Real Life Experience of Cooperation Enforcement Based on Reputation (CORE) for MANETs

Claudio Lavecchia, Pietro Michiardi, Refik Molva[1]
Institut Eurecom, 2229, Route des Crêtes
06560 Valbonne Sophia Antipolis
{first name.last name}@eurecom.fr

Abstract

Cooperation enforcement in mobile ad-hoc networks has become a hot topic within the scientific community. Entities belonging to a mobile ad-hoc network are prone to selfishness because being cooperative and participating to basic network functions such as routing and packet forwarding involves resource consumption for the benefit of others. Different approaches have been proposed to promote cooperation in such environments. An implementation of the cooperation enforcement mechanism named CORE [5] is presented in the following as well as a demonstration of its usage on a MANET testbed.

1. Introduction

The hype of trust establishment schemes that the research community is witnessing in recent years results in a proliferation of such mechanisms that target various issues rising at different layers of a communication system.

A particular instance of trust establishment schemes is represented by reputation mechanisms, in which the trust metric takes the form of a reputation measure associated to each entity taking part in a digital transaction. Reputation can be defined as the level of trust inspired by entities based on observations made on entities' past behavior. Intuitively, reputation can be thought of as a metric that drives and regulates the formation of dynamic communities that shares interests and have common goals.

A typical setting in which reputation schemes are used to regulate the formation and the survivability of digital communities is represented by peer-to-peer (P2P) file sharing systems. In P2P communities, reputation can be used to baffle greediness and selfishness of peers that make and use the system while at the same time suffer from the dilemma of constrained resources. Indeed, is there a reason to assume the volunteer participation to the community welfare if no countermeasures are in place to stimulate a fair distribution of the costs incurred by each individual to the community operation? In general the answer is negative, as it has been demonstrated by recent studies [2, 3]. Another interesting domain of application of this type of trust establishment schemes is offered by the mobile ad hoc networking paradigm. In mobile ad hoc networks (MANET), node participation to basic networking functions such as routing and packet forwarding is of fundamental importance. Recent studies [4] show that network performance can be severely degraded even when only a small fraction of the nodes that are part of an ad hoc network deny participation to the network operation. Again, scarce resources are at the origin of a selfish node behavior whereby nodes (and end users operating those nodes) do not want to share the (energetic) costs incurred by the network operation for the benefit of others.

In this paper we focus on a reputation system used to stimulate node participation to the execution of the packet forwarding function in MANETs. We present an overview of the CORE [5] reputation system architecture from an implementation point of view and detail the demonstrative setting in which we carried out the proof-of-concept validation of CORE. The reader should refer to [6] for a detailed description and analysis of CORE.

2. CORE System Architecture

The CORE reputation system uses the watchdog mechanism [7]. A watchdog can be defined as a software component installed on the nodes of a network with the aim of observing neighboring nodes behavior with respect to participation to basic network functions. The solution proposed hereafter addresses only the packet forwarding function and has been implemented and tested on a MANET testbed made of nodes relying on off-the-shelf 802.11b hardware. A MANET node implementing the watchdog mechanism must be able to overhear all the packets that are sent within its wireless channel. To do so, the 802.11b WLAN adapter needs to be operated in the so-called promiscuous mode. This functionality is implemented at WLAN adapter firmware and driver level and enables the WLAN adapter to pass all the packets received to upper layers for further processing. In our MANET testbed we use Dell TrueMobile 1150 WLAN adapters that are operated by the Orinoco driver which works in promiscuous mode out of the box. Once the WLAN adapter is set in promiscuous mode, the packets are captured using the pcap [8] C libraries. Those libraries are available for Windows OS as well, making the porting effort of the CORE mechanism to Windows OS acceptable. The CORE reputation system has been implemented as a Linux daemon, the implementation architecture is illustrated

[1] This research was partially supported by the Information Society Technologies program of the European Commission, Future and Emerging Technologies under the IST-2001-38113 MOBILEMAN project and by the Institut Eurecom.

in Figure 1. Here follows the detailed description of the modules that compose the system:

Sniffer Module: monitors the packets that pass across layer 2 of the TCP/IP stack. This module passes the relevant fields of packet headers to the analyzer module for further analysis in the form of packet descriptors.

