figure 10: Comparison of True Positive Detection Rate for Scalability

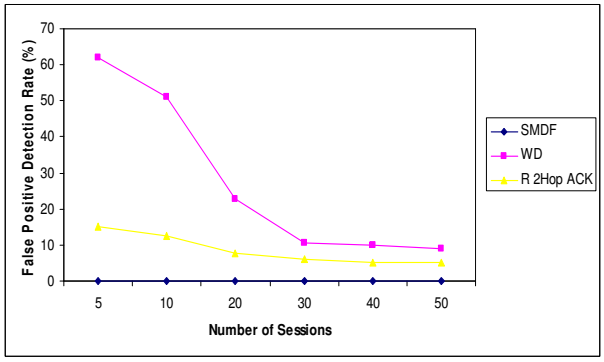

Figure 11: Comparison of False Positive Detection Rate for Scalability

5. Comparison with Related Work

As mentioned before, the main objective of our misbehaviour detection framework is to provide a set of components and mechanisms that can detect and eliminate misbehaviour at low energy and communication overhead cost but with high accuracy. The problem of node misbehaviour in wireless ad hoc network has been treated by many research groups, and many mechanisms have been proposed. Our framework shares some similarities with prior work carried out in other projects. In this section we compare our framework with these works.

The first and most famous mechanism in misbehaviour detection in wireless ad hoc network is the Watchdog [5]. We have compared our SMDF results with the Watchdog results on five of our six different metrics using simulation, and found that SMDF outperform the Watchdog in four of these metrics. The five metrics are true positive detection rate, false positive detection rate, power consumption rate and scalability. Where as the Watchdog has lower overhead than SMDF. Nonetheless, all of the other Watchdog drawbacks including partial dropping do not exist in our SMDF, which is the gain we obtain from our extra but low overhead. There are many other detection mechanisms especially the reputation mechanisms such as [7 18, 6, 17, 1] using the watchdog as their main monitoring component. Consequently, they inherit all the drawbacks that the watchdog suffers, even though their other system components are efficient. This gives our SMDF clear advantages over all of the mechanisms that adopting the watchdog concept in their detection system.

We have also compared our framework with the other types of mechanism do not use the watchdog as their monitoring component. The most recent solution of these is the Random Two-Hop ACK [9]. Our comparison through simulation showed us that our SMDF outperforms the Random Two-Hop ACK in four of our six simulation metrics. These matrices are communication overhead, false positive detection Rate, power consumption rate and scalability. Although, we have similar true positive detection rate as the Random Two-Hop ACK. However, through our simulation comparison we noticed that the Random Two-Hop ACK can often detect the full dropping case more than the partial one.

Our framework evaluation and comparison with other existing mechanisms shows it performs better and has novel aspects that do not exist in other mechanisms.

6. Discussion

Whilst the SMDF solves an interesting problem with some novel aspects there remains several shortcomings. This section outlines these problems.

Waiting until the end of all sessions to check node misbehaviour reduces the communication overhead considerably as we have seen in the simulation results. However, it will increase the delay before detection. It is a trade-off issue as reducing the cost is more valuable and important than increasing the waiting time. In some application such as video streaming, it is important to detect misbehaviour immediately as it occurred and not wait until the end of all sessions involved in the network. This shortcoming can be reduced by fixing a timeout to the sessions waiting instead of waiting for the end independently of the period, if not all the session terminate upon the timeout only the closed sessions will be aggregated and another timeout will be set for the remaining ones.

The Session-based Misbehaviour Detection Framework (SMDF) we proposed in this work assumes that nodes only drop data packets and not control packets. If the nodes drop the control packets, the SMDF cannot detect them, as it only deals with data packets dropping. The increased amount of control packets is not preferable due to the overhead they generate. However, moderate number of control packets is important, and as such, dropping them will affect the performance of the network. Selfish nodes drop both data and control packets whereas malicious node targeting only data packets. This make our solution directly applicable in the context of malicious nodes but needs to be completed with some mechanism for control packets monitoring to comprehensively deal with selfish nodes.

