Performance Analysis of the CONFIDANT Protocol Cooperation Of Nodes — Fairness In Dynamic Ad-hoc NeTworks

Sonja Buchegger
IBM Zurich Research Laboratory
Säumerstrasse 4, CH-8803 Rüschlikon
sob@zurich.ibm.com

Jean-Yves Le Boudec
EPFL-DSC
CH-1015 Lausanne, Switzerland
jean-yves.leboudec@epfl.ch

Abstract: Mobile ad-hoc networking works properly only if the participating nodes cooperate in routing and forwarding. However, it may be advantageous for individual nodes not to cooperate. We propose a protocol, called CONFIDANT, for making misbehavior unattractive; it is based on selective altruism and utilitarianism. It aims at detecting and isolating misbehaving nodes, thus making it unattractive to deny cooperation. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. The detailed implementation of CONFIDANT in this paper assumes that the network layer is based on the Dynamic Source Routing (DSR) protocol. We present a performance analysis of DSR fortified by CONFIDANT and compare it to regular defenseless DSR. It shows that a network with CONFIDANT and up to 60% of misbehaving nodes behaves almost as well as a benign network, in sharp contrast to a defenseless network. All simulations have been implemented and performed in GloMoSim.

1 Introduction

The CONFIDANT protocol works as an extension to a reactive source-routing protocol for mobile ad-hoc networks. For the simulation implementation, we have chosen Dynamic Source Routing (DSR) as the base protocol. In the following subsections we briefly describe what we need to know of DSR, describe the attacks we support, and specify how we want to thwart them.

1.1 Background: the DSR Protocol

Dynamic Source Routing is a protocol developed for routing in mobile ad-hoc networks and was proposed for MANET by Broch, Johnson and Maltz [JM99]. In a nutshell, it works as follows: Nodes send out a ROUTE REQUEST message, all nodes that receive this message forward it to their neighbors and put themselves into the source route unless they have received the same request before. If a receiving node is the destination, or has a route to the destination, it does not forward the request, but sends a REPLY message containing the full source route. It may send that reply along the source router in reverse order or issue a ROUTE REQUEST including the route to get back to the source, if the former is not possible due to asymmetric links. ROUTE REPLY messages can be triggered by ROUTE REQUEST messages or gratuitous. After receiving one or several routes, the source picks the best (by default the shortest), stores it, and sends messages along that path. In general, the better the route metrics (number of hops, delay, bandwidth or other criteria) and the sooner the REPLY arrived at the source (indication of a short path - the nodes are required to wait a time corresponding to the length of the route they can advertise before sending it in order to avoid a storm of replies), the higher preference is given to the route and the longer it will stay in the cache.

In case of a link failure, the node that cannot forward the packet to the next node sends an error message towards the source. Routes that contain a failed link, can be 'salvaged' by taking an alternate partial route that does not contain the bad link.

1.2 Attacks against routing

The lack of infrastructure and organizational environment of mobile ad-hoc networks offer special opportunities to attackers. Without proper security, it is possible to gain various advantages by malicious behavior: better service than cooperating nodes, monetary benefits by exploiting incentive measures or trading confidential information; saving power by selfish behavior, preventing someone else from getting proper service, extracting data to get confidential information, and so on.

Several routing and forwarding attacks on DSR have been described in [BB02]. We aim at protection against the following types of misbehavior.

- o No forwarding (of control messages or data).
- o Unusual traffic attraction (advertises many excellent routes or advertises routes very rapidly, so they are deemed good routes).
- o Route salvaging, i.e., rerouting to avoid a broken link, although no error has been observed.
- o Lack of error messages, although an error has been observed.
- o Unusually frequent route updates.
- o Silent route change (tampering with the message header of either control or data packets).

1.3 Thwarting Attacks

A method for thwarting attacks is prevention. According to Schneier [Sch00], a prevention-only strategy only works if the prevention mechanisms are perfect; otherwise, someone will find out how to get around them. Most of the attacks and vulnerabilities have been the result of bypassing prevention mechanisms. Given this reality, detection and response are essential (see also Section 2 for a discussion of strong prevention mechanisms such as [BH00]). In this paper, we propose a method based on detection of misbehavior, followed by a reaction. We would like to achieve that only good behavior pay off in terms of service and reasonable power consumption.

Detection has to trigger a reaction of other nodes that results in a disadvantage for the malicious node. We propose that packets of malicious nodes should, upon detection of the node's malice, not be forwarded by normally behaving nodes. If, however, a node was wrongly accused of being malicious or turns out to be a repenting criminal that is no longer malicious and has behaved normally for a certain amount of time, some sort of 're-socialization' and re-integration into the network communications should be possible.

With the scheme we present in this paper, it is disadvantageous for nodes to behave maliciously; it is inspired by an example in ecology explained in Section 3.1.

1.4 Organization of the Paper

The remainder of this paper is organized as follows: related work is discussed in Section 2, followed by a description of the CONFIDANT protocol in Section 3. Section 4 gives a first performance evaluation of CONFIDANT, in the case where attacks are "no forwarding". Future work is outlined in Section 5 and conclusions are drawn in Section 6.

2 Related Work

Anderson and Stajano [AS99] authenticate users by 'imprinting' according to the analogy of ducklings acknowledging the first moving subject they see as their mother, but enabling the devices to be imprinted several times. The imprinting is realized by accepting a symmetric encryption key from the first device that sends such a key. They do not address routing or forwarding, however, user authentication and authorization are an important prerequisite for trust in the network layer also in mobile ad-hoc networks.

