A fully decentralized trust management and cooperation incentives framework for wireless user-centric networks

Ballester Lafuente, Carlos

A Fully Decentralized Trust Management and Cooperation Incentives Framework for Wireless User-Centric Networks

THÈSE

présentée à la Faculté d'Economie et de Management
de l'Université de Genève par

Carlos Ballester Lafuente

sous la codirection de

Dr. Jean-Marc Seigneur

Prof. Dimitri Konstantas

pour l'obtention du grade de

Docteur ès Économie et Management
mention Systèmes d'Information

Membres du jury de thèse:

Prof. Giovanna Di Marzo Serugendo (Président du jury)
Dr. Jean-Marc Seigneur (Co-directeur de thèse)
Prof. Dimitri Konstantas (Co-directeur de thèse)
Prof. Sviatoslav Voloshynovskiy,
Prof. Yanjun Zuo, University of North Dakota

Thèse N° 5

Genève, le 24 Avril 2015

La Faculté d'Economie et de Management, sur préavis du jury, a autorisé l'impression de la présente thèse, sans entendre, par là, émettre aucune opinion sur les propositions qui s'y trouvent énoncées et qui n'engagent que la responsabilité de leur auteur.

Genève, le 24 Avril 2015

La Doyenne
Maria-Pia VICTORIA FESER

Impression d'après le manuscrit de l'auteur

Table of Contents

Résumé

Selon l'Union Internationale des Télécommunications (UIT), le nombre d'abonnés utilisant les services mobiles d'Internet à large bande est passé de 268 millions en 2007 à un impressionnant 2,1 milliards en 2013, représentant plus de 50% de l'utilisation d'Internet dans le monde. Ce précédent ainsi que l'émergence et la croissance rapide de l'informatique ubiquitaire et mobile permettant le réseautage social, le contenu contextuel généré par l'utilisateur n'importe quand n'importe où, la réalité augmentée ou même l'humain augmenté, a alimenté les besoins des utilisateurs d'une connectivité permanente, où qu'ils soient et en toutes circonstances.

Les réseaux sans fil d'aujourd'hui sont partiellement formés par des nœuds (par exemple les points d'accès Internet, smartphones, femtocells) qui sont détenus et portés par les humains. En tant que tel, ces architectures de réseaux centrés sur l'utilisateur (UCNs) donnent lieu à de nouvelles architectures Internet, où l'accès à large bande est complété par exemple par les Wireless Fidelity (Wi-Fi) clouds, ayant une forte implication de l'utilisateur d'Internet final. Cela représente un changement de paradigme dans l'évolution de l'Internet, du fait que l'utilisateur puisse être en contrôle des parties du réseau d'une manière qui peut être reconnue (ou non) par les parties prenantes de l'Internet. Dans de tels scénarios où plusieurs étrangers doivent interagir pour le bien de la transmission de données robustes, la confiance et l'incitation a la coopération sont d'une importance vitale car ceux-ci établissent un moyen pour les nœuds impliqués dans le système de communiquer entre eux d'une manière fiable pour partager services et informations, et surtout, pour former des communautés qui aident à maintenir les modèles de connectivité robustes.

Alors que les systèmes de gestion de la confiance peuvent être utilisés de manière isolée afin de fournir robustesse aux architectures réseau, l'incitation à la coopération peut être utilisée pour compléter et collaborer avec les systèmes de gestion de la confiance afin que les utilisateurs puissent en bénéficier en utilisant le système, donc encourageant le bon comportement de l'utilisateur. Dans l'ensemble, la relation entre la gestion de la confiance et de l'incitation à la coopération peut grandement aider dans la construction d'une architecture vraiment solide et fiable, cela étant le principal objectif de recherche de cette thèse.

Dans cette thèse nous avons conçu, développé et implémenté un cadre de gestion de la confiance et d'incitation à la coopération entièrement décentralisé pour les réseaux Wi-Fi centrés sur l'utilisateur et nous l'avons évalué à travers des simulations et par des tests à l'aide d'un prototype réel.

Le système que nous proposons est composé de trois éléments principaux, l'Identity Manager, le Trust Manager et le Cooperation Manager. L'Identity Manager doit être capable de traiter avec de multiples crypto-ids par utilisateur. Le Trust Manager est composé d'un module d'adaptation de la confiance dispositionnel et par une métrique résistante aux attaques de genre Sybil tirée de l'ouvrage du Seigneur [1]. Le Cooperation Manager doit fournir les bonnes incitations telles que des points ou une monnaie virtuelle, qui peuvent être échangés contre des services et peuvent être acquises lorsqu'ils leur sont

donnés, pour aider la gestion de la confiance à parvenir à une architecture globale plus sûr et plus fiable.

Grâce à la simulation, nous avons prouvé que l'adaptation de la confiance dispositionnelle peut améliorer la prévention des nœuds malveillants à hauteur de 60% -70% dans des environnements très hostiles, qui sont définis comme des environnements où la large majorité des nœuds sont malveillants, soit en ne fournissant pas le service promis soit en effectuant des attaques telles que les attaques Sybil, blanchissement, autopromotion et similaires. En ce qui concerne le schéma d'incitation a la coopération, nous avons démontré avec des simulations que notre schéma d'incitation améliore les conversions malveillantes dans près de 40% de cas de plus que ceux de la métrique comparée, qui est celle des recommandations crédibles par Koutrouli et al. [2]. Cette amélioration est due au fait que nos mesures d'incitation à la coopération se concentrent sur le niveau le plus haut de la fonctionnalité du système, qui est le partage et d'obtenir la connectivité, ce qui rend digne de se comporter correctement, parce-qu' il comprend un plus grand profit, tandis que les incitations métriques visent une couche intermédiaire du système, qui sont les recommandations, ne faisant pas digne que la plupart du temps a être bien comportés, parce-que pas un grand avantage est gagné. En ce qui concerne les mesures d'amorçage et les temps d'épuisement de données, grâce à la simulation nous avons obtenu des résultats qui sont largement dans des limites acceptables étant donné que la base d'utilisateurs totale pour le cadre du monde est assez grand, tout en utilisant des chaînes de FOAF qui permettent aux points de confiance d' être transférés ou prêtés d' une entité à une autre tout au long de la chaîne, fournissant ainsi de fortes incitations à la coopération et rassurant l'efficacité de notre système. Enfin, grâce à des tests réels avec un prototype Android, nous avons évalué la faisabilité d'un tel cadre avec l'obtention de la moyenne des temps de configuration de la connexion qui sont acceptables la plupart du temps et des temps de connexion qui sont assez longs pour rendre notre système utilisable.

Dans l'ensemble, nous pensons que notre système, sa mise en œuvre par la simulation et ses essais de prototypes ont obtenu des résultats satisfaisants et ont prouvé hors de tout doute raisonnable que nous avons amélioré l'état actuel de la littérature examinée dans les domaines connexes de cette thèse.

Abstract

According to the International Telecommunication Union (ITU), the number of subscribers using mobile broadband Internet services has raised from 268 million in 2007 to an impressive 2.1 billion users in 2013, accounting for more than the 50% of the world's Internet usage. This previous fact and the emergence and fast growth of ubiquitous computing enabling mobile social networking, contextual anytime anywhere user generated content, augmented reality or even augmented human, has fuelled the users' needs for permanent connectivity wherever they are and under all circumstances.

Wireless networks today are partially being formed by nodes (e.g. Internet access points, smartphones, femtocells) that are owned and carried by humans. As such, these User-centric Network architectures (UCNs) are giving rise to new Internet architectures, where broadband access is complemented by e.g. Wireless Fidelity (Wi-Fi) clouds, having a strong involvement of the Internet end-user. This represents a paradigm shift in the Internet evolution, as the user may be in control of parts of the network, in a way that is acknowledged (or not) by Internet stakeholders. In such scenarios where several strangers are expected to interact for the sake of robust data transmission, trust and cooperation incentives are of vital importance as these establish a way for the nodes involved in the system to communicate with each other in a safe manner, to share services and information, and above all, to form communities that assist in sustaining robust connectivity models.

While trust management systems can be used in isolation in order to provide robustness to a given architecture, cooperation incentives can be used to complement and collaborate with trust management systems as users can benefit from them while using the system, thus encouraging user's good behaviour. All in all, the relation between trust and cooperation incentives can greatly help into building a really solid and reliable architecture, being this the main purpose and scope of this thesis.

In this thesis we have designed, developed and implemented a fully decentralized trust management and cooperation incentives framework for user-centric network environments and we have evaluated it through simulation and through testing of a real prototype.

Our proposed framework is composed by three main components, the Identity Manager, the Trust Manager and the Cooperation Manager. The Identity Manager should be able to deal with multiple crypto-ids per user. The Trust Manager is composed by a dispositional trust adaptation module and by a proven Sybil attack resistant trust metric taken from the work of Seigneur [1] and, finally, the Cooperation Manager should provide the right incentives such as points or a virtual currency, which can be exchanged for services and can be gained when providing them, to help trust management to achieve an overall more secure and reliable architecture.

Through simulation we have proved that dispositional trust adaptation can improve the avoidance of malicious nodes as much as a 60%-70% in very hostile environments, which are defined as environments where the vastly majority of the nodes are malicious either by not providing the promised service

or by carrying out attacks such as Sybil attacks, whitewashing, self-promoting and the like. Concerning the cooperation incentives schema we have demonstrated through simulation that our incentive schema improves malicious conversions in near a 40% more of the cases than the compared metric, which is Credible Recommendations by Koutrouli et al. [2]. This improvement comes due to the fact that our incentives are focusing on the uppermost level of the system's functionality, which is sharing and getting connectivity, making it worthy to behave correctly as it comprises a bigger benefit, while the compared metric's incentives are targeting a mid-layer of the system, which are the recommendations, not making it worthy as much of the time to be well behaved, as not that big benefit is gained from it. Regarding bootstrapping measurements and data depletion times, through simulation we have obtained results that are well inside acceptable ranges given that the total user base for the framework in the world is big enough while using FOAF chains, which allow trust points to be transferred or lent from one entity to another along the chain thus providing strong incentives for cooperation, reassuring the effectiveness of our system. Finally, through real testing with an Android prototype we have assessed the feasibility of such a framework, obtaining average connection setup times, which are acceptable most of the times, and connection times, which are long enough to make our framework usable.

All in all, we think that our framework, its implementation through simulation and its prototype testing have achieved satisfactory results and have proven beyond a reasonable doubt that we have improved the current state of the reviewed literature for the related fields of this thesis.

Acknowledgements

First and foremost, I would like to express my most sincere gratitude to my mother, my father, and my brother for their encouragement and support during these amazing years doing my Ph.D. at the University of Geneva. They are the main reason of my success and I will never be able to thank them enough.

I also would like to thank the rest of my family, grandmother, cousins, aunts, uncles…without them and their support I would not be where I am today.

