Nodes Bearing Grudges:
Towards Routing Security, Fairness, and Robustness in Mobile 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

Devices in mobile ad hoc networks work as network nodes and relay packets originated by other nodes. Mobile ad hoc networks can work properly only if the participating nodes cooperate in routing and forwarding. For individual nodes it might be advantageous not to cooperate, though. The new routing protocol extensions presented in this paper make it possible to detect and isolate misbehaving nodes, thus making it unattractive to deny cooperation. In the presented scheme, trust relationships and routing decisions are made based on experienced, observed, or reported routing and forwarding behavior of other nodes. A hybrid scheme of selective altruism and utilitarianism is presented to strengthen mobile ad hoc network protocols in their resistance to security attacks, while aiming at keeping network throughput, or goodput, high. This paper focuses particularly on the network layer, using the Dynamic Source Routing (DSR) protocol as an example.

1. Introduction

Mobile ad hoc networks do not rely on any fixed infrastructure but communicate in a self-organized way. Their properties lead to new vulnerabilities to attacks unknown in infrastructure-based or wired networks. In this paper we address these requirements for more fairness and robustness of mobile ad hoc networks. An example of a mobile ad hoc network is being developed within the Terminodes[1] project [8] which is about large mobile ad hoc networks. It is different from other mobile ad hoc networks as proposed in the MANET (mobile ad hoc networks) working group of the IETF [11] in that the network is a *wide-area*, self-organized network. The wide area aspect raises scalability issues and, furthermore, the Terminodes network is not limited to an organization who could enforce cooperation. Therefore, there is a need for incentives to cooperate in order to encourage the nodes to forward packets, although doing so consumes their resources. The issues discussed in this paper are relevant for both Terminodes and MANET-style mobile ad hoc networks.

[1] http://www.terminodes.org

One of the protocols presented and discussed in the MANET working group of the IETF is the Dynamic Source Routing (DSR) protocol [9]. It is briefly presented in Section 5 and serves as an example of security vulnerabilities and what can be done to eliminate them. An extension to DSR is proposed for this purpose.

1.1. Special Security Issues for Mobile Ad Hoc Networks

In addition to authentication, integrity, confidentiality, availability, access control and non-repudiation (see [16] for details), which have to be addressed differently in a mobile, wireless, battery-powered and distributed environment, mobile ad hoc networks raise the following security issues:

Cooperation and fairness: There is a trade-off between good citizenship, i.e. cooperation, and resource consumption, so nodes have to economize on their resources. At the same time, however, if they do not forward messages, others might not forward either, thereby denying them service. Total non-cooperation with other nodes and only exploiting their readiness to cooperate is one of several boycotting behavior patterns. Therefore, there has to be an incentive for a node to forward messages that are not destined to itself. Attacks include incentive mechanism exploitation by message interception, copying, or forging; incorrect forwarding; and bogus routing advertisement.

Confidentiality of location: In some scenarios, for instance in a military application, routing information can be equally or even more important than the message content itself [6].

No traffic diversion: Routes should be advertised and set up adhering to the chosen routing protocol and should truthfully reflect the knowledge of the topology of the network. By diverting the traffic in the following ways, nodes can work against that requirement:

Routing: To get information necessary for successful malicious behavior, nodes can attract traffic to themselves or their colluding nodes by means of false routing advertisements. Although only suitable for devices that have enough power, a lot of information can be gathered this way by malicious nodes for later use to enable more sophisticated attacks.

Denial-of-service attacks can be achieved by bogus routing information (injecting of incorrect routing information or replay of old routing information or 'black hole routes') or by distorting routing information to partition the network or to load the network excessively, thus causing retransmissions.

Forwarding: Nodes can decide to forward messages to partners in collusion for analysis, disclosure, or monetary benefits; or may decide not to forward messages at all, thus boycotting communications.

1.2. Motivation for Attacks

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. In contrast, Section 3 provides a rationale why cooperation can pay off.