Zhou and Haas [ZH99] employ asynchronous threshold security and share refreshing for distributed certification authorities for key management in mobile ad-hoc networks. They take advantage of inherent redundancies in mobile ad-hoc networks given by multiple routes to enable diversity coding, allowing for byzantine failures given by several corrupted nodes or collusions. The approach is a potentially strong prevention mechanism, however, to the best of our knowledge, the impact on the network and security performance have not been published.

Smith, Murthy and Garcia-Luna-Aceves [SMGLA97] examined the routing security of distance vector protocols in general and developed countermeasures for vulnerabilities by protecting both routing messages and routing updates. They propose sequence numbers and digital signatures for both routing messages and updates and including predecessor information in routing updates. Digital signatures have also been suggested for the OSPF routing protocol by Murphy and Badger [MB96]. It remains to be investigated whether and how digital signatures can be employed in mobile ad-hoc networks. The CONFIDANT protocol also addresses routing misbehavior but in addition gives strong incentives for correct forwarding.

Buttyán and Hubaux proposed incentives to cooperate by means of so-called nuglets [BH00] that serve as a per-hop payment in every packet or counters [BH01] in a secure module in each node to encourage forwarding. One of their findings is that increased cooperation is beneficial not only for the entire network but also for individual nodes, which conforms to our results. The main differences to the CONFIDANT protocol are that nuglets or counters are limited to a one-to-one interaction, whereas in the CONFIDANT protocol misbehavior results in bad reputation propagating to more than one node and that the CONFIDANT protocol addresses additional issues in the network layer such as traffic diversion. The question of a tamper-proof security module remains controversial [PPSW97], but might prove to be inevitable. As opposed to nuglets and counters, the CONFIDANT protocol does not need tamper-proof hardware for itself, since a malicious node does neither know the entries of its reputation in other nodes nor does it have access to all other nodes for potential modification. The secure module might still be necessary for complementary protection such as authentication.

Marti, Giuli, Lai and Baker [MGLB00] observed increased throughput in mobile ad-hoc networks by complementing DSR with a watchdog (for detection of malicious behavior) and a ‘pathrater’ (for trust management and routing policy, every path used is rated), which enable nodes to avoid malicious nodes in their routes. Their approach does not punish malicious nodes that do not cooperate, but rather relieves them of the burden of forwarding for others, whereas their messages are forwarded without complaint. This way, the malicious nodes are rewarded, and reinforced in their behavior. In contrast, with our protocol we would like to achieve the opposite.

The SAR (Security-aware Ad-hoc Routing) protocol by Yi, Naldburg and Kravets [YNK01] modifies AODV to include security metrics for path computation and selection. They define trust levels according to organizational hierarchies with a shared key for each level, so that nodes can state their security requirements when requesting a route and only nodes that meet these requirements (trust level, metrics) participate in the routing. Questions not addressed by this protocol yet include the mechanism for key distribution, knowledge of the keys of the other nodes, what happens when a node leaves the group with the shared trust level and how trust hierarchies are defined in the first place, especially in civilian applications. SAR relies on tamper-proof hardware.

3 When Nodes Bear Grudges: The CONFIDANT Protocol

We now describe the protocol. First we give the rationale and explain how it finds its root in an ecological analogy. Then we describe the components of CONFIDANT, assumed to be present in every node. Lastly, we describe the protocol with free text and a finite state machine.

3.1 The Selfish Gene: from birds to network nodes

As explained by Richard Dawkins in 'The Selfish Gene' [Daw76], reciprocal altruism is beneficial for every ecological system when favors are granted simultaneously, so there is an intrinsic motivation for cooperation because of instant gratification. The benefit of behaving well is not so obvious in the case where there is a delay between granting a favor and the repayment. This is the case when, in mobile ad-hoc networks, nodes forward on behalf of each other. An ecological example used by Dawkins [Daw76] explains the survival chances (and thus gene selection) of birds grooming parasites off each other's head, which they cannot clean themselves.

Dawkins divides birds into two types: 'suckers' that always help and 'cheats' that have other birds groom parasites off their head but fail to return the favor. In this system, clearly the cheats have an advantage over the suckers, but both are driven to extinction over time. Dawkins then introduces a third kind of bird, the 'grudger' that starts out being helpful to every bird, but bears a grudge against those birds that do not return the favor and subsequently no longer grooms their heads.

According to Dawkins, simulation has shown that when starting with a majority population of cheats and marginal groups of both suckers and grudgers, the grudgers win over time. Winning is defined as having the greatest benefit, assuming a cost for grooming another bird's head and a profit for having one's head groomed, with a loss leading to extinction and profit leading to multiplication of the species. The rationale is as follows: the suckers do favors more than they get because of the large number of cheats, so the number of suckers decreases, whereas the number of cheats increases. The grudgers also suffer from some loss, but less than the suckers. Once the suckers are extinct, the grudgers grow rapidly at the expense of the cheats, because they do not help a cheat twice and cheats are also not helped by other cheats. After a while, the number of cheats decreases more slowly, because the probability of a first-help by a grudger increases with a higher population of grudgers. Over all, the population of the grudgers grows, whereas the other species become extinct.

Defining suitable cost and profit to routing and forwarding favors and keeping a history of experiences with non cooperating nodes achieve the same as the grudger species, i.e., driving the cheats out of business. In a very large ad-hoc network, convergence can be very slow, and keeping a history of all bad experiences with other nodes equals large storage requirements and long lists to go through. Therefore, we propose the following ideas, which are incorporated in the CONFIDANT protocol explained in the next section, to speed up the triumph of grudger nodes:

- learn from observed behavior: employ 'neighborhood watch' to be warned by observing what happens to other nodes in the neighborhood, before having to make a bad experience oneself,
- learn from reported behavior: share information of experienced malicious behavior with friends and learn from them.