We intentionally examine the situation when wireless ad hoc network is stationary, as this was the main target investigation of this research. Our solution is suitable for most applications of wireless sensor networks. However, it is possible to have mobile sensor nodes. Dealing with mobility represents an open perspective to our work. In our SMDF Decision component we have used a fixed threshold of tolerance in which node will be judged as misbehaving after exceeding it. This is fine for a stationary wireless ad hoc network and static WSN scenarios. However, it is more efficient to use variable threshold which can change according to the network topology and scenarios. The threshold can be also estimated empirically for each network by first making simulations with no misbehaviour and calculate the threshold at each node for different scenarios that estimate the network. Then, retrieving the maximum value in all scenarios from the decision point and then consider it as a final threshold.

In some application such as battlefield or rescue operations there is no need to run all of the three components in every node. For example isolating and punishing such nodes would not be beneficial. In this case there is no need for the isolation component and it need to be switch OFF. Therefore, deciding when to turn components ON/OFF is another challenge that needs to be resolve. One suggestion is to add a separate component with *intelligent* decision capability to SMDF in order to deal with such situation. More efficient suggestion would be to add to each component of SMDF this intelligent capability to decide itself when and where to function according to the network and nodes status.

For the purpose of evaluation and comparison with other approaches we have used simulations techniques. Performance evaluation through simulations is helpful but will not reflect the reality 100%. It will thus be fascinating to see the actual performance of our complete SMDF

framework by integrating every component that it consisted of. By doing that we could measure new parameters that will add more understanding of the reality and that cannot be performed clearly through simulations. Once the above tasks have been completed successfully, it would be interesting to implement the complete model in an experimental test-bed to see its practical feasibility. This task appears feasible in the near future as the price of advanced sensors and handheld devices are already decreasing gradually.

7. Conclusion

The main direction of our work has been to look for an effective approach that can satisfy our initial requirements. The result is a new low cost framework entitled Session-based Misbehaviour Detection Framework (SMDF). It consists of three components, the detection component, the decision component and the isolation component. The most important contribution is the detection component that contains our novel Session-based Misbehaviour Detection Protocol SMDP to detect selfish or malicious nodes that drop packets partially or completely to launch either black-hole or data dropping attacks. For the decision component we have integrated an existing Bayesian approach to decide whether the node deliberately misbehaved or not [12]. For the Isolation component, we have used an existing approach [12] and used an Observation-Based Protocol to isolate misbehaving nodes. It uses neighbouring observation experience to isolate misbehaved nodes.

We analysed and evaluated the proposed schemes by simulation techniques. Our evaluation was focused on six important parameters, namely packet delivery ratio, overheard, true positive detection rate, false positive detection rate, power consumption rate and scalability. By comparing our results to those of other mechanisms available on literature, we showed that our solution has low cost in terms of communication overhead than other approaches. We showed also that our framework has the lowest false positive detection rate amongst other approaches, and that it has highest value of true positive detection rate compared with other approaches. Our evaluation also showed that our solution has lower energy consumption rate compared with other existing approaches. The experiments showed also that our framework is scalable and can work with higher number of nodes, especially in wireless sensor networks. It is important to emphasise that though the proposed framework was developed for stationary wireless ad hoc network and static wireless sensors network, the ideas by this framework are still extendable for other mobile wireless networks.

To achieve the grand vision of pervasive computing where applications are enhanced through tools such as wireless sensors and integrated using mobile ad hoc networks many problems need to be solved. However, remarkable progress has been made in the last decade and we believe our SMDF contribution, addressing fairness within wireless ad hoc network, will help make a step toward this future.

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] F. 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] L. Buttyan, 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] 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. Portoroz, Slovenia: Kluwer Academic.

[7] S. Buchegger, 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.

[12] D. Djenouri and N. Badache, "Struggling Against Selfishness and Black Hole Attacks in MANETs" Journal of Wirless Communications and Mobile Computing (WCMC), Wiley & sons Publisher.

[13] T. Fahad, D. Djenouri, and R. Askwith *"On Detecting Selfish Packet Droppers in MANET: A ovel Low Cost Approach"*, The Third IEEE International Symposium on Information Assurance

and Security IAS'07, pp 56-61, Manchester, UK, August 2007.

[14] T.Fahad, D.Djenouri, R.Askwith, M.Merabti *"A New Low Cost Sessions-Based Misbehaviour Detection Protocol (SMDP) for MANET"*, 21st IEEE International Conference on Advanced Information Networking and Applications Workshops Proceedings (AINAW'07), Niagara-falls, pp. 882-887, Ontario, Canada, May 2007.