My most sincere thanks also go to my supervisor Dr. Jean-Marc Seigneur for offering me an excellent guidance and supporting me throughout all these thesis years. Without his help I would have never reached the end of my adventure, and I am very grateful to him.

I would like to also specially mention my thesis co-supervisor Prof. Dimitri Konstantas for his guidance, understanding and interesting discussions during all these past years.

Besides my supervisors, I would like to thank the rest of my thesis committee: Prof. Giovanna Di Marzo Serugendo, Prof. Sviatoslav Voloshynovskiy and Prof. Yanjun Zuo for their encouragement, time, and insightful comments.

Also a big thank you to all the people working in Battelle, you have made my life easier and better.

Last but not least, I want to sincerely thank all my friends from Valencia, Geneva, and all around the world for all their support, all the adventures, all the great moments shared together, all the parties and all the fun we have had in the last years, and for sure that we will have in the years to come. You guys are the best and I couldn't ask for better friends than you!

Thank you all!

Carlos Ballester Lafuente

Publications Related to this Ph.D.

As First Author:

1. Carlos Ballester Lafuente, Jean-Marc Seigneur, “Extending Trust Management with Cooperation Incentives: Achieving Collaborative Wi-Fi Sharing Using Trust Transfer to Stimulate Cooperative Behaviours”, 8th IFIP WG 11.11 International Conference on Trust Management (IFIPTM), 7th-10th July 2014, Singapore.
2. Carlos Ballester Lafuente, Jean-Marc Seigneur, Rute Sofia, Waldir Moreira, Christian Silva, “Trust Management in ULOOP”, Springer Lecture Notes in Social Networks: Book Chapter, ISBN 978-3-319-05217-5, May 31 2014, Springer.
3. Carlos Ballester Lafuente, Jean-Marc Seigneur, Thibaud Lyon, "Collaborative Wireless Access to Mitigate Roaming Costs", IARIA ICNS, 22nd-25th April 2014, Chamonix, France
4. Carlos Ballester Lafuente, Jean-Marc Seigneur, "Achieving Collaborative Wi-Fi Sharing without Changing Current Technologies", IEEE TrustCom - TrustID Symposium, 16th-18th July 2013, Melbourne, Australia.
5. Carlos Ballester Lafuente, Jean-Marc Seigneur, Paolo Di Francesco, Valentin Moreno, Rute C. Sofia, Waldir Moreira, Alessandro Bogliolo, Nuno Martins, "A User-centric Approach to Trust Management in Wi-Fi Networks", IEEE InfoCom – Demo Track, 15th-19th April 2013, Turin, Italy.
6. Carlos Ballester Lafuente, Jean-Marc Seigneur, "Dispositional Trust Self-Adaptation in User-Centric Environments", IEEE AINA, 26th-28th March 2013, Barcelona, Spain.
7. Carlos Ballester Lafuente, Jean-Marc Seigneur, Waldir Moreira, Paulo Mendes, Linas Maknavicius, Alessandro Bogliolo and Paolo di Francesco, "Trust and Cooperation Incentives for Wireless User-Centric Environments", IADIS e-Society, 10th-13th March 2012, Berlin, Germany.
8. Carlos Ballester Lafuente, Jean-Marc Seigneur, "Crowd Augmented Wireless Access", IEEE Augmented Human, 08th-09th March 2012, Megeve, France.
9. Carlos Ballester Lafuente, Xavier Titi and Jean-Marc Seigneur, "Flexible Communication: A Secure and Trust-Based Free Wi-Fi Password Sharing Service", IEEE TrustCom - UbiSafe Symposium, 2011, Changsha, China.

As Co-Author:

1. Alessandro Aldini, Alessandro Bogliolo, Carlos Ballester Lafuente, Jean-Marc Seigneur, “On the Trade-off among Trust, Privacy, and Cost in Incentive-Based Networks”, 8th IFIP WG 11.11 International Conference on Trust Management (IFIPTM), 7th-10th July 2014, Singapore.

2. Jean-Marc Seigneur, Carlos Ballester Lafuente, Alfredo Matos, "Secure user-friendly Wi-Fi access point joining," *Wireless Communications and Networking Conference (WCNC), 2013 IEEE* , vol., no., pp.4718,4723, 7-10 April 2013.
3. Alessandro Bogliolo, Paolo Polidori, Alessandro Aldini, Waldir Moreira, Paulo Mendes, Mursel Yildiz, Carlos Ballester Lafuente, Jean-Marc Seigneur, "Virtual currency and reputation-based cooperation incentives in user-centric networks," *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International* , vol., no., pp.895,900, 27-31 Aug. 2012
4. Xavier Titi, Carlos Ballester Lafuente and Jean-Marc Seigneur, "Trust and Reputation Management for Detecting Untrustworthy Access Points", International Conference on Reputation - Society, Economy, Trust. Montpellier, France, 19 September 2011.
5. Xavier Titi, Carlos Ballester Lafuente and Jean-Marc Seigneur, "Boosting Trustworthy Hotspot QoE Rating with Implicit Hotspot QoS Evidence", IADIS International Conference e-Society 2011, Avila, Spain 10-13 March.
6. Xavier Titi, Carlos Ballester Lafuente and Jean-Marc Seigneur "Trust Management for Selecting Trustworthy Access Points", International Journal of Computer Science Issues (IJCSI), Volume 8, Issue 2, March 2011.

List of Figures

List of Tables

List of Formulas

Acronyms

Acronym	Full text
2/3/4G	Second/Third/Fourth Generation (Mobile Networks)
AODVM	Ad-Hoc On-Demand Distance Vector Multipath
AP	Access Point
BFS	Bread First-Search
CCS	Command and Control System
CPU	Central Processing Unit
DHT	Distributed Hash Table
DoS	Denial of Service
DSR	Dynamic Source Routing
D_T	Dispositional Trust
FOAF	Friend Of A Friend
GUI	Graphical User Interface
HSDPA	High-Speed Downlink Packet Access
IDS	Intrusion Detection System
ISP	Internet Service Provider
Kbps	Kilobits per second
Mbps	Megabits per second
MCPS	Mobile Cyber Physical Systems
MD5	Message-Digest Algorithm 5
MP-DSR	Multi-path Dynamic Source Routing
OS	Operative System
P2P	Peer-to-Peer
QoE	Quality of Experience
QoS	Quality of Service
SDK	Software Development Kit
SHA-(1/2/3)	Secure Hash Algorithm
SLA	Service Level Agreement
SMR	Split Multipath Routing
SoA	State of the Art
TRS	Trust and Reputation Systems
T_T	Trust Threshold
TTP	Trusted Third Party

UCN	User-Centric Network
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network

Chapter 1. Introduction

According to the International Telecommunication Union (ITU) [3], the number of subscribers using mobile broadband Internet services has raised from 268 million in 2007 to more than 2.1 billion users in 2014, accounting for more than the 50% of the world's Internet usage. This previous fact and the emergence and fast growth of ubiquitous computing [4] enabling mobile social networking, contextual anytime anywhere user generated content, augmented reality or even augmented human [5], has fuelled the users' needs for permanent connectivity wherever they are and under all circumstances.

The flexibility of wireless technologies is allowing the Internet to expand in a user-centric way, which is particularly relevant when considering that wireless technologies such as Wireless Fidelity (Wi-Fi) currently complement Internet access and can effectively be seen as the last hop to the end-user. This fact becomes even more relevant taking into account the vast amount of Wi-Fi access points that can be found nowadays in urban environments. In these scenarios, the end-user is empowered by becoming a key and active element of the connectivity model.

In the following sections we will detail the main context and motivation of this Thesis, its purpose and scope and the research questions that originated it as well as its main contributions to this field.

1.1 Context and Motivation

Wireless networks today are partially being formed by nodes (e.g. Internet access points, smartphones, femtocells) that are owned and carried by humans. As such, these User-centric Network architectures (UCNs) are giving rise to new Internet architectures, where broadband access is complemented by e.g. Wireless Fidelity (Wi-Fi) clouds, having a strong involvement of the Internet end-user. Such User-Centric Network architecture can be seen in **Figure 1**.

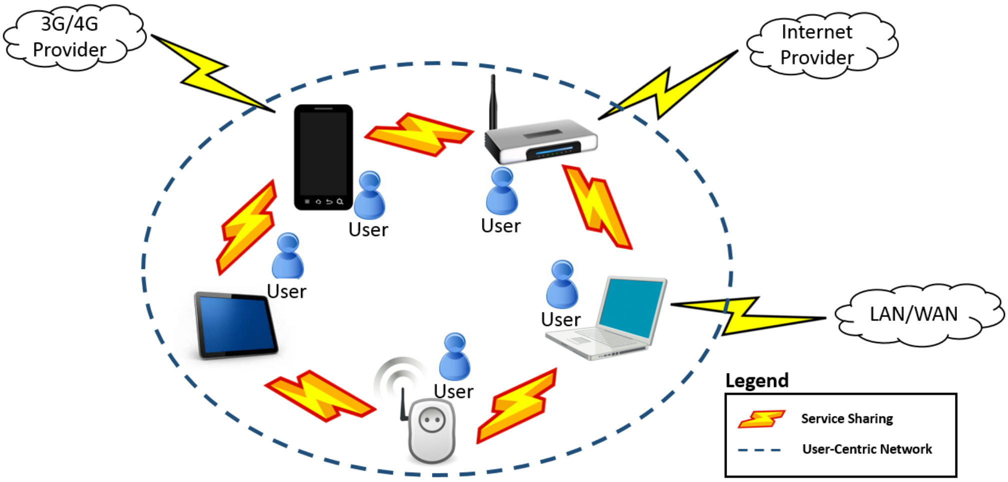


Figure 1. User-Centric Network Architecture

This represents a paradigm shift in the Internet evolution, as the user may be in control of parts of the network, in a way that is acknowledged (or not) by Internet stakeholders. In such scenarios where several strangers are expected to interact for the sake of robust data transmission, trust and cooperation incentives are of vital importance as these establish a way for the nodes involved in the system to communicate with each other in a safe manner, to share services and information, and above all, to form communities that assist in sustaining robust connectivity models.

UCNs have a few characteristics which make them complex to control: i) they are supported both by static, fully dedicated nodes as well as by nodes provided by end-users on-the-fly; ii) in their majority, they are complementary to broadband access but are located on the last hop to the Internet end-user, which is not always accessible from the operator perspective; iii) as some nodes are carried by Internet end-users, their networking composition and organization follows a social behaviour, highly affected by human movement features. UCNs are therefore in the category of self-organizing systems, where often anonymous nodes can try to become part of the system increasing the risk of Sybil [6] and other [7] attacks, and as such, UCNs require that trust and reputation systems/metrics should be attack resistant in order to ensure that the architecture is as robust as possible.