3.2 CONFIDANT Components

CONFIDANT consists of the following components, as shown in Figure 1: **The Monitor, the Reputation System, the Path Manager, and the Trust Manager.** The components are present in every node.

3.2.1 The Monitor (Neighborhood Watch)

In a wireless networking environment, the nodes most likely to detect non-compliant 'criminal' behavior are the nodes in the vicinity of the criminal and in some cases the source and the destination, if they detect unusual behavior or do not get proper responses. The latter is not always the case, for instance in the case of replay. One approach to protocol enforcement and detection of damaging behavior (intrusion, misuse of cooperation incentives, denial of service, etc.) suggested here is the equivalent of a 'neighborhood watch', where nodes locally look for deviating nodes.

The nodes of the neighborhood watch can detect deviations by the next node on the source route by either listening to the transmission of the next node or by observing route protocol behavior. By keeping a copy of a packet while listening to the transmission of the next node, any content change can also be detected. All misbehaviors listed in Section 1.2 can be indicated. However, in the GloMoSim simulations, used for this report, only "no forwarding" attacks are implemented.

As a component within each node, the monitor registers these deviations from normal behavior. As soon as a given bad behavior occurs, the reputation system is called.

3.2.2 The Trust Manager

In an ad-hoc environment, trust management has to be distributed and adaptive [BFL96]. This component deals with incoming and outgoing ALARM messages.

ALARM messages are sent by the trust manager of a node to warn others of malicious nodes. Outgoing ALARMS are generated by the node itself after having experienced, observed, or received a report of malicious behavior. The recipients of these ALARM messages are so-called *friends*, which are administered in a friends list. How to win friends in a mobile ad-hoc network dynamically is still on our research agenda, however, for the moment we consider friends to be configured in a way similar to device imprinting as described by Anderson and Stajano [AS99] on a user-to-user basis.

Incoming ALARMs originate from either outside friends or other nodes, so the source of an ALARM has to be checked for trustworthiness before triggering a reaction, thus there is a filtering of incoming ALARM messages according to the trust level of the reporting node. A mechanism similar to the trust management in Pretty Good Privacy (PGP) for key validation and certification is used here for mobile ad-hoc networks for trust management for routing and forwarding. In PGP [Zim93], several levels of trust can be expressed, e.g. 'unknown', 'none', 'marginal', and 'complete'. When PGP calculates the validity of a public key, it examines the trust level of all the attached certifying signatures. It computes a weighted score of validity. For example, two marginally trusted signatures might be deemed credible as one completely trusted signature. The weighting scheme is adjustable so that it can require a different number of marginally trusted signatures to judge a key as valid. We use the same principle but for the purpose of determining whether there is sufficient trusted evidence for the misbehavior of a node.

The trust manager consists of the following components.

- o An alarm table containing information about received alarms.
- o A trust table managing trust levels for nodes to determine the trustworthiness of an alarm.
- o A friends list containing all friends a node potentially sends alarms to.

For routing and forwarding, trust is important when making a decision about

- o providing or accepting routing information,
- o accepting a node as part of a route, and
- o taking part in a route originated by some other node.

3.2.3 The Reputation System (Node Rating)

Reputation systems are used in some online auctioning systems. They provide a means of obtaining a quality rating of participants of transactions by having both the buyer and the seller give each other feedback on how their activities were perceived and evaluated. For a detailed explanation of reputation systems see Resnick *et al.* [RZFK00].

To avoid a centralized rating, local rating lists and/or black lists are maintained at each node and potentially exchanged with friends. In the route request nodes can include that *black sheep* be avoided for routing, which also alarms nodes along the way. Nodes can look up senders in the black list containing the nodes with bad rating before forwarding anything for them. The problem of how to distinguish alleged from proven malicious nodes, i.e. how to avoid false accusations, can be lessened by timeout and subsequent recovery or revocation lists of nodes that have behaved well for a specified period of time. Another problem is scalability and how to avoid blown-up lists, which can also be addressed by timeouts.

The reputation system in this protocol manages a table consisting of entries for nodes and their rating. The rating is changed only when there is sufficient evidence of malicious behavior that is significant for a node and that has occurred a number of times exceeding a threshold to rule out coincidences. The rating is then changed according to a rate function that assigns different weights to the type of behavior detection, namely the greatest weight for own experience, a smaller weight for observations in the neighborhood and an even smaller weight to reported experience. The rationale for this weighting scheme is that nodes trust their own experiences and observations more than those of other nodes.

Once the weight has been determined, the entry of the node that misbehaved is changed accordingly. If the rating of a node in the table has deteriorated so much as to fall out of a tolerable range, the path manager is called for action. Bearing in mind that malicious behavior will ideally be the exception rather than the norm, the reputation system is built on negative experience rather than positive impressions. The issues of positive change and timeout are still to be addressed in detail.

3.2.4 The Path Manager

The path manager performs the following functions:

- Path re-ranking according to security metric, e.g. reputation of the nodes in the path.
- Deletion of paths containing malicious nodes.
- Action on receiving a request for a route from a malicious node (e.g. ignore, do not send any reply).
- Action on receiving request for a route containing a malicious node in the source route (e.g. ignore, alert the source).