1.3. Thwarting Attacks: Objectives

We would like to achieve the following effect with our protocol: malicious behavior and non-cooperation should be punished and should not pay off. Detection of this kind of behavior is key but not the only point. The detection has to lead to a reaction of other nodes such that it results in a disadvantage for the malicious node. This punishment can very well be by means of isolation, but not positive isolation in being isolated from the society's duties but above all the society's rights. Packets of malicious nodes should, upon detection of the node being malicious, not be forwarded by the normally behaving nodes. If, however, a node was wrongly accused of being malicious or turns out to be a repenting criminal equivalent who 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.

Prevention, detection and reaction: According to Schneier [14], 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.

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 novel approach to increase fairness, robustness and cooperation which is motivated in Section 3, outlined as a protocol in Section 4 and applied to DSR in Section 5. The rest of this paper consists of an outline of future work in Section 6 and the concluding Section 7.

2. Related Work

Anderson and Stajano [1] 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. Haas employs threshold security, allowing for several corrupted nodes or collusions [7]. Garcia-Luna-Aceves et al. [15] looked at security of distance vector protocols in general. For the Terminodes project, incentives to cooperate by means of so-called nuglets [4] that serve as a per-hop payment in every packet have been suggested by Buttyan et al. to ensure forwarding. The scheme suggested here in the following sections addresses additional issues in the network layer such as traffic diversion.

Marti et al. [12] 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 used path 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.

We would like to achieve the contrary, namely to make only good behavior pay off in terms of service and reasonable power consumption. With the scheme we present in this paper, it is disadvantageous for nodes to behave maliciously. An example of how this can work in biology is presented subsequently.

3. Bearing Grudges: The Selfish Gene

As explained in Richard Dawkins' 'The Selfish Gene' [5], reciprocal altruism is beneficial for every biological system when favors are granted simultaneously, so there is an intrinsic motivation for cooperation due to instant gratification. The benefit of behaving well is not so obvious in the case of a delay between granting a favor and repayment, which is the case when, in mobile ad hoc networks, nodes forward for each other. A biological example used in 'The Selfish Gene' [5] 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' which always help and 'cheats' which 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' which 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 head.

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 of having one's head groomed, a loss leading to extinction and profit leading to multiplication of the species. The rationale is as follows: The suckers help more than they get favors due to the large number of cheats, so the number of suckers decreases, while 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 don't 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.

4. Application and Improvements: The Grudger Protocol

4.1. From Birds to Network Nodes

Defining suitable cost and profit to routing and forwarding favors and keeping a history of experiences with non-cooperating nodes achieves the same as the grudger species, 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 suggest the following ideas, which are incorporated in a protocol explained in the next section, to speed up the winning of grudger nodes:

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

The metric for success is the resulting *goodput*, i.e. the number of bits per unit of time forwarded to the correct destination, minus any bits lost or retransmitted.

4.2. Components in Each Node

The protocol containing the improvements to the grudger's scheme 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 and they are described in detail subsequently:

4.2.1 The Monitor (Neighborhood Watch)

In a 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 neighbors of the neighborhood watch can detect deviances 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

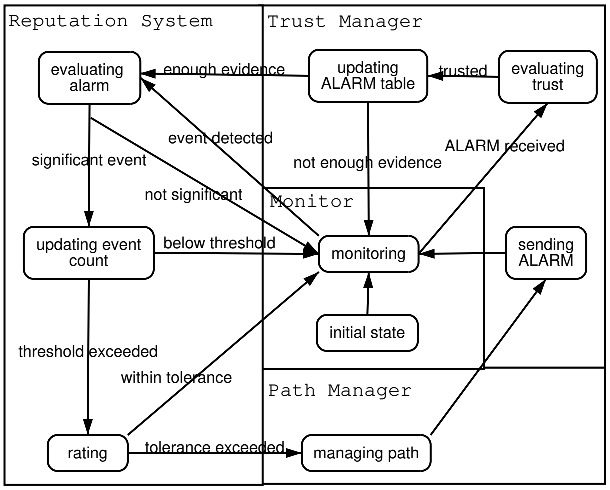

Figure 1. Trust architecture and finite state machine within each node.

while listening to the transmission of the next node, any content change can also be detected. In this paper we focussed on the detection of observable routing and forwarding misbehavior in DSR as listed in section 5.2. In general, the following types of misbehavior can be indicated:

- o no forwarding (of control messages nor data),
- o unusual traffic attraction (advertises many very good routes or advertises routes very fast, 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).