The aim of this thesis is to build a framework that will combine a fully decentralized trust management system with cooperation incentives, in order to persuade users to behave in a correct manner to create a sustainable and scalable systems in exchange of some form of compensation/remuneration, in order to achieve an overall more robust architecture.

1.2 Purpose and Scope

While trust management systems can be used in isolation in order to provide robustness to a given architecture, cooperation incentives can be used to complement and strengthen trust management systems as users can benefit from them while using the system, thus encouraging user’s good behaviour.

By providing cooperation incentives, there are economic dynamics involved, encouraging the users to keep using the system in a rightful way as they benefit from it. Incentives provide the user with adequate network resources or useful information, based on the node’s interaction to its communities and to/from external communities. This in turn, encourages the user to earn a good reputation level, as other users are more likely to interact with high-reputation users than low-reputation ones, reinforcing the trust and reputation system. In particular, reputation acts as an incentive as long as it represents an enabling condition for taking part in some kind of community, for providing some kind of service, or for taking advantage of some premium services. Reputation cannot be traded for credits or money, but a trusted community node is more likely to be involved in a remunerative task, and in the other hand, the remuneration required by a node to provide some kind of service/resource affects the loss of reputation of that node in case of poor quality of the service/resource provided. Also, earning a high reputation can be seen as an incentive itself, as due to social dynamics, being able to display a high reputation level can be seen as a sign of higher “social status”.

The lack of trust between users can influence their level of willingness. Thus, another aspect to increase the willingness level in this case is user interest, which must be considered in cooperation. Users sharing the same interest, although being completely unknown to one another, can be easily encouraged in carrying information on behalf of others. At this point, cooperation not only helps users disseminate information quickly but it also contributes to sparing resources from users who are not interested in that specific content. Cooperation shall be easily encouraged if users share some social relationship. Thus, social ties have an important role in making cooperation among users even more reliable.

All in all, the relation between trust and cooperation incentives can greatly help into building a really solid and reliable architecture and it is a topic that should be subject to further research, thus the purpose and scope of this thesis.

The proposed Trust Management and Cooperation Incentives framework for fully decentralized user-centric network environments is composed by three main components, the Identity Manager, the Trust Manager and the Cooperation Manager. The Identity Manager should be able to deal with multiple crypto-ids per user, as we assume it is not possible to control users not having more than one virtual identity, and also it should be able to link different crypto-ids pertaining to a same user in order to be able to fill existing trust gaps if needed as presented in Seigneur’s PhD thesis [1]. A crypto-id is namely the public key of a user, or the hash of that public key.

The Trust Manager is composed by a dispositional trust adaptation module and by a Sybil attack resistant trust metric. *Dispositional Trust* (D_T) is defined as the general willingness of a given user to trust others if they have not interacted together before.This first module is in charge of adapting dispositional trust according to the context (amount of malicious interactions, the surrounding environment, etc…). The second module is the Sybil attack resistant trust metric, which is needed because as previously mentioned each user can have multiple virtual identities and thus Sybil attacks in this kind of scenario is of high concern. Finally, the Cooperation Manager should provide the right incentives to help trust management to achieve an overall more secure and reliable architecture. It should provide technical incentives, such as bandwidth, and support a wireless UCN economy in the form of points or a virtual currency, which can be exchanged for services and can be gained when providing them within the agreed Service Level Agreement (SLA) or other previously agreed parameters.

Trust has been already widely used, but not many times has been used in combination with a cooperation manager and cooperation incentives. In this thesis, we are going to use as starting point a well-known Sybil resistant metric called Trust Transfer by Seigneur [1], in order to analyse how cooperation incentives in the long term can influence such trust metric. The idea is to apply trust transfer as initial metric in the framework and to see how it evolves in the long term in combination with cooperation incentives.

Mainly the framework is focused on wireless environments, more concretely wireless UCN environments, but it might as well be applied to other environments if possible.

1.3 Projects that Contributed to this Thesis

There have been several European Framework Seven [8] projects that have contributed to the different parts of this thesis. We detail them briefly in the next subsections and we explain for each of them in which way they have contributed to this thesis.

1.3.1 ULOOP

ULOOP [9], [10] stands for "*User-centric Wireless Local Loop*". The project tries to bring in a new approach to user-centricity by exploring user-provided networking aspects in a way that expands the reach of a multi-access backbone. ULOOP main expected results are user-centric open-source software and a large-scale realistic demonstrator.

The main contributions of ULOOP to this thesis are in the areas related with cooperation incentives and virtual currency.

1.3.2 TEFIS

TEFIS [11] stands for "*Testbed for Future Internet Services*" and it aims to support Future Internet of Services Research by offering a single access point to different testing and experimental facilities for communities of software and business developers to test, experiment, and collaboratively elaborate knowledge. The project is developing an open platform to access heterogeneous and complementary experimental facilities addressing the full development lifecycle of innovative services with the appropriate tools and testing methodologies.

The main contributions of TEFIS to this thesis are in the field of decentralized networks and service sharing.

1.3.3 MUSES

MUSES [12] stands for "*Multiplatform Usable Endpoint Security*" and its overall purpose is to foster corporate security by reducing the risks introduced by user behaviour. Nowadays, information is highly distributed amongst corporate servers, the cloud and multiple personal devices like PDAs, tablets and smart phones. These are not only information holders but also user interfaces to access corporate information. Besides, the Bring Your Own Device (BYOD) practice is becoming more common in large organisations, posing new security threats and blurring the limits between corporate and personal use.

MUSES plans to design and develop a system for corporate security that goes beyond existing endpoint protection platforms (EPPs), which have only been thought from the corporate point of view and with the operators of security centres in mind.

The main contributions of MUSES to this thesis are in the domain of survivability (bootstrapping times, data depletion) and decentralized service sharing.

1.4 Research Questions and Contributions

As previously stated, in this thesis we intend to study how trust management and cooperation incentives can work together in a fully decentralized way in user-centric network environments. We aim for ***decentralized*** user-centric network environments as we think they are the natural evolution for the more traditional centralized networks and that this approach will be the trend for network architectures in the near future.

From this starting point, there are many questions stemming which need to be answered and which motivate this thesis, being the most prominent the following ones:

- *Can current trust management and cooperation incentives frameworks/metrics be applied in a fully decentralized manner in user-centric wireless network environments?*
- *Is there any framework/metric for user-centric wireless network environments which is fully Sybil attack resistant?*
- *Can computational trust management be used to empower cooperation incentives among those users participating in the UCN?*

Which leads us to the key question at the core of this thesis:

Can trust management and cooperation incentives be coupled into a unique fully decentralized framework to improve user-centric wireless network environments?

Within the context of this thesis we have designed, implemented and evaluated through simulation our framework, which combines a fully Sybil resistant computational trust metric with cooperation incentives into a fully decentralized framework for user-centric network environments.

The contributions of this thesis are the following:

- A new state of the art dispositional trust adaptation algorithm, which contributes to the attack resistance of the trust metric of choice and which has been evaluated through simulations [13].
- The integration of a trust metric, which already has proven to be resistant to Sybil attacks taken from the work of Seigneur [1], enhanced with cooperation incentives in form of points [14], [15].
- A probabilistic study on chains of trust and friend of a friend (FOAF) chains in small-world network subsets [14] in order to further improve the effectiveness of cooperation incentives in the form of points and their possible propagation through the user-centric network.

- A combination of all of the above contributions in the form of a fully decentralized framework, using a trust metric proven to be resistant to Sybil attacks [1] and which can deal with multiple identities per user.
- A large scale simulation of the framework using AnyLogic on real life environments (e.g., airport, ski resort, etc.) [14], [16]
    - proving its feasibility and performance from:
        - Bootstrapping point of view
        - Resource depletion point of view
    - comparing the Trust Transfer [1] trust metric combined with a cooperation incentive schema against other related trust metric using AnyLogic [17] simulation tool, which has never been done before for this metric, taking into account:
        - Number of users connected to selfish or malicious APs
        - Number of potential malicious APs vs. those who actually turn to be selfish regardless of the offered incentives to behave correctly.
- A real life experimental study using a prototype implementation [18] in order to obtain accurate and meaningful results on the performance of such a framework and system.

1.5 Thesis Outline

After this introductory Chapter 1, the rest of this thesis, is organized as follows. In Chapter 2 we introduce the background knowledge needed to better understand the rest of this thesis. Following, in Chapter 3 we present the current State of the Art (SoA) regarding Trust Management, Cooperation Incentives and Fully Decentralized Sustainable Wireless Networks. Afterwards, in Chapter 4 we describe the framework model and design, both from a formal definition and from an architectural point of view. Chapter 5 presents the implementation and evaluation of our framework and finally, in Chapter 6, we discuss those results and the framework strengths and weaknesses concluding this thesis and summarizing its main contributions and possible future work to be done.

Chapter 2. Background Theory

This section introduces the main theoretical definitions and knowledge related to trust management, fully decentralized user-centric environments and cooperation incentives. The aim of this chapter is to give a brief overview of the concepts and to give the reader the background required to better understand the following parts of this thesis.

2.1 Trust Management Aspects

Trust management and reputation are key points when it comes to fully decentralized and user-centric environments, given that they are undergoing constant changes both in terms of composition (i.e. users might enter or leave the system at any given time) and it is frequent to encounter newcomers or devices with which there has been no previous interaction. Trust and reputation are linked but can act separately and they come into play in order to proactively enforce security and safety, both in an automated and manual fashion, in order for the participants to be able to take the best decisions. There are several definitions for trust and reputation depending on the field and the scope in which they are applied. Also, these systems are prone to several specific attacks that rely on the very foundations on which trust and reputation are built and in the way they work and should be taken into account.

2.1.1 Human Trust and Computational Trust

In order to better understand this thesis, the different existing kinds of trust and their formal definitions should be introduced. In the human world, trust exists between two interacting entities and is very useful when there is uncertainty in the result of the interaction. The requested entity uses the level of trust in the requesting entity as a mean to cope with uncertainty, to engage in an action in spite of the risk of a harmful outcome. There are many definitions of the human notion trust in a wide range of domains, with different approaches and methodologies: sociology, psychology, economics, pedagogy, etc. These definitions may even change when the application domain changes. However, it has been convincingly argued that these divergent trust definitions can fit together as stated in McKnight et al. [19]. Romano's [20] recent definition tries to encompass the previous work in all these domains as follows:

"Trust is a subjective assessment of another's influence in terms of the extent of one's perceptions about the quality and significance of another's impact over one's outcomes in a given situation, such that one's expectation of, openness to, and inclination toward such influence provide a sense of control over the potential outcomes of the situation."

Computational Trust applies the human notion of trust to the digital world, which normally is seen as malicious rather than cooperative.

