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, Switzerland
sob@zurich.ibm.com

Jean-Yves Le Boudec
EPFL-IC-LCA
Ecublens
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.

Categories and Subject Descriptors

C.2.2 [Computer Systems Organization]: Computer-Communication Networks—*Network Protocols*

General Terms

Algorithms, Performance, Economics, Reliability, Security, Human Factors

Keywords

routing, cooperation, reputation, mobile ad-hoc networks, fairness, robustness, trust

MOBIHOC'02, June 9-11, 2002, EPFL Lausanne, Switzerland.

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 about 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 [8]. In a nutshell, it works as follows: Nodes send out a ROUTE REQUEST message, all nodes that receive this message put themselves into the source route and forward it to their neighbors, 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 route 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 are gratuitous. After receiving one or several routes, the source selects the best (by default the shortest), stores it, and sends messages along that path. The better the route metrics (number of hops, delay, bandwidth, or other criteria) and the sooner the REPLY arrives at the source, the higher the preference given to the route and the longer it will stay in the cache. When a ROUTE REPLY arrives very quickly after a ROUTE REQUEST has been sent out this is an indication of a short path, since the nodes are required to wait for a time corresponding to the length of the route they can advertise, before sending it.. This is done in order to avoid a storm of replies. 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 vari-

ous advantages by malicious behavior, such as

- better service than cooperating nodes,
- monetary benefits by exploiting incentive measures or trading confidential information,
- saving power by selfish behavior,
- preventing someone else from obtaining proper service,
- extracting data to get confidential information, and so on.

Several routing and forwarding attacks on DSR have been described in [3]. We aim at protection against the following types of misbehavior.

- No forwarding (of control messages or data).
- Traffic deviation: unusual traffic attraction (advertises many excellent routes or advertises routes very rapidly, so they are deemed good routes) or the opposite (claims to have only bad routes).
- Route salvaging, i.e., rerouting to avoid a broken link, although no error has been observed.
- Lack of error messages, although an error has been observed, or vice versa.
- Unusually frequent route updates.
- 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 [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 (see also Section 2 for a discussion on strong prevention mechanisms such as [4]).

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.

Thus, in our scheme detection has to trigger a response, i.e., 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 offender that is no longer malicious and that 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 [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. 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 [18] 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 yet been published.

Smith, Murthy, and Garcia-Luna-Aceves [15] 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 as well as including predecessor information in routing updates. Digital signatures have also been suggested for the OSPF routing protocol by Murphy and Badger [11]. 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 [4] that serve as a per-hop payment in every packet or counters [5] 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 a 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 [12], but might prove inevitable. As opposed to nuglets and counters, the CONFIDANT protocol does not need tamper-proof hardware for itself, since a malicious node neither knows 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 [9] observed that throughput increased in mobile ad-hoc networks by complementing DSR with a 'watchdog' for detection of non-forwarding nodes and a 'pathrater' (for trust management and routing policy, every path used is rated), which enable nodes to avoid non-forwarding nodes in their routes. Ratings are kept about every node in the network and the rating of actively used nodes is updated periodically. 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, we would like to achieve the opposite with our protocol.

The Security-aware Ad-hoc Routing (SAR) protocol by Yi, Naldburg, and Kravets [16] 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' [6], 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 as obvious when there is a delay between granting a favor and the repayment. This occurs when, in mobile ad-hoc networks, nodes forward on behalf of each other. An ecological example used by Dawkins [6] 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 obtaining 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 more favors than they receive because of the large number of cheats, so the number of suckers decreases, whereas the number of cheats increases. The grudgers also suffer 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 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 results 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 also 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.

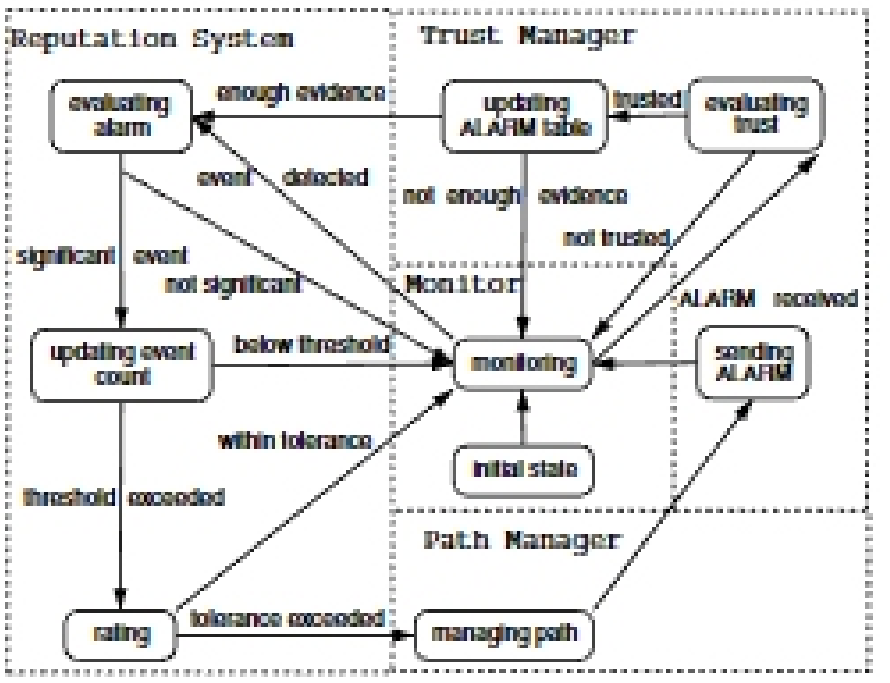

Figure 1: Trust architecture and finite state machine within each node.