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

4.2.2 The Trust Manager

In an ad hoc environment, trust management has to be distributed and adaptive [2]. 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. Incoming ALARMS originate from outside friends, whereas outgoing ALARMS are generated by the node itself after having experienced, observed or been reported malicious behavior. The following functions are performed by the trust manager:

- o Trust function to calculate trust levels,
- o Trust table entries management for trust level administration,
- o Forwarding of ALARM messages,
- o 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 [18], several levels of trust can be expressed, e.g. 'unknown', 'none', 'marginal', and 'complete'. When PGP is calculating 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 to require a different number of marginally trusted signatures to judge a key as valid.

The trust manager consists of the following components:

- o Alarm table containing information about received alarms,
- o Trust table managing trust levels for nodes,
- o Friends list containing all friends a node sends alarms to.

The trust manager administers a table of friends, i.e. nodes that are candidates to receive ALARM messages from a given node, and how much they are trusted. Trust is important when making a decision about the following issues:

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

4.2.3 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. [13].

In order to avoid centralized rating, local rating lists and/or black lists are maintained at each node and potentially exchanged with friends. The nodes can include *black* *sheep* in the route request to be avoided for routing, which also alarms nodes on 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 and thus 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 enough evidence for 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:

- o Own experience: greatest weight,
- o Observations: smaller weight,
- o Reported experience: weight function according to PGP trust.

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 hopefully be the exception and not the rule, the reputation system is built on negative experience rather than positive impressions. The questions of positive change and timeout are still to be addressed in detail.

4.2.4 The Path Manager

The path manager performs the following functions:

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

4.3. Information Flow in 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, as defined initially for the type of node, for different nodes can have different security requirements, it is checked whether the event has occurred more often than a predefined threshold high enough to distinguish deliberate malicious behavior from simple coincidences such as collisions. If that occurrence threshold is exceeded, the reputation system updates the rating of the node that caused that event. If the rating subsequently turns out to be intolerable, the information is passed on to the path manager that 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 next subsection.

4.4. Information Flow Between Nodes

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).

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 enough 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 4.3. Enough 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.

Authentication is a prerequisite for the protocol to work and assumed to exist here. One way to achieve authentication is by using PGP along with distributed certification authorities. Without authentication, nodes can denounce each other at will and a trust management scheme is not feasible.

5. Extension to DSR

5.1. 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 at Carnegie Mellon University [9]. 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 the preference given to the route and the longer it will stay in the cache. In case of a link failure, the node that can not forward the packet to the next node sends an error message towards the source. Routes that contain a failed link, can 'salvage' the route by bypassing the bad link.

5.2. Attacking DSR

We found the following ways of attacking DSR, targeting availability, integrity, confidentiality, non-repudiation, authentication, access control or any combination thereof:

1) Incorrect forwarding: acknowledge ROUTE REQUEST, send new request or do not forward at all. This works only until upper layers find out.

2) Bogus routing information or traffic attraction: reply to ROUTE REQUEST, also gratuitous, to advertise a non-existent or wrong route.

3) Salvage a route that is not broken. If the salvage bit is not set, it will look like the source is still the original one.

4) Choose a very short reply time, so the route will be prioritized and stay in the cache longer.

5) Set good metrics of bogus routes for priority and remaining time in the cache.

6) Manipulate flow metrics for the same reason.

7) Do not send error messages in order to prevent other nodes from looking for alternative routes.

8) Use bogus routes to attract traffic to intercept packets and gather information.

9) Use promiscuous mode to listen in on traffic destined for another node.

10) Cause a denial-of-service attack caused by overload by sending route updates at short intervals.

5.3. Detection of Attacks in DSR

With the exception of the promiscuous listening in 9), all of the attacks listed above correspond to observable events the monitor component in each node can detect either at once or at the latest when they happen repeatedly:

1) Forwarding: this can be detected by *passive acknowledgement*, i.e. keeping a copy of a packet until having confirmed correct forwarding by listening to the transmission of the next hop node.

2), 8) Bogus routing: a strong indication would be when an intermediate node sees itself advertised on a route it does not have. As a last resort, if a node cannot tell whether a route is real or bogus, it can at least detect the lack of forwarding as in 1). Unusually increased frequency of route advertising can be detected as in 10).