Analyzer Module: Receives packet descriptors from the sniffer module and analyzes those descriptors to deduce whether the neighbors are being cooperative or not.

The analyzer module includes an expectation table. Packet descriptors that correspond to packets for which forwarding is expected by a neighbor are stored in this table. The scheduler included in the analyzer module triggers a timeout each time that a packet descriptor is written in the expectation table. Upon timeout expiration on a packet descriptor, the analyzer verifies if the corresponding packet has been forwarded from the neighbor to the next hop. In such case it deduces that the neighbor that forwarded the packet has been cooperative and a positive observation is passed to the reputation module. If the packet has not been forwarded before the timeout expiration, the node that was expected to forward the packet is suspected to be selfish and a negative observation is passed to the reputation module. The analyzer module features an ARP interface that is needed to perform some basic neighbor discovery functions.

Reputation Module: In the original version of CORE [5], the reputation value associated to a node is evaluated in a sophisticated way. The interested reader should refer to [6] for a detailed description and analysis of advanced reputation evaluation functions. For sake of simplicity and in order to provide a proof of concept evaluation of CORE, the real life implementation of our cooperation enforcement mechanism is based on a simpler reputation function described hereafter. The reputation module uses a weighted average function to calculate reputation values for neighbors according to observations provided by the analyzer and stores those values in a reputation table. When the reputation of a neighbor falls below a given threshold, it issues punishment requests to the punishment module.

In the current implementation the reputation function is given by:

$$R_K(a) = \sum_{k=0}^{k=B-1} W_k \frac{Obs_a(K-k)}{B} \quad (1)$$

where:

a is the node that is being observed by the watchdog.

B is the number of observation that the node that executes the watchdog keeps in its local observation buffer.

W_k is the weight given to the k-th observation.

K is the actual absolute discrete time.

$Obs_a(K-k)$ is the value of the observation at time $K-k$

Possible observation values are: (+1) if the watchdog detects a cooperative behavior, (-1) if the watchdog detects a selfish behavior.

Punishment Module: Punishes selfish neighbors by denying packet forwarding through the "iptables" Linux framework [9]. The proposed architecture has the advantage of identifying clear interfaces among the system modules, thus allowing the interoperability of the watchdog module with other systems that calculate and exploit the reputation of other nodes, such as for example the one proposed in [7]. Another advantage of this solution resides in the very limited network overhead introduced by its operation: the only additional traffic is a pair of ARP request/reply generated each time that a node running the watchdog analyzes packets that involve a previously unknown neighbor node.

3. CORE Demonstration

The implementation of CORE has been integrated on our MANET testbed. A demonstration has been developed to show the system behavior.

The MANET testbed is composed of 4 nodes, equipped with a WLAN adapter. Two of the nodes are laptops running Windows OS, the third one, where the watchdog software is executed is a laptop running Linux OS, the fourth node can be either a laptop or a Compaq iPaq PDA, in both cases it runs Linux OS.

The demonstration objectives are:

- To show how the CORE system maintains a reputation state for all the neighbors of a MANET node. Two states are possible for a neighbor: selfish and cooperative.
- To show how the economical approach that drove the CORE mechanism design effectively motivates a user that is leaning toward selfishness to reconsider his objectives and be cooperative.
- To show the effectiveness of inherent reintegration mechanism proposed by CORE.

The MANET testbed nodes are logically disposed in a row. We assume the existence of bidirectional wireless links between neighbor nodes. Nodes are disposed as follows:

A↔W↔S↔B

Ideally the physical distances between two non-neighboring nodes of the network are larger that the WLAN card transmission range. Two non-neighboring nodes of the network that wish to communicate will pass through an intermediate node that acts as a router. In reality, the WLAN cards transmission ranges are so that the physical separation in an indoor environment is achieved only when two non-neighboring nodes are more than some tens of meters far from each other and this is quite hard to handle in a demo environment. For this reason we use some "tricks" to logically separate two non-neighboring nodes. Those tricks involve the usage of firewalls or Linux "iptables" framework. Node W is the node that runs the watchdog. This node observes the behavior of its neighbors. According to the observations, node W maintains for each neighbor a reputation state. State for a neighbor can be either cooperative or selfish. With respect to the equation (1), the

watchdog parameters have been set as follows: $B=4$, $W_i=0.25$ (all the weights are equal).