3.3 Protocol Description

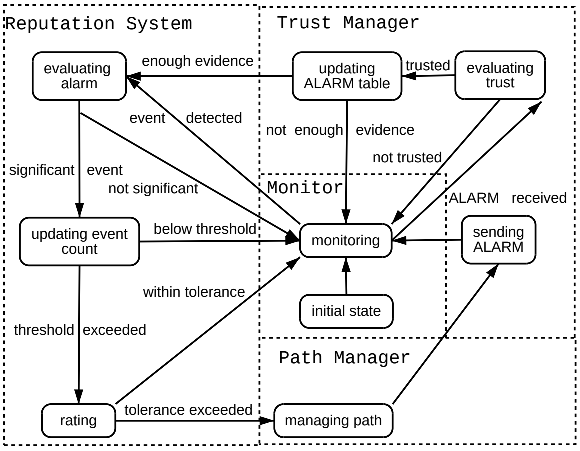

Figure 1: Trust architecture and finite state machine within each node.

As shown in Figure 1, each node monitors the behavior of its next hop-neighbors. If a suspicious event is detected, the information is given to the reputation system. If the event is significant for the node, it is checked whether the event has occurred more often than a predefined threshold that is high enough to distinguish deliberate malicious behavior from simple coincidences such as collisions. What constitutes the significance rating can be defined for different types of nodes according to their security requirements. If that occurrence threshold is exceeded, the reputation system updates the rating of the node that caused that event. If the rating turns out to be intolerable, the information is relayed to the path manager, which proceeds to delete all routes containing the intolerable node from the path cache. The node continues to monitor the neighborhood, and an ALARM message is sent as described in the following:

In order to convey warning information, an ALARM message is sent by the trust manager component. This message contains the type of protocol violation, the number of occurrences observed, whether the message was self-originated by the sender, the address of the reporting node, the address of the observed node, and the destination address (either the source of the route or the address of a friend that might be interested). In the present simulation implementation, the ALARM is sent to the source of the concerned route.

When the monitor component of a node receives such an ALARM message, it passes it on to the trust manager, where the source of the message is evaluated. If the source is at least partially trusted, the table containing the ALARMs is updated. If there is sufficient evidence that the node reported in the ALARM is malicious, the information is sent to the reputation system where it is again evaluated for significance, number of occurrences and accumulated reputation of the node as explained in Section 3.2.3. Sufficient evidence means that either the source of the ALARM is fully trusted or that several partially trusted nodes have reported the same and their respective assigned trust adds up to a value of one entirely trusted node or more.

4 Performance Analysis

4.1 Goal

The objective of this performance analysis is to determine the impact of the CONFIDANT routing protocol extensions on metrics as described in Section 4.2 in an ad-hoc network where a part of the population acts maliciously. The regular DSR protocol is used as a reference. For all these metrics, we want to investigate the scalability in terms of number of nodes, fraction of malicious nodes, and mobility.

For future work, our goal is also to learn how protocol parameters such as thresholds should be set. Given these parameters we will determine how many friends per benign node are needed to tolerate a given percentage of malicious nodes.

4.2 Metrics

The following metrics are considered.

Throughput, Goodput, Dropped Packets. One metric is the resulting total *goodput* G of a network with n nodes, i.e. the data forwarded to the correct destination. We express this as:

$$G = \frac{\sum_{i=1}^{n} Originated}{\sum_{i=1}^{n} Received} \tag{1}$$

As opposed to the throughput, packet loss and retransmissions are taken into account. The goodput is directly influenced by packet loss. Packet loss can occur due to general network conditions causing link errors or unreachable nodes, but packets can also be lost because an intermediate node intentionally drops them. The latter is the only form of packet loss directly attributable to malicious behavior. We therefore use the number of intentionally dropped packets as a metric, both in absolute numbers and relative to the number of packets originated.

Overhead. Since the cost of internal computation in terms of energy consumption is negligible compared to the cost of a transmission, we look at the overhead caused by extra messages and define the total overhead O in a network of n nodes as follows. We consider each transmission *tx* of a control message, not only origination or reception.

$$O = \frac{\sum_{i=1}^{n} ALARM_{tx}}{\sum_{i=1}^{n} RREQ_{tx} + RREP_{tx} + ERROR_{tx}} \tag{2}$$

We use this ratio to determine how much extra overhead the CONFIDANT extensions cause relative to the regular routing overhead. The overhead that can be clearly attributed to the CONFIDANT extensions are the ALARM messages transmitted. ROUTE-REQUEST, ROUTE-REPLY and ERROR messages in the case of DSR or, to be more general, any messages needed for rerouting depend on the underlying routing protocol. The CONFIDANT protocol points out the identity of misbehaving nodes and allows the routing protocol to reroute around them.

Profit. We try to determine whether cooperation pays off for a node. One metric that directly reflects a cost-benefit trade-off is the ratio of how many of the transmissions of a node are originated or received by the node itself versus how many are just forwarded as an intermediate node on behalf of other nodes. Thus we look at the ratio of originated to transmitted packets. Assuming a cost c_f of forwarding a packet (composed of power, CPU usage, memory usage) and a benefit b_r when receiving a packet as a destination or b_s when having an own packet received by the destination, we define the profit p of a node as

$$\begin{aligned} p = {} & b_r \sum Packets_{received} \\ & + b_s \sum Packets_{sent\_successfully} \\ & - c_f \sum Packets_{transmitted}, \end{aligned} \tag{3}$$

The total profit P for the network of n nodes is denoted by:

$$P = \sum_{i=1}^{n} p_i \tag{4}$$

4.3 Simulation Setup

For the performance analysis of the protocol extensions, the metrics are observed in various network scenarios given by different modifications of the DSR protocol. The first network we analyze is a regular well-behaved DSR network which is used as a reference.

We then introduce compromised nodes that do not cooperate. These malicious nodes do not forward messages for other nodes. The next kind of network we use for analysis is a network containing a certain fraction of malicious nodes but no defense mechanism, we call it ‘defenseless’.