3) Salvaging: indicated by the reception of a salvaged packet without having received a link error message first.

4) Reply time too short: can be detected by comparing reply time to actual route length.

5), 6) Metrics of bogus routes too good: detectable by comparing metrics to actual quality.

7) Lack of error messages: indicated in the case when a node receives a link error message from its own link layer but no explicit error message by other nodes in the range.

10) Route updates too frequent: detectable by keeping timestamp of last update to compare.

Depending on the type of device, some events may be less important such that the effort to monitor and detect these particular events may be omitted.

5.4. Grudging Nodes in DSR

The suggested scheme works as an extension to a routing protocol. In this example, normal DSR information flow (ROUTE REQUEST, ROUTE REPLY messages) as explained in Section 5 takes place. Once non-cooperative behavior has been detected and exceeds threshold values, an ALARM message is sent. Figures 2 through 5 show the flow of messages and data from route discovery to the detection of malicious behavior and subsequent rerouting. In more detail:

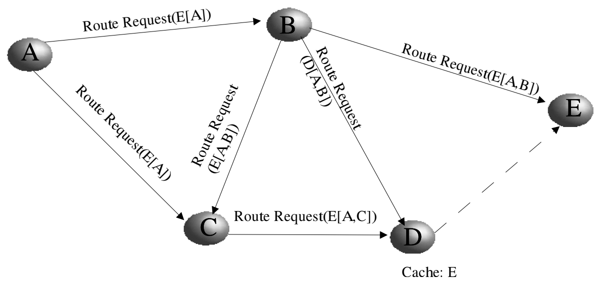

Figure 2. Route request: *A wants to send to E.*

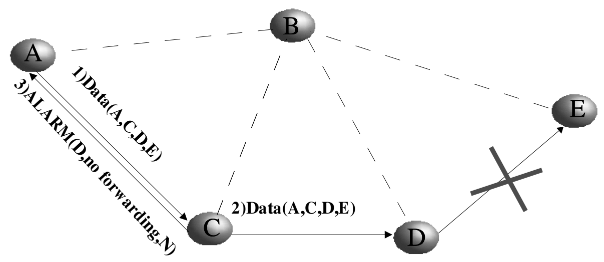

Figure 4. Data flow and alarm: *A sends data and receives an ALARM from C that D does not forward.*

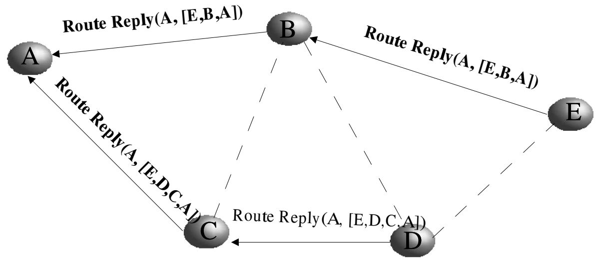

Figure 3. Route reply: *both D and E know a path to E.*

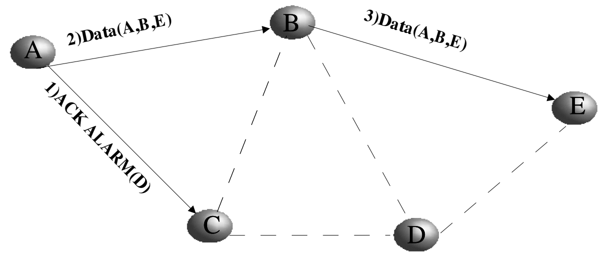

Figure 5. Act on alarm: reroute: *A uses an alternate path to E.*

Figure 2 shows DSR route discovery for a path from node A to node E. Every node forwards the request to its neighbors unless it has already received the same route request or has a path cache entry for the desired destination.

Figure 3 shows the reply messages of the destination node itself, node E, and from node D, which has a path to E. The reply message contains the reversed source route to the destination and is sent to the source. In the case of unidirectional links, or if generally the route can not be reversed, node E would send the reply along a path to A that it has in its route cache. If there is no path to A in the route cache, E has to perform a route discovery itself to get to A. In this route request, the already found path from A to E is included.