Node S is the node that is operated by the selfish user. As a selfishness model, we assume that the administrator of this node stops forwarding traffic originated by nodes A or W and directed to node B and vice versa when the battery level of the its device falls below a given threshold.

We assume a non-selective selfishness model that is implemented by blocking the forwarding of other nodes traffic through the usage of Linux "iptables" framework.

Node B serves the HTTP connection requests coming from node W while node A acts as an FTP server for connection requests coming from node S.

During the demonstration, node S changes its state from cooperative to selfish. As a result of this change of attitude, node W HTTP connection requests to node B will fail.

CORE console running on node W shows the observations performed by the watchdog on the neighbors' behavior. When node S becomes selfish, a line appears on the CORE console to show the neighbor change of attitude. As soon as this transition is detected, the watchdog issues a punishment request to the CORE punishment module. Node S is immediately punished by node W, which stops forwarding node S packets. From this moment on, the FTP connection requests from node S to node A will stall.

When node S realizes that its FTP connection requests are blocked by node W, it decides to become cooperative again. This behavior transition is again captured by the watchdog on node W and shown on the CORE console. Node W immediately restarts forwarding node S packets. From this moment node S is reintegrated in the network and its FTP connections to node A will be successful again.

4. Conclusion

To the best of our knowledge, this article presents the first real-life implementation of a cooperation enforcement mechanism based on reputation which does not have an impact on underlying routing protocol adopted in the MANET (as opposed to [10]). Cooperation enforcement represents a fundamental building block for a heterogeneous MANET where no a priori trust relationships can be established among peers. The implementation of the software component described in this work constitutes a starting point for a thorough analysis (through measurements) of the impact of a cooperation enforcement mechanism on the operation of a heterogeneous MANET. For our future work we plan to investigate the behavior of the system in heavy load conditions to test the effectiveness of promiscuous listening. We will work on the fine tuning of the system parameters illustrated in (1) and study the introduction of an adaptive observation sampling factor to mitigate the CPU load generated by promiscuous listening and improve the responsiveness of the system behavior to the network conditions. We plan as well to investigate the impact of different reputation functions on the system accuracy.

References

[1] http://www.ebay.com

[2] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, Free-Riding and Whitewashing in Peer-to-Peer Systems, ACM SIGCOMM'04 Workshop on Practice and Theory of Incentives in Networked Systems (PINS), August 2004

[3] Kevin Lai, Michal Feldman, Ion Stoica, John Chuang, Incentives for Cooperation in Peer-to-Peer Networks, in Proceedings of Workshop on Economics of Peer-to-peer Systems, June 5-6 2003

[4] Pietro Michiardi, Refik Molva, Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks, in Proceedings of European Wireless 2002 Conference

[5] Pietro Michiardi, Refik Molva, CORE: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks, IFIP CMS02, Communication and Multimedia Security Conference, September 2002

[6] Pietro Michiardi, Cooperation enforcement and network security mechanisms for mobile ad hoc networks, PhD Thesis

[7] Sergio Marti, T.J. Giuli, Kevin Lai, Mary Baker, Mitigating routing misbehavior in mobile ad hoc networks, MobiCom00, International Conference on Mobile Computing and Networking, 2000

[8] http://www.tcpdump.org

[9] http://www.netfilter.org

[10] Sonja Buchegger, Jean-Yves Le Boudec, Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks, in Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), 2002

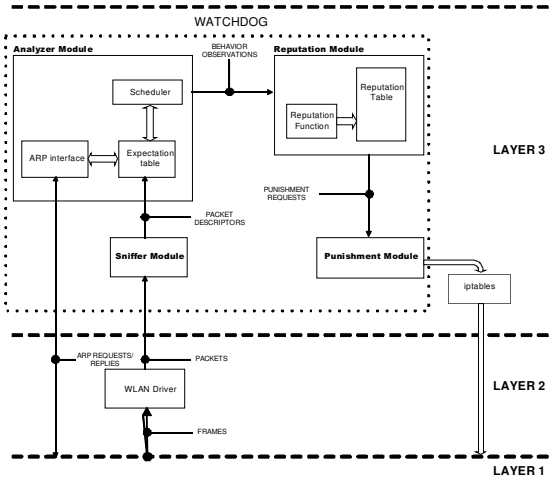

Figure 1. CORE Implementation Architecture