Then we use a version of DSR that we enhanced with CONFIDANT extensions and refer to it as ‘fortified’. The first enhancement towards a fortified network is the reaction of a node on its own bad experience. If a node notices that its next-hop neighbor does not forward, it will avoid that node for future communications. The second enhancement is to include the case when the neighbor node fails to forward a packet for some other node and it is detected. The third enhancement is given by warnings (ALARM messages) sent to the source by friends that observe that a node is behaving maliciously. To take this one step further, nodes can use the information contained in ALARM messages that they overhear promiscuously, irrespective of whether they are the actual destination of the message, i.e., the source of the compromised route or a friend. In the simulation, every benign node is a friend of the source and informs the source when packets are maliciously dropped by the next hop. This represents an almost ideal case given the presence of malicious nodes, except that nodes in this implementation do not propagate ALARMs to friends other than the source. Future performance analysis will determine the number of friends actually needed to sufficiently limit the influence of malicious nodes.

Out of the variety of routing and forwarding attacks on DSR found in [BB02] , we concentrate on forwarding defection for this performance analysis, because it can be detected easily and its impact on network performance can be measured.

The simulation is implemented on GloMoSim [ZBG98], a simulator for mobile ad-hoc networks. Unless otherwise specified, the experiments were repeated ten times with varying random seed. The seed influences the placement and movement of the nodes. Whenever confidence intervals are shown in plots, the confidence level on these intervals is 95 %.

The fixed parameters for the simulation are listed in Table 1. The radio range, sending capacity and MAC have been chosen to represent an off-the-shelf device, the speed is uniformly distributed between 0 and 20 m/s to offer a range of users that are in a fixed location, walking, or driving a car; the chosen area approximately represents the center of a town. The simulation time is chosen to be long enough to potentially roam the whole area. The mobility model chosen is the *Random Waypoint Model*, in which nodes move to a random destination at a speed uniformly distributed between 0 m/s and

Parameter	Level
Area	1000 m × 1000 m
Speed	uniformly distributed between 0 and 20 m/s
Radio Range	250 m
Placement	uniform
Movement	random waypoint model
MAC	802.11
Sending capacity	2 Mbps
Application	CBR
Packet size	64 B
Simulation time	900 s

Table 1: Fixed Parameters

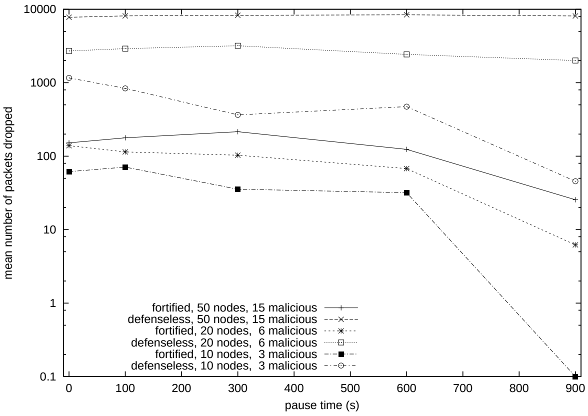

Figure 2: Mean number of packets dropped versus pause time.

a specified maximum speed. Once they reach this destination, they stay there for as long as specified in the *pause time* parameter. The reason for this movement model is to have a random movement with pauses with the aim to reflect realistic user behavior. The placement has been chosen to start with a good network connectivity over the whole area. Finally, CBR has been chosen for traffic (we refer to it as applications) to avoid protocol particularities of more complicated protocols such as TCP. The application is defined as follows. A client constantly sends to a server which in turn responds to the client. The client-server-pairs have been randomly generated for the simulation.

The factors varied are the total number of nodes in the network, the percentage of malicious nodes, the pause time and the number of applications. The rationale for the choice of these factors is given in Section 4.5.

4.4 Simulation Results

Figure 2 shows the mean number of packets dropped, varying the pause times and the network size, i.e. the number of nodes, but keeping the fraction of malicious nodes fixed at a third of the total population. At any time during the simulation 10 CBR-connections are active. In the defenseless network, the number of packets intentionally dropped is up to two orders of magnitude greater than in the network fortified by CONFIDANT. The results are fairly constant with respect to mobility, only decreasing slightly in the case of an almost static network at a pause time of 900 s. The fortified network is a little more sensitive to mobility. This can be explained by the increased probability of meeting a previously unknown malicious node when nodes move around more. For future work, in order to determine convergence of the protocol, it will be interesting to see the distribution of dropped packets over the time of the simulation to the point when the nodes have met many other nodes already and judged their reputation.

When looking at the number of packets dropped from a network-size perspective, it can be seen from Figure 3 that the difference in performance increases with the total number of nodes in the network. The fortified network keeps the number of dropped packets fairly constant irrespective of the network size, whereas the defenseless network deteriorates significantly with increasing total number of nodes.

In Figure 4, the confidence intervals are shown for the mean ratio of number of packets dropped to packets originated. The analyzed network consists of 50 nodes and the number of applications was increased to 30 in order to observe the behavior in a more heavily loaded network. DSR fortified with CONFIDANT extensions loses only a small fraction of packets (always less than 3%) because of

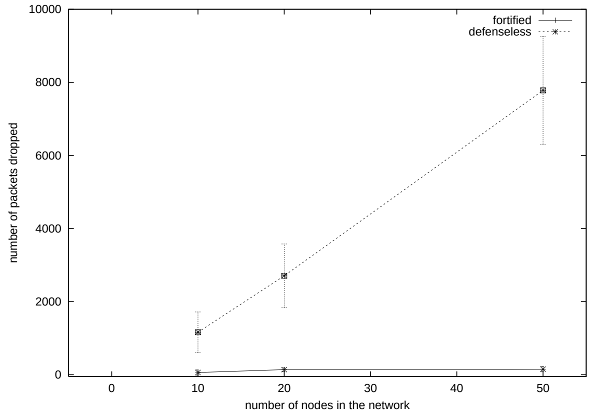

Figure 3: Mean number of packets dropped versus number of nodes, one third is malicious.

malicious nodes, whereas regular, defenseless DSR faces a loss of a significant number (around 70%) of the packets, all other parameters being equal. The defenseless network does not benefit from a more static network, as opposed to the fortified network.