In Figure 4 data flow is from node A to node E via node C and D. In this case, node A has chosen this route according to some metrics and preferred it over the route via B. During the data flow, node C detects that node D does not behave correctly. In this example, node D does not forward the data destined for node E. After the occurrence of the bad behavior of node D was observed by node C for a number exceeding a threshold, node C triggers an ALARM message to be sent to the source, node A. Upon reception of the ALARM message as shown in Figure 5, node A acknowledges the message to the reporting node C and decides to use the alternate path via node B to send the data to the destination node E.

5.5. Analysis

Detailed simulations in GloMoSim [17] are under way, see [3] for a sketch of the simulation design and methodology. Preliminary results have shown that even if the DSR protocol is only fortified by reacting to forwarding defection, in the presence of malicious nodes only the first few packets are dropped (according to the defined threshold plus the time it takes to react) in the fortified version of DSR, whereas all of the packets are dropped using the normal defenseless DSR protocol. For analytical evaluation we are investigating the use of Game Theory. The increased security will come at the price of some overhead, the exact amount of which is being investigated and simulated, but inherently the price of no communication at all due to malicious nodes is higher than any overhead by one extra message in the protocol,the ALARM message.

6. Future Work - Next Steps

The next steps will consist of implementing more of the approaches discussed so far in simulations for evaluation and performance analysis. The focus is on finding a sustainable relationship between the total number of nodes in the network, the number of malicious nodes that can be tolerated and the number of friends per node needed to achieve that. We are analyzing the scalability, the cost/benefit ratio, and the goodput increase and overhead for achieving security as defined in this paper.

We will look at further issues that have not been addressed in this paper, for instance what happens to a node in a remote location, where friends might be far away, or how to deal with colluding nodes. Other interesting issues include rumor spreading and transitive trust in a 'small world' [10] and how it could be used to locate friends.

7. Conclusions

Mobile ad hoc networks exhibit new vulnerabilities to security attacks. As opposed to traditional networks, mobile ad hoc networks do not rely on any infrastructure and central authorities, they can be highly dynamic and mobile and operate over unreliable wireless media. When designing protocols for mobile ad hoc networks, special care has to be taken to include security mechanisms for the increased requirements in this environment. New ways of distributing trust can be implemented by introducing the notion of friends and making their cooperation pay off. This paper identifies the special requirements of mobile ad hoc network security, robustness, and fairness, and it introduces 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 scheme of detection, alerting and reaction. Security is a major challenge for mobile ad hoc networks, because good citizenship can not be assumed in an open world, where anyone can join the network. Depending on the extent to which the security issues are addressed, people might be reluctant to use mobile ad hoc networks.

References

[1] R. Anderson and F. Stajano. The resurrecting duckling. Lecture Notes in Computer Science, Springer-Verlag, 1999.

[2] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of IEEE Conference on Security and Privacy, Oakland, CA*, 1996.

[3] S. Buchegger and J.-Y. L. Boudec. IBM Research Report: The Selfish Node: Increasing Routing Security in Mobile Ad Hoc Networks. RR 3354, 2001.

[4] L. Buttyan and J.-P. Hubaux. Enforcing service availability in mobile ad-hoc wans. MobiHOC, 2000.

[5] R. Dawkins. *The Selfish Gene*. Oxford University Press, 1989 edition, 1976.

[6] A. Fasbender, D. Kesdogan, and O. Kubitz. Variable and scalable security: Protection of loccation information in mobile IP. In *Proceedings of the 46th IEEE Vehicular Technology Conference, Atlanta*, pages 963–967, 1996.

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

[8] J.-P. Hubaux, J.-Y. L. Boudec, S. Giordano, and M. Hamdi. The terminode project: Towards mobile ad hoc WANs. In *Proceedings of MOMUC'99 San Diego*, 1999.

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

[10] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. Cornell Computer Science Technical Report 99-1776, 1999.

[11] MANET. http://www.ietf.org/html.charters/manet-charter.html, 2000.

[12] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pages 255–265, 2000.

[13] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

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

[15] B. R. Smith, S. Murthy, and 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.

[16] W. Stallings. *Network and Internetwork Security*. IEEE Press, 2 edition, 1995.

[17] X. Zeng, R. Bagrodia, and M. 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.

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