[15]T.Fahad. "Session-Based Misbehaviour Detection Framework for Mobile Ad hoc Network ", Ph.D. thesis, School of Computing and Mathematical Sciences, Liverpool John Moores University, 2008.

[16] M. Conti, E. Gregori, G. Maselli. Improving the performability of data transfer in mobile ad hoc networks. The 2nd IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON 2005), Santa Clara, CA, Sept 26–29, 2005.

[17] H. Miranda, L. Rodrigues. *Friends and foes: preventing selfishness in open mobile ad hoc networks*. The 23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003), Providence, RI, May 19–22, 2003; 440–445.

[18] Q. He, D. Wu, and P. Khosla. "*SORI: A Secure and Objective Reputation-Based Incentive Scheme for Ad Hoc Networks", in IEEE Wireless Communications and Networking Conference WCNC 2004,* Atlanta, GA, 2004.

[19] S. Buchegger, J.-Y Le-Boudec. A robust reputation system for p2p and mobile ad-hoc net- works. 2nd Workshop on the Economics of Peer-to-Peer Systems, Barkeley, CA, June 4–5, 2004.

[20] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for the parallel simulation of large-scale wireless networks," in *The 12th Workshop on Parallel and distributed Simulation. PADS'98*, Banff, Alberta, Canada, May 1998, pp. 154–161.

[21] J. Broch, D.B. Johnson, and D.A. Maltz, "*The dynamic source routing protocol for mobile ad hoc networks. Internet draft"*, 2004, IETF, URL: http://www1.ietf.org/mail-archive/web/ietf-announce/current/msg02559.html.

[22] W. Du, J. Deng, Y.S. Han, and P.K. Varshney. "*A Witness-based Approach for Data Fusion Assurance in Wireless Sensor Networks",* in *GLOBECOM '03,*2003.

[23] T. Fahad, R. Askwith. "*A Node Misbehaviour Detection Mechanism for Mobile Ad-hoc Networks",* in *The Seventh Annual Postgraduate Network Symposium (PGNet 2006),* Liverpool, UK, 2006.

[24]T. Fahad, R. Askwith, M. Howarth, G. Pavlou. "*Detecting Selfish Nodes in Wireless Mobile Ad-Hoc Networks",* in *1st Conference on Advances in Computer Security and Forensics,* Liverpool, UK, 2006.

Author Biographies

Tarag Fahad recently completed his PhD at Liverpool John Moores University, UK. He received two master degrees in Information Technology Systems from Heriot-Watt University and in Mobile and Distributed Networks from Leeds Metropolitan University in the UK. His research interests include Security in mobile ad hoc networks, wireless sensors networks and intrusion detection. He is involved in projects on network security and network management and his PhD subject was in the area of nodes misbehaviour detection in Mobile Ad Hoc Networks.

Robert Askwith is a Senior Lecturer in Computer Systems with the School of Computing and Mathematical Sciences at Liverpool John Moores University. In 1996 he received the BSc in Software Engineering from the school before joining the Distributed Multimedia Systems research group to study for a PhD in Computer Network Security, which he was awarded in 2000. His research interests include Computer and Network Security, User Privacy in Networked Systems, Personal Networking including Ad Hoc Networks, Networked Appliances and Pervasive Computing.

Djamel Djenouri obtained his PhD in 2007 at USTHB and currently working as a Research Associate at the CERIST centre of research in Algiers. He also obtained his Engineering Degree in Computer Science and his Master's Degree in Computer Science from the University of Science and Technology USTHB (Algiers), respectively in 2001 and 2003. He works mainly on ad hoc networking, especially on the following topics: Security, Power Management, Routing Protocols, MAC Protocols.

Madjid Merabti joined Liverpool John Moores University as a Senior Lecturer in Computing Science. He is now Professor of Networked Systems. His current research interests include architectures, services and protocols for distributed multimedia systems, including multimedia content and extraction, mobile networks, security, and e-commerce technology support. He is involved in a number of projects in these areas where he leads the Distributed Multimedia Systems Group. Madjid Merabti is co-chair for the Security and Network Management track of Globecom 2004. He is also chair of the EPSRC funded UK PostGraduate Networking Symposium (PG Net).