A computational model of trust was first proposed by Marsh [21]. In social research there are three main types of trust: interpersonal trust, based on past interactions with the trustee; dispositional trust, which accounts for trustor's general disposition towards trust, independently of the trustee; and system trust, provided by external means such as insurance or law. Computational trust systems can be either centralized or decentralized. In centralized mechanisms, ratings from direct interactions are provided to a central node, which acts as a trusted authority who stores the computed trust values and derives a global reputation metric for each of the participants in the system, making it publicly available. In decentralized mechanisms there are no central authorities to collect ratings or compute reputation levels, so that ratings are either stored in distributed storage systems or privately kept by each participating node as a result of its direct experience. Trust values are then individually computed at each node based only on the previous experience of that particular node, on the data possibly available from distributed storage systems, and on the recommendations coming from other nodes.

2.1.2 Dispositional Trust

Dispositional Trust (D_T) is defined as the general willingness of a given user to trust others if they have not interacted together before. As such, as a first step the owner of the node will set up this value manually. The D_T setup is done in the boot-up phase and it may or may not remain the same during the node's lifetime. However, as it might provide a better protection for the user, depending on the surrounding environment of the node and other external factors, an adaptation process may be carried out to readjust D_T automatically.

2.1.3 Reputation and Recommendations

Tightly related to trust is the concept of reputation, which Abdul-Rahman et al. [22] define as follows:

> ***"Reputation is an expectation about an individual's behaviour based on information about or observations of its past behaviour."***

In online environments, where a given entity usually has less information to be able to compute the trustworthiness of other entities, their reputation information is typically used to calculate their trust level. An entity that has a higher reputation value is normally considered to be more trustworthy. Reputation can be computed in several ways:

- An entity may either rely on its direct observations
- It might rely on the experiences of other entities, which are called recommendations
- It might use a combination of both to determine the reputation of another entity.

2.1.4 Sybil and Other Attacks

A very well-known identity multiplicity attack in the field of computational trust due to collaboration is Douceur's Sybil attack [6]. Douceur argues that in large scale networks where a centralised identity authority cannot be used to control the creation of virtual identities, a powerful real-world entity may create as many virtual identities as it wishes and in doing so challenges the use of a majority vote and flaw trust metrics. This is especially important in scenarios where the possibility to use many pseudonyms is facilitated and provided by the trust engine. In fact, a sole real-world entity can create many pseudonyms that blindly recommend one of these pseudonyms in order to fool the trust engine. The level of trust in the latter virtual identity increases and eventually passes above a threshold, which makes the decision to trust (the semantics of this depend on the application). Additional categorization of attacks, as described in Hoffman et al. [7], include:

- **Self-Promoting** - Attackers manipulate their own reputation by falsely increasing it.
- **Whitewashing** - Attackers avoid the consequence of abusing the system by using some vulnerability to repair their reputation. Once they restore their reputation, the attackers can continue with their malicious behaviour.
- **Slandering** - Attackers manipulate the reputation of other nodes by reporting false data to lower the reputation of the victim's nodes.
- **Orchestrated** - Attackers orchestrate their efforts and employ several of the above strategies.
- **Denial of Service (DoS)** - Attackers cause denial of service by preventing the calculation and dissemination of reputation values.
- **Impersonation** - Impersonation refers to the situation where one user/node tries, and actually succeeds, into adopting the identity of another user/node. This should be seen as a serious threat for trust and reputation systems, as the attacker who succeeds in doing so can exploit then his victim's reputation and already established trust relationships to his/her own benefit. By making use of public key cryptography, messages between nodes can be signed, and the node receiving these messages can then check the signature, or the absence of it, in order to validate them. As only the original node should have the private key, which is able to generate such a signature, the receiver is able to detect an attempt of impersonation [23],[24].

2.2 Cooperation Incentives Aspects

The "Tragedy of the Commons" [25] states that it is not possible to avoid the depletion of a shared resource by individuals, acting independently and rationally according to each one's self-interest. Despite their understanding that depleting the common resource is contrary to the group's long-term best interests they will still deplete it, as this behaviour is inherent to the human nature. Even though the tragedy of the commons was first applied to mainly economic and sociology fields, it can be extrapolated to P2P and other sharing services as can be seen in [26] and [27]. Without a strong incentive being present, there is no real reason for users to share back as much at least as they got available when some other user shared, as it is in the very human nature to be self-interested agents, thus acting exclusively for their own benefit. This lack of incentives will ultimately render the service unusable, as there will be no resources to share, but many users willing to use shared resources. For example, in a system where the main service shared is connectivity, if there are no incentives for the users to share their own data access the result would be that most of the users would try to access the service in order to get a connection without ever sharing their own, resulting in a fast data depletion for the users that would be sharing their access, thus quickly rendering the system unusable. In the other hand, if there would be incentives as for example points gained when sharing that can be afterwards used to get access through other users, the participants in the system would be much more prone to share their own access when possible as that would be the only effective mean to get something in return.

Cooperation incentives can complement trust and reputation systems by persuading users to behave in a correct way in exchange of some form of compensation/remuneration and by encouraging cooperation among them. This mechanism aims at providing incentives for users to share their available resources and even effort. The main encouraging aspect to lead users to cooperate is to provide them with resources and help (i.e., efforts) whenever he/she most needs them. This aspect is fully supported by the fact that cooperation incentive mechanism should employ rules to differentiate good users (those who cooperate whenever possible) from bad users (those who only consume others' resources and may have low trust and reputation levels). This differentiation will surely make the non-cooperative behaviour unattractive in the system and cooperating becomes a must as everybody wins. While this does not avoid the tragedy of the commons to a full extent, it greatly diminishes its impact as greedy behaviour provides no gain and it is mostly a downside in the long run.

2.2.1 Rewarding Good behaviour

The success of any cooperative infrastructure depends on the willingness of users to share resources, which represents a key factor for the success of UCNs.

By providing cooperation incentives, there are rewards involved that encourage the users to keep using the system in a rightful way as they benefit from it. Incentives can provide the user with adequate network resources or useful information based on the node’s interaction with its peers and user communities. This in turn, encourages the user to earn a good trust level, as other users are more likely to interact with highly trusted users rather than untrusted ones as the former are more likely to provide a given service and do so up to a certain service level agreement.

2.2.2 Points and Other Rewards

Whenever intrinsic motivation is not enough, cooperation incentives are required to provide additional extrinsic motivation, which can be viewed as a reward capable of realigning individuals' utility to public utility. There are several forms of rewards that can be considered, amongst which we can find *reputation*, which enhances the status of a cooperative individual in a group; *reciprocity*, describing the evolution of cooperative behaviour, which may be influenced by the probability of future mutual interactions and *virtual currencies or points*, which entails the quantification of the value of cooperative decisions.

Reputation can act as an incentive in UCNs as long as it represents an enabling condition for taking part in some kind of group activity, for providing some kind of service, or for taking advantage of some others. Although reputation cannot be traded for money, a trusted community node is more likely to be involved in remunerative tasks. On the other hand, associating tangible effects to reputation can motivate community members to take part in the trust-management system and provide their feedback.

Virtual currency or points are often adopted in online communities both to support monetization, and to provide a guarantee of reciprocity (reducing the risk of betrayal which could keep people from taking pro-social decisions). Virtual currency was first introduced in online games and social networks as a mean to buy and sell virtual goods without making use of real money, thus avoiding security issues, taxation, and mistrust. Virtual currency systems are subject to many requirements (including transferability, anonymity, usability, and scalability) and exposed to many issues (including forgery, double spending, cheating, and speculation) which make them hard to implement.

Existing solutions can be broadly classified into centralized and decentralized systems. Centralized systems rely on a trusted third party (TTP) which acts as a bank. In online systems, the TTP is directly involved in any transaction and it provides a real-time guarantee to the merchant. In offline systems, on the contrary, the merchant accepts payments from the user without interacting with the bank and deposits them later on. Offline systems work as long as the merchant is guaranteed that the TTP will either accept the deposit or be able to identify and punish the cheating user. Decentralized systems, on the contrary, do without centralized TTP and rely on peers to protect merchants from cheating. Regardless of their nature, all virtual currency systems rely to some extent on cryptography, which is used to avoid forgery, to identify cheating users, to guarantee anonymity, or to enable traceability. In the last decade virtual currency has been often adopted as a cooperation incentive framework in ad-hoc networks [28]. Decentralized and offline virtual currency systems are the most suited to be applied to UCNs and peer-to-peer networks. Bitcoin [29], Nuglets [28], and WhoPay [30] are representative examples.

2.3 Fully Decentralized User-Centric Environments Aspects

The fact that network connectivity is being shifted towards the end-users' devices, coupled with the strong predominance nowadays of smart mobile devices owned by those users, introduces several new issues that should be addressed. Given the resource limited nature of those devices, there are several constraints that should be addressed in order to guarantee the survivability of such fully decentralized user-centric networks.

Following, we list the main resource limitations that affect such devices and we detail the constraints and downfalls linked to them.

2.3.1 Battery Depletion

Opposite to non-portable or semi-portable (i.e. laptops) devices, which are usually connected to a power source, mobile devices often rely on their battery life in order to carry out their basic functions. This imposes a constraint on such devices, as battery is a limited resource and certain actions can deplete it faster. In mobility situations which usually are highly dynamic and occur in very different kinds of scenarios, it might be difficult for the owner of those devices to find a power source where to recharge the battery or it might be inconvenient as this often implies becoming stationary for the time of recharging, thus breaking the purpose of mobility itself.

2.3.2 Memory and Storage Space Limitations

Even though technology advances at a fast pace, and storage is becoming less of a problem nowadays, it is important to acknowledge that mobile devices still have limited physical storage capabilities and that memory is somewhat limited too. This might not be a problem or issue for a regular daily use, and thus it is not of a concern for this thesis, but it is still remarkable to take into account that there might be situations in which such devices might run short on these resources, becoming effectively a limiting factor.

2.3.3 Data Depletion

Mobile data limits and monthly quotas are a big concern for users and effectively one of the most restraining limitations present in current mobile devices. Roaming costs incurred by users when operating their smartphones in another country and also extra costs derived from going over a certain monthly data allowance for local users might deter those from using any application, accessing data or providing services such as Wi-Fi sharing when on that situations. In this thesis we have focused mainly on studying this limitation and not battery or storage limitations, as this is the most relevant constraint related to our framework given that it is based on the sharing of connectivity in between the agents that integrate the network.

In the next two sections we dive into this issues and present more insight providing some real data obtained from French operators' price plans.

Mobile Data Limits

In order to justify our claims, we have made a thorough compilation of the costs of monthly subscription prices for most of the French based operators in the market.

Table 1 presents for each of those operators their monthly data allowance, what measures are taken when the user goes over the monthly data allowance and if there are other particular conditions linked to the subscription.

Table 1. French providers price comparison.