Figure 5 shows how the CONFIDANT protocol copes with a varying percentage of malicious nodes in the total network population. The pause time is set to 0 to stress the CONFIDANT protocol with a very dynamic network, where it cannot use the advantage of improving with more stability which it showed in the previous figures. The number of applications is equally deliberately set as high as 30 for increased load. It can be seen that in a defenseless network, already a small percentage of malicious nodes can wreak havoc. There is not much difference in the number of intentionally dropped packets as the percentage of malicious nodes increases. This can be explained by the fact that it does not matter where on the path a packet is lost. The network fortified with CONFIDANT is more sensitive to the percentage of malicious nodes, however, it still keeps the number of deliberately dropped packets low even in a very hostile environment as given by more than half the population acting maliciously - given that there are enough nodes to provide harmless alternate partial paths around malicious nodes.

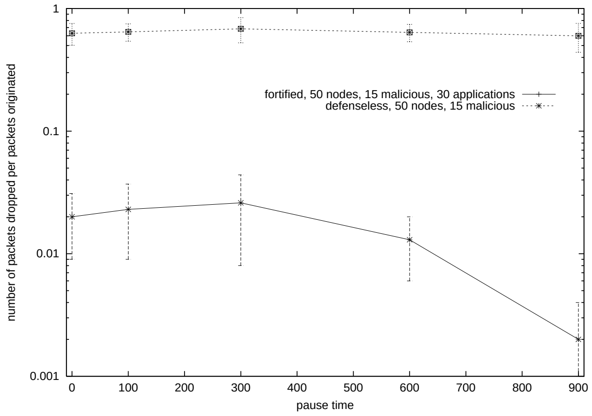

Figure 4: Number of packets dropped per number of packets originated by 30 applications, 20 simulation runs.

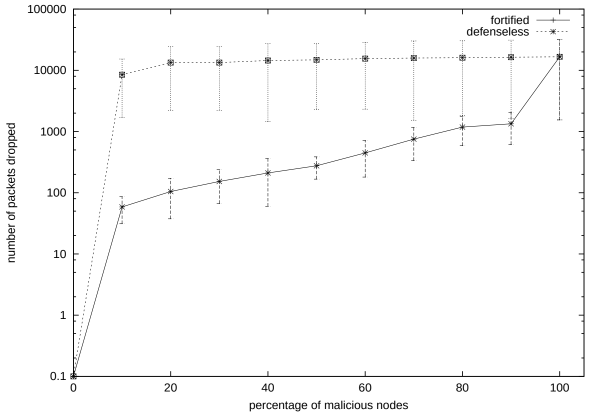

Figure 5: Number of packets dropped, 50 nodes, 30 applications, 0 pause time, varying percentage of malicious nodes

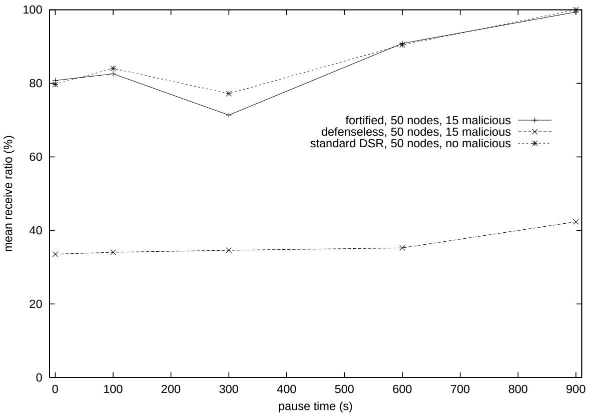

Figure 6: Goodput expressed as the ratio of received to sent packets, one third of 50 nodes is malicious, 20 simulation runs.

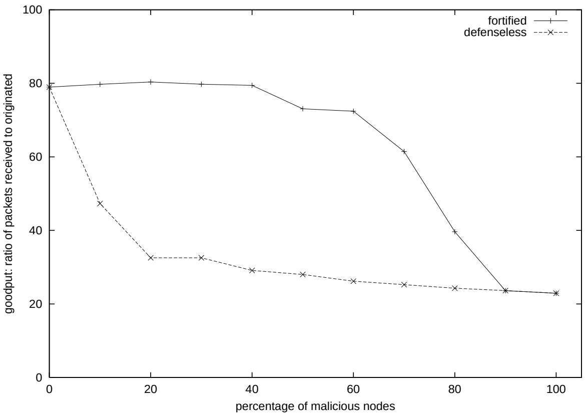

Figure 7: Goodput, 50 nodes, 30 applications, 0 pause time, varying percentage of malicious nodes.

In comparing the ratio of packets sent and received in Figure 6, the performance of the fortified network in which a third of the population behaving maliciously is very close to that of a regular benign DSR network without malicious nodes. The reason that the ratio is below 100 % even in a benign network is that losses are not only due to malicious nodes dropping packets but also to link errors or because nodes have moved away too quickly for the protocol to catch up.

The goodput versus the percentage of malicious nodes is depicted in Figure 7. The network is again highly mobile with a pause time of 0 s, which explains the goodput of only about 80% even for a network containing no malicious nodes. The fortified network keeps this performance up in the presence of up to 40% malicious nodes and deteriorates only slightly in the presence of up to 60% malicious nodes. With 90% or more malicious nodes finally, the fortified network cannot improve the performance anymore. The fact that even in a population of only malicious nodes there is still a goodput of about 20% can be explained by a portion of the communication happening between nodes that are within each others radio range.