Operator	Monthly Allowance	Going over the monthly allowance	Other Conditions
SFR	2 GB	Stopped until recharged. 2 € each 100 MB recharge	-
Prixtel	3 GB / 20€	0.10 € per extra MB consumed	-
Sosh	1 GB	Bandwidth is lowered, no extra cost	P2P forbidden
Bouygues Telecom	Different offers	Bandwidth is lowered, no extra cost	0.2 € per MB in prepaid
Orange	Different offers	Bandwidth is lowered, no extra cost	-
La Poste Mobile	Different offers	Bandwidth is lowered, no extra cost	Portable modem, VoIP, P2P forbidden
Coriolis	500 MB	Bandwidth is lowered, no extra cost	Portable modem forbidden
M6 Mobile	Different offers	Data access stopped until next month	P2P forbidden
Free	3 GB	Bandwidth is lowered, no extra cost	If lower subscription than 3 GB, then 0.05€ per extra Mb
B&You	3 GB	Bandwidth is lowered, no extra cost	If lower subscription than 3 GB, then 0.05€ per extra MB
E Leclerc	2€ each 30 MB	Data access stopped until next month	Without internet option, 0.39€ per MB. Portable modem, VoIP and P2P forbidden
Virgin Mobile	Different offers	Above 10€ subscription, lower bandwidth. Under 10€ subscription, 0.05€/MB above 100 MB	Portable modem, VoIP, P2P forbidden
Auchan Telecom	500 MB	Bandwidth is lowered, no extra cost	Streaming, VoIP, P2P and portable modem forbidden
Numericable	Different offers	Above 10€ subscription, lower bandwidth. Under 10€ subscription,	-

		0.4€/MB above 20 MB	
NRJ Mobile	Different offers	Bandwidth is lowered, no extra cost	VoIP, P2P forbidden

As can be seen in the previous table, even though some of the operators only lower the available bandwidth to the user when going over the monthly data allowance, in many cases this is only true for the higher priced subscriptions, while more modest ones usually incur in some type of extra cost per MB after the monthly allowance is surpassed. Moreover, many of the subscriptions either stop the data access until the next month or directly start charging a per-MB-fee after this limit has been passed without further warning. Also, it is noteworthy that many operators forbid by default the use of the smartphone as portable modem (commonly known as tethering or portable hotspot) and the use of P2P traffic.

High Roaming Costs

In addition, high roaming costs make the access of mobile data while abroad very expensive, and thus, impede users to access applications and other online sources normally, as the price they might pay in order to use these services would escalate very quickly. A recent study on international roaming costs [31] carried out by the OECD, sets the average price per MB when roaming in the EU/EEA area at an average of 2.60€.

Even though a recent European legislation procedure has been set to abolish roaming charges in the whole of the Eurozone as of 15th December 2015 [32], roaming charges will still apply vastly all over the rest of the world, making still highly expensive to achieve data connectivity though mobile devices all around the world.

2.3.4 Client-Side Authentication is Difficult

In our view, assuming worldwide user strong authentication that would ensure giving only one digital identity to any user is not realistic. So far, all initiatives to achieve it have not succeeded; a global PKI (Public Key Infrastructure) has been deemed not feasible, and social and federated logins, even though useful, cannot be tied properly to a real world identity and identities can be easily faked or spoofed.

That is why in order to guarantee the survivability of a fully decentralized user-centric network, trust management and cooperation incentives play a key role, as they provide the means for soft security without the need for strong authentication mechanisms and they also foster and enforce good behaviour on users.

Chapter 3. Related Work

There has been a lot of work and research already done in the fields of Trust Management and Reputation, Cooperation Incentives and Survivability of decentralized user-centric networks. The aim of this section is to present the most relevant work already done and related to this thesis in these fields, in order to give an exhaustive overview of the State of the Art (SoA).

This extensive review will allow us to identify the commonalities already addressed by existing frameworks, identify their shortcomings and pitfalls and identify the gap present in these fields in order to be able to better compare them to our own framework.

3.1 Related Work on Trust Management

This section introduces the related work already done on trust management. The main goal of the metrics and frameworks reviewed in this section is to help the user select the most trustworthy agent to interact with as well as some of them focusing on protecting the system from Sybil attacks.

In the work presented in the thesis of S. Ries [33], the author provides an approach based on trust management in order to improve the selection of reliable interaction partners. The main goal of his approach is to estimate the trustworthiness of a given entity with the highest possible accuracy to improve the average quality of the interactions with it. The trustworthiness of an agent or entity is derived from the evidence collected during past interactions. In order to achieve that, current Bayesian trust models are extended and improved regarding the following aspects:

- Better integration of the characteristics of the application context
- More intuitive access to the trust model
- Better integration of recommendations by third parties.

The last aspect is crucial as there are many scenarios where direct evidence between entities is unattainable or scarce. The proposed approach provides a solution involving the robust integration of recommendations provided by third parties. It considers possible attacks by entities providing in purpose misleading recommendations, either individually or collectively. The approach is validated through simulation, showing results over a set of 15 populations, which have been canonically derived from the system model, modelling entities with different typical behaviours. Furthermore, the results obtained by simulation regarding collaboration between agents in an opportunistic network prove that the model provides high accuracy about the estimation of an entity's trustworthiness and the average quality of interactions to find the best interaction partner.

Seigneur [1] studies an approach for trading privacy for trust based on the linkage of pseudonyms, as it has been proposed to mitigate the inherent conflict between trust and privacy while effectively protecting against Sybil attacks. In order to achieve this trade, the concept of *fusionym* is introduced, consisting on the calculation of a unique trust value supposed to reflect the overall trustworthiness brought by a set of linked pseudonyms, which are the different virtual identities belonging to a same entity. Their model allows for self-recommendations during the privacy/trust trade, but this introduces the possibility of Sybil attacks. Trust Transfer is then used to achieve safe fusionym while still protecting against Sybil attacks for positive recommendations, by moving some of the trustworthiness of the recommending entity to the trustworthiness of the trustee. Trust Transfer implements a trust engine with the following characteristics and components:

- A decision-making component, which is called when a requested entity has to decide which action should be taken due to a request made by another entity, the requesting entity.
- The Entity Recognition (ER) module, which deals with the virtual identities of the agents present in the system.
- The chosen action should maintain the appropriate cost/benefit ratio.

The basis for the model is that depending on the benefits that can be obtained by having a trust level over a certain threshold, people may be less reluctant to trade part of their privacy for increased trust. The choice of evidence in order to disclose privacy while increasing trustworthiness should be based on the following principle: *“no more evidence than needed should be linked”*. By allowing any entity to make *(self-)* recommendations, the authors implicitly support a change of identity, where existing evidence can be transferred and linked to the new identity through a recommendation. A list of pseudonyms owned by the requesting entity could be sent back as potential new recommenders but the overall trust level of the evidence provided by these entities would be discounted by the recommendation process. In the other hand, sometimes using fusionym can be more useful as no trust is discounted. Thanks to Trust Transfer, safe fusionym is possible even if self-recommendations are used before the disclosure of the link between the pseudonyms. At the same time, it removes the risk of Sybil attacks for positive recommendations, being this one of the major contributions to the field of trust management.

Martucci et al. [34], propose an identity management model which supports role-based pseudonyms and which can support the use of trust and reputation systems while still providing a reasonable amount of privacy protection and anonymity and at the same time avoiding Sybil attacks. Users’ privacy protection requires actions that cannot be linked one another, implying that an external observer is not able to link two actions or their outcomes belonging to the same user. The problem they face is that building up trust and reputation usually requires long-term identifiers that can be in fact linked over several transactions for a given agent or entity. To tackle this problem they propose an architecture to generate pseudonyms based on roles, which in turn are bound to a given set of services called a service context. Their proposal offers unique long-term identifiers which are the basis for trust and reputation systems, allowing to build behaviour histories about the other entities in the system, while still providing unlinkability between the actions performed by a given entity in different service contexts and detecting Sybil identifiers to avoid whitewashing, badmouthing and other Sybil related attacks. In order to achieve Sybil-free pseudonyms, they use a cryptographic construction which generates one self-signed pseudonym for each of the contexts the user has to interact with, all derived from an initial identifier provided by a trusted third-party on the bootstrap step (*hence, it is not a fully decentralized system*). Contexts are directly created by the service providers present in the system, and they contain a unique hashed value. By combining this hashed value with a newly auto-generated public key and their initial identifier, the agents can generate context-specific pseudonyms to interact with that given context. To conclude, they demonstrate that it is still possible to detect if a single agent has created more than one pseudonym for a unique context by means of cryptographic calculations, hence effectively avoiding the possibility of Sybil attacks while still preserving the privacy of the entity as actions carried out in different contexts cannot be linked.

For Ziegler et al. [35], besides understanding the information and relations in between entities, knowing about their credibility is equally important and crucial, and thus trust and trust metrics are needed to evaluate trust relationships between individuals. One of their main contributions to the field is an extensive trust metric classification dividing trust metrics into two big groups, namely a global one, which takes into account all peers and trust links connecting them in the whole system and a local one, which takes into account personal bias and compute a more personalized trust, further subdividing these categories into more specific ones. The second contribution the authors make in their work is ***Appleseed***, a trust metric designed for a Semantic Web scenario and which is based on spreading activation strategies. Appleseed works with partial trust graph information where nodes are queried only when needed, and where nodes make their manual trust values publicly available, thus posing a threat to privacy. Finally, the authors compare their trust metric with Advogato [36], and evaluate its attack resistance. Advogato is a trust metric that evaluates a set of peer certificates in order to be able to accept new user accounts, where the certificates are represented as a graph, with each account as a node and each certificate as an edge, in order to accept as many valid accounts as possible while reducing the impact of attackers.

The problem of emerging P2P applications and their requirement for decentralized access control is addressed in Ingram's work [37]. In his paper, he surveys the current work on overlay networks, trust and identity certification. Then, he focuses on the particular problem of distributing evidence to be used in trust-based decisions by presenting a system that solves this in a highly scalable way and that resists attacks such as false recommendations and collusion. A key feature of his model is the use of recommendations to exchange trust information between principals, which creates a requirement for an effective and scalable mechanism to distribute such information across the network where millions of principals can be present and most of whom do not know each other. Any entity in the system that can exchange trust information and perform actions is called a principal. They are identified by ids not linked to the real world to preserve anonymity, and those ids are cryptographically generated. The model takes into account premises such as that the patterns of interaction may be random and not exhibit much locality of reference and also that any information sent via the network can be falsified, including recommendations and routing information, yet the system must remain secure. The system provides a set of certification third-parties which need to sign the virtual identities in order to avoid Sybil by making it very expensive to obtain extra identities. The signing would only occur if the third-party entity has already had relation in the past with the requesting entity, such as banks, internet service providers (ISPs), etc. As there are only so many of these entities a user in the system can actually have interacted with, obtaining a meaningful amount of multiple valid ids under the control of a same user is deemed hard enough not to be a real risk. The system uses a peer-to-peer (P2P) distributed hash table (DHT) overlay in order to exchange information in between participants and to store this information in the form of profiles, and uses partly a trust engine derived from the SECURE project [38]. The system returns all that has been observed about a given principal and all trust values are pre-computed and do not require a process of convergence over multiple iterations. There is a distributed, shared evaluation of the accuracy of recommendations – called meta-trust by the author. It cannot be considered fully decentralized as it relies on third-party trusted certification authorities to ensure the uniqueness of virtual ids.

The work of Kamvar et al. [39], deals with the sharing and distribution of information in P2P environments, as experience shows that the anonymous, open nature of these networks offers an almost ideal environment for the spread of self-replicating inauthentic files and identities. The solution presented in this work consists on an algorithm to decrease the number of downloads of inauthentic files in a peer-to-peer file-sharing network, which assigns each peer a unique global trust value, based on the peer's history of uploads. They present a distributed and secure method to compute global trust values and by having peers use these global trust values to choose the peers from whom they download, the network effectively identifies malicious peers and isolates them from the network. In simulations, this reputation system, called *EigenTrust*, has been shown to significantly decrease the number of inauthentic files on the network, even under a variety of conditions where malicious peers cooperate in an attempt to deliberately subvert the system. With their solution, they address 5 points that every trust and reputation system should comply with when dealing with P2P networks:

1. The system should enforce policies by itself, meaning that it should come from the users and not from any central entity
2. The system should maintain anonymity, using virtual identities for the agents in the system rather than real identities
3. Newcomers shouldn't be given any starting advantages so they are always assigned a reputation value of 0, having to steadily build it over the time
4. It should have minimal overhead
5. It should be robust to malicious collectives of peers.

In EigenTrust, the global reputation of each peer is given by the local trust values assigned to it by other peers, weighted by the global reputations of the assigning peers. Trust values are normalized so no peer can assign arbitrarily high or low values to other peers in order to subvert the system. Then trust is aggregated by taking into account peers' recommendations weighted with the trust the node has on those other peers. Each peer has a number *M* of score managers and since each peer also acts as a score manager, it has assigned a set of daughters referenced by the indexes of peers whose trust value computation is covered by the peer. As a score manager, a peer also maintains the opinion vector of its daughter peers and it also learns the set of peers which downloaded files from its daughter peers, receiving trust assessments from these peers referring to its daughter peer. Finally, a peer also gets to know the set of peers which its daughter peers downloaded files from and the trust assessments on those peers from its daughter peers. Their results have shown a reduced number of inauthentic files on the network under a variety of threat scenarios and furthermore, that rewarding highly reputable peers with better quality of service incentivizes honest peers to share more files and to self-police their own file repository for inauthentic files.

Tegeler et al. [40], try to find a solution for identity authorization and validation in decentralized dynamic networks such as peer-to-peer networks, where it is common that nodes can interact with other unknown nodes and it is a difficult task to achieve without a centralized authority. To optimize the quality of service experience and to exclude malicious nodes, various trust and reputation systems (TRSs) have been proposed but most of these rely on properties that require a central authority or other elements that are incompatible with decentralized environments. Hence, the authors argue that TRSs are vulnerable to Sybil attacks as most of these mechanisms usually rely on the assumption that most nodes in a network behave well and cannot handle multiple virtual identities owned by a unique entity. To tackle the problem they present *SybilConf*, a simple yet effective scheme to drastically increase the cost of maintaining an identity, thus reducing the Sybil attack's impact. The basic underlying idea is to combine public/private keying with a puzzle creation algorithm to enable Sybil protection in a given system. Their puzzle creation relies on the fact that factoring numbers into primes is a computationally costly operation, which can deter entities from abusing the system or issuing/showing malicious requests or behaviours. In order to create computationally costly puzzles, the authors have created a protocol on which a node would send a challenge to the requesting node in order for it to solve it prior to engaging in any activity with such aforementioned node. The protocol consists on the following 3 steps:

1. Node A sends to node B his public key, which acts as its identifier, and three integers *i, j* and *k*, where *i* and *j* are offsets and *k* is a length which is used to take a k-long piece of A's and B's public key from offset i and j respectively. This message is signed with A's private key to prevent any alteration of the message while in transit.
2. When B receives this message, if it is willing to solve the puzzle it needs to concatenate a k-long portion of A's public key starting on offset *i* to a k-long portion of his own public key starting at offset *j*, and to find the prime factorization of the result of this concatenation.

3. Finally, B sends the result of this factorization back to A along with his public key, in an again signed message with his own private key.

As prime-factorization is hard to achieve but extremely easy to check, A will not have to spend vast amounts of computation power to check if the result of the prime factorization operation is right, thus avoiding possible denial of service attacks coming from B. Meanwhile, B would have spent a considerable amount of computing power in solving the puzzle, thus rendering multiple identities useless and very expensive to maintain.

In Quercia et al. [41], the authors address the problem of using mobile devices such as smart phones, with which people may create and distribute different types of digital content, being one of the problems that digital content is easy to create and replicate and may likely swamp users rather than benefiting them. To avoid that their solution allows users to organize content producers that they know and trust in a web of trust. Users may also use this web of trust to form opinions about content producers with whom they have never interacted before and these opinions will then determine whether content is accepted or not. They call the process of forming opinions trust propagation and they have designed a mechanism for mobile devices that effectively propagates trust and that is lightweight and distributed, as opposed to previous works that focus on centralized propagation). Their mechanism uses a graph-based learning technique which they evaluate in terms of predictive accuracy against a large real-world data set and also taking into account the computational performance on mobile devices. The basic steps of their solution are as follows:

1. Device A determines the trust relationships that the propagation schema may find relevant for predicting A’s trust in Device B;
2. A restricts its attention to the subset of the web of trust that it knows and that includes those relationships;
3. From this subset, A builds a relationship graph;
4. Finally, it applies a machine learning technique to determine a function that predicts A → B.

In their evaluation, they have measured the prediction accuracy, robustness and overhead by applying their methodology to a large subset of data available from Advogato’s community, showing that their algorithm’s accuracy is as high as 82.9%. In the cases where the algorithm failed to produce an accurate prediction (17.1%) the actual rating and the predicted rating differed by one order of magnitude only. Finally, from computational overhead point of view, the maximum amount of data to send is 30 kB and the runs of the algorithm take an average of 2.8 ms.

Fallah et al. [42] tackles the problem of applying reputation systems in self-organizing mobile ad hoc networks, where a node uses the recommendations made by the others when evaluating the reputation of the node under consideration. This renders the system vulnerable to the Sybil attacks when incorrectly handled, as an attacker can forge several identities and use them to recommend itself as a well-behaved node or downgrade the reputation of a honest node by down voting it repeatedly. In order to counter this, the authors propose a strategy for reputation systems that discourages Sybil attacks. The underlying concept is that for a Sybil identity to remain trustworthy, it should be active and sincere in recommending other nodes, but to be able to carry out an effective attack, the owner of the multiple identities should incur the cost of maintaining the trustworthiness of its Sybil identities. Such a feature can be exploited to design a reputation system where the attack becomes more costly than a cooperative behaviour. In their system each interaction involves one node requesting a service from another node, which may be accepted or refused. Each node has one or more pseudonyms – virtual ids – and can get new ones whenever it wants. Nodes are not demanded to pay for their new identities and it is also assumed that the ids of two different nodes are disjoint. Regarding the reputation system, every node records in a table the reputations of those identities it is aware of and for each of those identities, there are two entries: its reputation in delivering services and its reputation in making honest recommendations. The recommendation requests from an identity that does not participate in recommending the others are disregarded, which the authors assume to be a sufficient incentive for a node to be active and sincere in making recommendations. Furthermore, recommendations from an id which is known for its dishonest recommendations are not used in computing a node's reputation thus making necessary for a Sybil identity to be active and honest enough to be afterwards able to launch an attack. Moreover, to make a recommendation nodes should consume some resources, which makes maintaining the trustworthiness of a large number of Sybil identities very costly for an attacker. In their simulation, they use a multi-stage game, on which in every stage two nodes interact and they decide in between "*To Sybil*" or "*Not to Sybil*". When a node makes the decision on which strategy to use, it computes as well the cost incurred from using it and the utility gained, meaning that a node opting for a "*To Sybil*" strategy has to take into account the cost of maintaining all the Sybil identities up to the current stage of the game and the resources wasted by deploying the attack and maintaining those identities. The results of the simulation prove that with their system, the more nodes that comply with a good behaviour, the less a Sybil attacker could get from her attack, which makes the system even more resilient. It is noteworthy nevertheless that they take some other security assumptions for granted, which are solved by extra mechanisms on top of their strategy.

SybilDefender by Wei et al. [43] leverages particular characteristics present in social networks in order to detect Sybil nodes in the system. The authors base their work in the fact that Sybil nodes tend to be surrounded by other Sybil nodes and that they usually have to make a small cut in order to reach the honest region, which honest nodes do not need to do. In contrast with SybilGuard of Yu et al. [44], the authors of this paper claim that their system can cope with much bigger social networks in smaller time raising effectively the lower bound of Sybil nodes to be detected and population to be dealt with. The edge linking an honest node with a Sybil node is called an attack edge and their system has three main components:

1. A Sybil Identification algorithm, which takes a social graph, a known honest node and a suspect node as inputs and outputs whether the suspect node is Sybil or not. This algorithm is based on the fact that as there is a small cut between the honest region and the Sybil region, the random walks originating from a Sybil node tend to get stuck into the Sybil region. Also, the authors assume that the size of the Sybil region is not comparable to the size of the honest region so the number of nodes traversed by the random walks originating from an honest node will be larger than the number of nodes traversed by the random walks originating from a Sybil node.
2. A Sybil Community Detection algorithm, which once a Sybil node has been found detects other Sybil nodes in the surroundings of the previously found node, forming effectively a "*Sybil Community*".
3. Limiting the number of attack edges, which consists on allowing users to rate their relationships from friends to strangers so even if a user can create many Sybil identities, they would not get many friends assignments among honest users.

Their evaluation shows that SybilDefender can correctly identify the Sybil nodes even when the number of those comes close to the theoretically detectable lower bound, and it can also effectively detect the Sybil community surrounding a Sybil node.