Figure 8 shows the throuphput of clients and servers according to the CBR applications used. Clients send at a constant bit rate of 2 Mbits, the servers respond according to the packets they receive. The fortified version is not very close to the benign network, but it can also take advantage of longer pause times, i.e., a less mobile network, whereas the performance of the defenseless version remains unacceptable.

Figure 9 shows the ratio of ALARM messages in the total number of control messages transmitted. It is always lower than 3%, although factors chosen, namely 'number of nodes', 'number of applications' and 'fraction of malicious nodes', are at their maximum according to Table 2, thus presenting the least favorable case in these simulation boundaries. It is also an upper bound given the parameters and factors of this simulation in that the threshold for sending an ALARM after having detected a forwarding failure is set to 1, i.e., every maliciously dropped packet detected is reported by an ALARM message.

4.5 Estimation of Factor Relevance

In order to find out which factors actually have an effect on the performance metrics and to reduce the number of experiments, a $2^k r$ factorial design according to Jain [Jai91] was performed, with k (the number of factors) being set to 3 and 5, r (the number of repetitions of the experiment) set to 10, resulting in 8 and 32 experiments or 80 and 320 simulation runs, respectively. Table 2 shows the factors

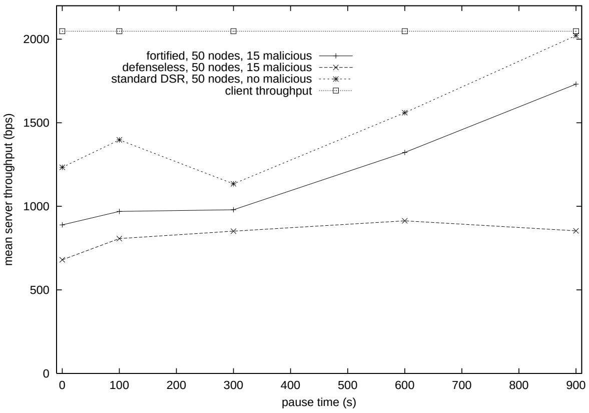

Figure 8: Mean client and server throughput in a network of 50 nodes with one third malicious, 20 simulation runs.

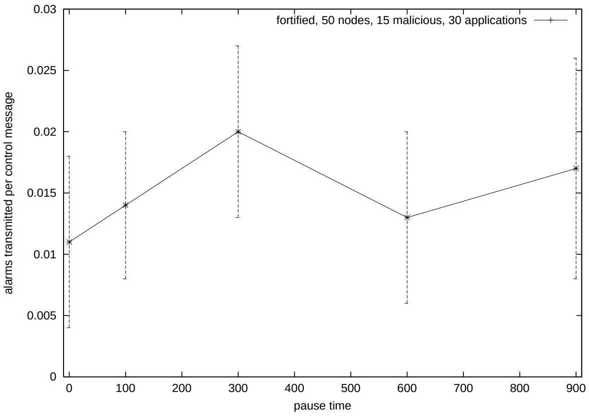

Figure 9: Mean overhead caused by the CONFIDANT protocol, 20 simulation runs

and the two extreme levels that were chosen for the experiments.

Factor	Level 1	Level 2
Number of nodes	10	50
Protocol	defenseless DSR	fortified CONFIDANT
Pause time	0 s	600 s
Percentage of malicious nodes	0.00%	33.33%
Number of applications	10	30

Table 2: Levels for factorial design

The choice for the number of nodes was made with the intention to show both a very small network that still allows for multiple paths and reasonable network connectivity given the area and a larger network to get insights on scalability. The pause times were chosen to reflect a very mobile network as well as a very moderately mobile one given that the duration of the simulation is 900 s. The extreme levels for the percentage of malicious nodes in the network population are motivated by the desire to show the behavior of a network with a very high but probably still manageable fraction of malicious nodes. This should then be compared to a totally benign network situation. The number of applications, i.e. ongoing CBR connections, were chosen bearing in mind both the capacity of nodes as well as scalability.

Table 3 shows the variation due to three factors, with a constant setting of one third of the network population being malicious nodes and 10 applications taking place in the network. It shows that the protocol, whether defenseless or fortified, has the greatest impact on the number of dropped packets in the presence of malicious nodes, which confirms the intuitive expectation. What is more surprising, is to see that the pause time alone, i.e., the dynamicity of the networks has very little influence relative to the other factors. With the exception of the combination of the protocol (which caused the most variation by itself) and the pause time (which had the smallest contribution to the variation by itself), all the combinations contribute significantly to the variation, which should not be neglected in the analysis. Although the percentage of malicious nodes has been kept at the constant of one third, the number of nodes also contributed significantly to the variation and was present in all the combinations that mattered.

Factor	Metric: dropped packets
A (Number of nodes)	9.97 %
B (Protocol)	60.78 %
C (Pause time)	1.17 %
AB	9.39 %
AC	10.11 %
BC	0.73 %
ABC	7.85 %
T (Total)	100.00 %

Table 3: Variation due to three factors and their combinations, 10 applications, one third malicious nodes

The results in absolute numbers of dropped packets are listed in Table 4. The experiments are shown with their combination of factors used according to Table 3.