Tran et al. [45] proposes a system called *GateKeeper*, which has as central component a ticket distribution process in order to manage node admission control. Each node acting as a ticket source disseminates a certain amount of tickets throughout the social network until a significant portion of the honest nodes receive some of them. The way to propagate the tickets employs a breadth-first- search (BFS) approach: Each node is placed in a conceptual way at a BFS-level according to its shortest path distance from the ticket source node. The source node then divides the tickets evenly and sends them to its neighbours who keep one ticket each and distributes the rest evenly among their neighbours at the next level. If a node does not have any outgoing links to the next level, it simply destroys all remaining tickets and this process continues until there are no tickets remaining. Since it is deemed that the attackers only control a small number of attack edges according to social network properties, a randomly chosen ticket source should be relatively far away from most attack edges, resulting in few tickets propagated along an attack edge. As a result, GateKeeper is able to directly use a received ticket as a token for a node's admission. To perform admission control, a controller first selects a number of random ticket sources by performing an equal number of random walks. Afterwards, the controller asks each of the chosen ticket sources for its reachable node list and it admits a node if and only if that node has appeared in more than a threshold number of reachable lists returned by the chosen ticket sources. In their simulation results, GateKeeper proves to work well both limiting the amount of Sybil nodes admitted to the system and maximizing the amount of honest nodes admitted to the system.

Finally, the thesis of Titi [46] tackles the problem of the inherent risk of selecting a Wi-Fi network to use from the ones that are in range, since as the author states, there is no way yet to select the most trustworthy Wi-Fi network nowadays. In his thesis, the author focuses on devising a way that can help users to choose the best Wi-Fi network around them.

In his PhD thesis, the author presents a solution that allows users to rate the networks they have used and to check that their assessments reflect the true network quality they have experienced by measuring and certifying the quality of service such as delay, jitter and packet loss. The measuring process follows an innovative protocol that certifies the measurement in different cases. In fact, there is one remaining case that we cannot certify. He mitigates this case with trust management.

The author has validated his solution through simulation using AnyLogic and comparing the results of his solution to those of two previous well-known trust metrics: EigenTrust and Salem.

The comparison of his solution with that proposed by Salem and with EigenTrust's algorithm shows that his solution is more robust than Salem's because it is resistant to attacks such as inserting malicious APs and inserting malicious users. Regarding EigenTrust, his solution can deal with a higher percentage of malicious APs among all APs than EigenTrust.

3.2 Related Work on Cooperation Incentives

This section introduces the related work done in the area of cooperation incentives. The main goal of the systems and frameworks reviewed in this section is to encourage the users to behave in the right way by providing incentives that foster cooperation and reward them.

Feldman et al. [47] focus on the issues present in P2P networks and that make the challenge of achieving cooperation more complicated than in other environments. Some of those issues are large populations, self-interest, zero-cost identities, dynamicity of the system and short-lived population. In order to address cooperation and to provide incentives, they have created a reciprocative decision function having the following three requirements:

1. It can use shared and subjective history;
2. It can deal with defections;
3. It is robust against different patterns of defection.

They use what they call normalized generosity in order to compute the probability of a peer cooperating with another peer. Generosity is the measure of the benefit an entity has provided relative to the benefit it has consumed, given by **Formula 1**:

$$g(i) = {}^{p_i}/_{c_i}$$

Formula 1. Generosity formula.

Then, the normalized generosity measures one peer's generosity relative to another peer's generosity as shown in **Formula 2**:

$$g_j(i) = {}^{g(i)}\!\big/_{g(j)}$$

Formula 2. Normalized generosity formula.

Using these concepts, the authors show through a game theoretic approach to cooperation in peer-to-peer networks how their approach addresses the challenges imposed by P2P systems, including large populations, high turnover, asymmetry of interest and zero-cost identities. Their results prove that the adoption of shared history and discriminating server selection techniques can mitigate those aforementioned challenges and also that cooperation can be established even in the presence of zero-cost identities through the use of an adaptive policy towards strangers. Finally, colluders and traitors can be kept in check via subjective reputations and short-term history respectively.

Koutrouli et al. [2] deal with the attacks and misbehaviours suffered by most P2P systems nowadays. Free riding and badmouthing are two of the most important problems affecting P2P systems and the authors argue that providing incentives can help reducing those problems. A credit-based recommendation exchange is proposed in order to provide incentives for honest participation in P2P reputation systems where payments for recommendations are based on the trustworthiness of peers regarding the accuracy of the recommendations they give. The payment value (v) is a virtual amount, which will be transferred between peers' virtual accounts but without performing a real currency transfer, and the formula is designed so the recommendation reputation of the buyer and that of the seller impact the calculation. If the former is higher than the latter, the payment will be always lower than one, while in the opposite situation it will be always higher than one. In their system, every peer starts with an initial account balance that determines the highest amount of recommendation exchange transactions that a new peer can get involved before running out of credit. The account balance of a peer is updated after each exchange. The maximum amount of times that a peer, which has always the worst recommendation reputation value than its peers, can participate in the system is limited due to the nature of the calculation formula. Finally, they also implemented a recommendation exchange protocol using an overlay that implements the payments. Simulation results show that the dishonest recommendation behaviour results in non-participation in the reputation system, whereas honest recommendation behaviour results in the maximum utility of the reputation system, thus effectively providing an incentive for honest recommendations and good behaviour.

Zhong et al. [48] propose Sprite, a simple, cheat-proof and credit based system in order to foster cooperation among selfish nodes in mobile ad-hoc networks by providing incentives to mobile nodes to cooperate and to report actions honestly. Compared with previous approaches, their system does not require any tamperproof hardware at any node. In this paper, they present a formal model of their system and prove its properties evaluating a prototype implementation, showing that the overhead of the system is small and that the mobile nodes can cooperate and forward each other's messages unless the resource of each node is extremely low. In their system design, when a node sends its own messages it loses some credits to the network because other nodes incur a cost when forwarding the messages. On the other hand, when a node forwards other nodes’ messages it gains credits and it is therefore able to send its messages later by using that earned credits. In order to get credits when forwarding other nodes’ messages, a node needs to report to a command and control system (CCS) the messages it has helped to forward. In order to save bandwidth and storage, instead of requiring the whole message as a report the system uses small receipts. Such receipts are derived from the content of the messages but do not expose the exact content thus, although Sprite requires that the CCS is trusted in terms of maintaining credit balance, the nodes do not need to trust the CCS in terms of message confidentiality. For a node to get the credits deserved, as verifying with absolute certainty that the node has indeed forwarded a given message is not possible with the current technology, the authors propose that a node gets credits for forwarding a message if and only if his successor also claims a receipt for forwarding the same message. The system does not require that the total amount of credits charged to the sender is equal to the total credits received by other nodes for a message. In fact, in order to prevent some variants of cheating, the CCS charges the sender more than it finally distributes among the rest of the nodes in the forwarding path. To prevent the potential cheating action of the intermediate nodes, the CSS reduces remarkably the amount of credits given to the intermediate nodes if the message is not reported to be received by the destination. With such reduction of credit the cheating nodes cannot get enough to even cover the minimum expense needed for this type of cheating, i.e., the cost of forwarding a receipt. Simulations prove that the system actually encourages and incentivizes the nodes to forward the messages at any time, unless their power is too low.

Wu et al. [49] state that cooperation incentives can greatly reduce a server’s workload by utilizing the user nodes’ resources. Such an application has to rely on the cooperation of all participating nodes but normally the nodes are selfish in nature and to appropriately incentivize the cooperation among each other is crucial. In order to tackle the aforementioned problem the authors design a simple but effective reward based incentive mechanism to encourage the nodes to upload and/or forward data in between them. By using a repeated game strategy to simulate their system, they study the long-term intelligent interactions between nodes and the content providers and they also design a cheating-proof mechanism analysing its effectiveness under the simulation settings. Their system implements an incentive mechanism under which the content providers rewards the nodes based on the data transmission rate they contribute to, including both the upload rate to other nodes and the download rate for data forwarding. Once a node dedicates its transmission rate it incurs an energy cost, while the content providers can benefit from a reduction in the high workload at their servers reducing as well the operational cost. In exchange the content providers reward the nodes which contribute in various forms, like real money and/or virtual credits or reputation points for advanced services. In their simulation approach the authors assume that all the nodes are homogeneous and use the same strategies at each round of the simulation game. A single game involves a content provider and any particular node, where the content provider aims to minimize its total cost, i.e., the cost of uploading and the cost of rewarding the nodes. With the results of their game simulation they prove the stability and efficiency of their solution and they also analyse the threatening strategy and cheating prevention mechanisms.

Zhang et al. [50] focus on P2P networks and their design, where incentive mechanisms that promote cooperation must be introduced and the existing ones are prone to various attacks. False reports, or worse, a colluding group of malicious nodes in a P2P network can manipulate the history information of its own members and the damage of these kind of attacks increases dramatically with the group size. Such malicious nodes or colluding groups are difficult to detect, especially in a large network without a centralized authority. In their paper the authors propose a new distributed incentive scheme in which the extent that a node can benefit from the network is proportional to its contribution. Malicious nodes can only attack others at the cost of their own interests and a colluding group cannot gain advantage by cooperation regardless of its size. The proposed schema includes three major components:

1. A distributed authority infrastructure;
2. A key sharing protocol;
3. A contract verification protocol.

In their system each node is associated with two parameters, namely money and reputation. The providers earn money and also reputation by serving others while the consumers pay money for the service. If a consumer thinks that the received service is not worth the money paid, it can report it to an authority specifying the amount of money it believes overpaid and if the authority can determine who is lying the liar is punished. Otherwise, the authority freezes the money claimed to have been overpaid and the sum will not be available to the provider nor to the consumer either, which eliminates any reason for the consumer to lie. If the provider is guilty, the consumer has the revenge and the provider’s reputation suffers and in the other hand, if the provider is innocent, the consumer does it at a cost because it has paid the price of the transaction while in addition, the falsely accused provider will not provide any service to that given node anymore. In the absence of a central authority the authors have designed a distributed authority infrastructure where each node is assigned a delegation, which consists of a given amount of nodes picked pseudo-randomly. This delegation keeps track of the node’s money and reputation and any anomaly in the information stored at the delegation members may indicate an attempt to forge data. In order to counter the possibility of data forging, the information is legitimate only if the majority of the delegation members agree on it, which implies that as long as the majority of the delegation members are honest, the information about that node cannot be forged. To compromise a delegation, the malicious or selfish nodes from a colluding group must constitute the majority of the delegation, but unless the colluding group is very large the probability for this to happen is small because the identifiers of the colluding nodes are randomly assigned by the system and the identifiers of the delegation are also randomly picked. The authors also present a key sharing protocol and a contract verification protocol to produce the contract proofs that are authorized by the delegations of the provider and the consumer of a transaction. Finally, they analyse the system properties and use simulations to evaluate the system performance showing that it has the potential to solve the free-riders problem in P2P networks.

Mahmoud et al. [51] present a receipt free cooperation incentive scheme for multi-hop wireless networks. The authors have designed a system in which the participating nodes submit payment reports containing the breakdown of charges and rewards combining them with undeniable security evidences. The fair reports can be cleared with almost no processing overhead, while in the cheating reports the evidences are requested and used to identify and evict the cheating nodes from the system. Their system, called RISE has four main phases:

1. A communication phase where the nodes are involved in a service exchange, the resulting evidences are stored and the payment reports are submitted to a centralized accounting centre.
2. A classifier phase on which the accounting centre classifies the reports into fair and unfair.
3. A cheater identification phase where the accounting centre requests the evidences to identify the cheaters.
4. Finally a credit account update phase where the accounting centre clears the payment.