Combination	dropped packets
A10BfC0	30.83
A10BdC0	551.67
A10BfC600	58.67
A10BdC600	1309.00
A50BfC0	118.83
A50BdC0	2836.00
A50BfC600	5.50
A50BdC600	1354.00

Table 4: Mean number of dropped packets for each experiment with ten runs

Table 5 shows the variations in the number of dropped packets due to five factors and relevant combinations. The combinations of factors are not listed if their individual contribution to the variance turned out to be negligible. In these 2^5r experiments, the protocol state does not have as much influence on the variance as in the 2^3r experiments. This can be explained by the fact that the number of packets dropped in a fortified network in the presence of one third malicious nodes is only on the order of tens or hundreds, whereas in a defenseless network thousands of packets are dropped. The fortified network behaves almost as well as a benign network, thereby levelling the difference. Again, the pause time only contributes an almost negligible share to the variation relative to the other factors. As can be expected the number of malicious nodes is responsible for a significant portion of the variation, when varied between zero and one third. Prominent among other combinations, which also contribute, the combination of the protocol and the number of malicious nodes causes quite a significant portion of the variance.

Factor	Metric: dropped packets
A (Number of nodes)	4.97%
B (Protocol)	15.17%
C (Pause time)	0.07%
D (Percentage of malicious)	17.68%
E (Number of Applications)	5.00%
AB	4.97%
AD	4.81%
BC	5%
BD	16.17%
CD	4.78%
ABD	4.81%
BCD	4.78%

Table 5: Variation due to five factors and relevant combinations

5 Future Work

The next step is to investigate the behavior of the CONFIDANT protocol over time, i.e., considering transient removal and convergence, to determine whether the performance converges and when.

Regarding the simulation implementation, we are currently working on enhancements such as a limited number of friends, timeouts for reputations, and different thresholds for events that are used to infer the malicious character of nodes. The thresholds directly impact the tolerance of the node's reputation. The next step will be to extend the implementation to observable attacks other than forwarding defection, e.g. route diversion.

For the CONFIDANT protocol itself, of interest are, for example, methods to efficiently distribute reputation information in order to avoid malicious nodes as early as possible.

The CONFIDANT protocol assumes that nodes are authenticated and that no node can pretend to be another in order to get rid of a bad reputation. Some mechanisms to ensure that are being investigated, e.g. those mentioned in Section 2. Future work will be to evaluate and incorporate suitable solutions in the CONFIDANT protocol.

6 Conclusions

Mobile ad-hoc networks exhibit new vulnerabilities to malicious attacks or denial of cooperation. When designing protocols for these networks, special care has to be taken to include fairness mechanisms for the increased requirements in this environment. New ways of distributing trust can be implemented by introducing the notion of friends and making cooperation pay off. This paper recognizes the special requirements of mobile ad-hoc network in terms of cooperation, robustness, and fairness, and analyzes the performance of a scheme to cope with them by retaliating for malicious behavior and warning affiliated nodes to avoid bad experiences. Nodes learn not only from their own experience, but also from observing the neighborhood and from the experience of their friends.

Observable attacks on forwarding and routing can be thwarted by the suggested CONFIDANT scheme of detection, alerting and reaction. Performance analysis by means of simulation shows a significant improvement in terms of goodput when DSR is fortified with the CONFIDANT protocol extensions. The overhead for this increase is very low. The CONFIDANT protocol is scalable in terms of the total number of nodes in a network and performs well even with a fraction of malicious nodes as high as 60%.

References

[AS99] Ross Anderson and Frank Stajano. The resurrecting duckling. Lecture Notes in Computer Science, Springer-Verlag, 1999.

[BB02] Sonja Buchegger and Jean-Yves Le Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks. In *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 403 – 410, Canary Islands, Spain, January 2002. IEEE Computer Society.

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of IEEE Conference on Security and Privacy, Oakland, CA*, 1996.

[BH00] Levente Buttyán and Jean-Pierre Hubaux. Enforcing service availability in mobile ad-hoc wans. In *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Boston, MA, USA, August 2000.

[BH01] Levente Buttyán and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. Technical Report DSC/2001/046, EPFL-DI-ICA, August 2001.

[Daw76] Richard Dawkins. *The Selfish Gene.* Oxford University Press, 1989 edition, 1976.

[Jai91] Raj Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, New York, 1989 edition, 1991. All you need to know about performance analysis.

[JM99] Dave B. Johnson and David A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, Mobile Ad Hoc Network (MANET) Working Group, IETF, October 1999.

[MB96] S. L. Murphy and M. R. Badger. Digital signature protection of the OSPF routing protocol. IEEE, 1996.

[MGLB00] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pages 255–265, 2000.

[PPSW97] Andreas Pfitzmann, Birgit Pfitzmann, Matthias Schunter, and Michael Waidner. Trusting mobile user devices and security modules. In *Computer*, pages 61–68. IEEE, February 1997.

[RZFK00] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[Sch00] Bruce Schneier. *Secrets and Lies. Digital Security in a Networked World.* John Wiley $ Sons, Inc, 1 edition, 2000.

[SMGLA97] Bradley R. Smith, Shree Murthy, and J.J. Garcia-Luna-Aceves. Securing distance-vector routing protocols. In *Proceedings of Internet Society Symposium on Network and Distributed System Security, San Diego, CA*, pages 85–92, February 1997.

[YNK01] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-aware ad-hoc routing for wireless networks. MobiHOC Poster Session, 2001.

[ZBG98] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A library for parallel simulation of large-scale wireless networks. Proceedings of the 12th Workshop on Parallel and Distributed Simulations–PADS '98, May 26-29, in Banff, Alberta, Canada, 1998.

[ZH99] Lidong Zhou and Zygmunt Haas. Securing ad hoc networks. In *IEEE Network magazine, special issue on networking security, Vol. 13, No. 6, November/Dezember*, pages 24–30, 1999.

[Zim93] P. Zimmerman. PGP user's guide, 1993.