In their system, the authors have implemented several measures to cope with the most common and known attacks affecting P2P networks:

- In order to deal with double clearance attacks, where the attacker claims a session’s payment multiple times to increase its rewards, and double spending attacks, where the attacker attempts to generate the same evidence for different sessions to pay once, each session has a unique identifier.
- Evidence forgery and manipulation attacks, where the attackers attempt to forge evidences or to manipulate valid evidences to steal credits, are not possible in RISE as it uses secure hash functions and signature schemes.
- Finding colluding hashes and forging signatures is not possible, as it requires a vast computing power.
- Free-riding attacks, where two colluding intermediate nodes manipulate the packets to add their data to communicate freely, are not possible since the integrity of the packets is verified at each intermediate node thus the packets are dropped if this kind of attack is detected.
- Finally, their payment model can counteract cheating actions through the encouragement of node cooperation. In order to prevent cheating, both the source and the destination nodes are charged for non-delivered messages.

Extensive analysis and simulations demonstrate that their system can clear the payment with almost no processing overhead while achieving a good security protection against the most common attacks.

Finally Aldini et al. [52] propose that the success of user-centric networks strongly depends on the willingness of the participants to cooperate and that incentives can help in encouraging users to cooperate. To this end, reputation-based incentives and remuneration incentives are introduced to increase the users' motivation and to discourage selfish behaviours. In their work, quantitative properties of cooperation incentives are defined and analysed through model checking. Their model considers users providing services, which are called providers and users receiving services, which are called requesters, presenting four phases of cooperation:

1. Discovery and request;
2. Negotiation;
3. Transaction;
4. Evaluation and feedback.

Their reputation system defines cooperative attitude, which depends on dispositional trust and on service trust level, which represents the threshold under which the service is not accessible. For the service request to be accepted by a given node, the trust computed for the provider should be higher than the service trust level threshold. The authors also introduce a virtual currency system where reputation-based and reward-based incentives are combined by including the trust level T of the provider towards the requester as a parameter affecting the cost of the negotiated service. Cost is calculated as follows in **Formula 3**:

$$C(T) = \begin{cases} C_{min} + \dfrac{C_{max} - C_{min}}{T'} \times (T' - T) \; if \; T < T' \\ C_{min} \; if \; T \geq T' \end{cases}$$

Formula 3. Cost computation.

Where the parameters are: C_{min}, which is the minimum reward asked by the provider regardless of his/her trust on the requester, C_{max}, which is the maximum reward asked to serve untrusted users, and T', which is the trust threshold above which the minimum cost is applied to the requester. Finally, they prove through Markov decision process analysis that mixing incentive strategies such as reputation and reward proves effective in inducing cooperative behaviours and also that cooperation incentives favour both requesters and providers, as honest requesters get services at a lower price and reputation and cooperative behaviours impact earnings in providers.

3.3 Related Work on Fully Decentralized Sustainable Wireless Networks

This section introduces the related work done in the area of fully decentralized sustainable wireless networks. The main goal of the systems and frameworks reviewed in this section is how to implement security and other measures to ensure their survivability given that:

1. They are composed of elements which are scarce in resources like energy, bandwidth or storage;
2. Routing overhead and a highly dynamic topology increase their complexity.

In Hubaux et al. [53], the authors tackle the problematic of security in mobile ad-hoc networks, as it is normally quite difficult to achieve given, among others, the vulnerability of the links, the limited resources available and the dynamically changing topology. In their work they start by defining the threats that affect the most these networks and which can be directed not only against the basic mechanisms but also against the security mechanisms themselves. Regarding the vulnerabilities affecting basic mechanisms, they highlight the risk of nodes being hijacked, eavesdropping, active interferences as the communication are carried out over the air, non-cooperative nodes, vulnerabilities related to the routing mechanisms and malicious neighbour discovery. Regarding the vulnerabilities affecting the security mechanisms directly, they address mainly the risks of cryptographic keys being compromised or replaced with other keys. In order to protect the basic mechanisms their choice is to use tamper resistant hardware and smart cards to protect the cryptographic information, while also aiming to protect the routing mechanisms by using watchdogs and rating paths in combination with intrusion detection systems (IDSs). Finally, in order to enforce the service, they use a virtual currency called *nuglets* as a cooperation incentive.

The work in Pirzada et al. [54] states that the execution and survival of an ad-hoc network is exclusively dependent on the cooperative nature and trustworthiness of its nodes. The problem they find is that it is actually this same dependency on intermediate nodes that makes an ad-hoc network vulnerable to passive and active attacks carried out by malicious nodes. There are a good amount of protocols that have been developed to secure ad-hoc networks using cryptographic schemes, but almost all of them rely on the presence of a central trusted authority and as the authors state, dependence on a central trust authority is an impractical requirement for ad-hoc networks as their dynamic topology and spontaneous nature makes this highly unfeasible. In order to tackle this problem, the authors propose a model which implements trust-based communication in ad-hoc networks and that also proves that a central trusted authority is not always a strong requirement. Their model introduces the notion of belief and provides a dynamic measuring of reliability and trustworthiness in a given ad-hoc network. Their trust model uses an adaptation from Marsh's [21] work, but modified in order to be used in ad-hoc networks. In their work, the authors make use of trust agents that reside on each of the network nodes and each agent operates independently and maintains its individual perspective of the trust hierarchy. In the regular operation cycle of an agent, it first gathers data from events in all the states, then it filters it and assigns weights to each event and finally it computes different trust levels based upon them. They after use this trust model to enhance the dynamic source routing (DSR) protocol in order to find the most trustworthy routes from one node to another, improving the survivability of the network by avoiding routes containing malicious nodes. Also, as the model presented operates passively and has minimal computation and energy requirements, it also improves the sustainability of the network by saving the energy and bandwidth of the nodes.

In Mitchell et al. [55] the authors address the survivability issues of mobile cyber physical systems (MCPSs), which comprise human actors, vehicles, or robots carrying sensors, assembled together for executing a specific mission in the battlefield or in an emergency response situation. In order to implement their solution they have developed a mathematical model to assess the survivability property of a given MCPS which is subject to energy exhaustion and to security failures. Their model-based analysis reveals that the optimal design for providing effective intrusion detection capabilities is to best balance energy conservation versus intrusion tolerance. This approach provides the highest possible survivability level. Finally, they test the effectiveness of the approach with a dynamic voting-based intrusion detection technique, demonstrate its validity through simulation. In their simulations, the authors consider possible attacks like node hijacking and data injection and they also use a Byzantine failure mode, in which if a third of the nodes are compromised the system fails. In their dynamic voting intrusion detection system each node periodically exchanges its routing information, location, and identifier with its neighbour nodes. Afterwards, a coordinator is selected randomly among neighbours so that the adversaries will not have a specific target beforehand. The coordinator selects a certain amount of vote participants randomly and it lets all voters to know each other identities so that each voter can disseminate its vote to others. At the end of the voting process all voters should have received the same result regarding if a certain node is regarded as good or bad, based on the majority vote. Finally the authors apply simulation to validate their model in order to identify the optimal design settings and to check that, despite the voting intrusion detection system, their approach effectively maximizes the lifetime of nodes improving the survivability of the network.

Chorzempa et al. [56] tackle the problem of robust, secure and efficient communications in highly unattended wireless sensor networks. The authors state that communication reliability is the key for the survivability of those networks, which often are deployed in hostile environments and are left unmanned, exposing the nodes to attacks like captured or tampered nodes. In their solution the authors present their Survivable and Efficient Clustered Keying (SECK) system, which introduces robustness and recoverability to a highest extent than other existing solutions. In the paper they prove through simulation that SECK is highly resistant against key and node captures and that it has several advantages over other key management schemes, among others lower computational cost and communication overhead. In order for their system to be able to scale to big sized networks, the authors propose a two-tier architecture where the basic tier is a set of nodes clustered around an aggregation-and-forwarding node (AFN) and the second tier is composed by the aggregation of these first level clusters. Each cluster needs a set of keys to be deployed and managed in order to secure communications between its nodes and SECK provides two sets of keys for each of the nodes in a cluster, namely administrative keys and session keys, which are distributed using the former. In order to manage administrative keys within a cluster, the authors use an exclusion basis system where the nodes have only a unique subset of those keys so the administrative keys cannot be compromised if a given node is captured. The capture of an aggregation-and-forwarding node is tackled by a two-step procedure which notifies all the nodes in the cluster if the AFN is captured, after what a new AFN is selected to reorganize the cluster around it. Their simulation results show that clusters can be kept into an ideal size of around 50 nodes, making the amount of hops in between them and the AFN smaller, being able to cut down communication overhead. Finally, the simulation also proves that SECK is resilient against node and key captures.

In Mohammad et al. [57] the authors claim that normally wireless sensor networks (WSNs) are deployed in adverse environments, where failures of sensor nodes and the disruption of connectivity are recurring issues. This implies that the organization of the WSNs needs to be able to adapt to the ever changing conditions in order to maximize survivability but always taking into account that energy efficiency in WSNs remains the main concern to achieve a longer network lifetime. In their solution they associate survivability and energy efficiency with the clustering of WSNs and they show that their scheme can actually increase the survivability and lifetime of such networks. Their solution consists on an easy to implement method named DED, which stands for distributed, energy-efficient, and dual homed clustering. DED provides robustness for WSNs without relying on the redundancy of dedicated sensors and uses the information already gathered during the clustering process to determine alternative backup routes from sources to observers, thus incurring in lower message overhead compared to other approaches. The correctness of the algorithm is proved analytically and their simulation results demonstrate that their solution is effective as for instance they take into account criteria such as the remaining lifetime of the nodes when volunteering to be cluster heads, the energy expenditure incurred in by the communication inside the cluster and also which cluster head would make the balance of the cluster better lifelong wise when being selected by the rest of the nodes. Finally, the authors compare their algorithm with other similar ones and conclude that DED performs better in general and results in lower energy expenditure and overhead.

The work by Kamal et al. [58] surveys and discusses a variety of survivability issues, challenges and mechanisms in multi-hop wireless networks. In their paper the authors provide a classification of the methods that can be used in order to improve network survivability, such as:

- Protection mechanisms like software/hardware redundancy or backup solutions or routes;
- Restoration mechanisms like service re-discovery or route re-computation;
- Hybrid mechanisms which are a mixture of the former two;
- Coding based mechanisms like error correction and detection codes.

Afterwards they enumerate the survivability issues and challenges related to network survivability, like scalability problems, network connectivity, path redundancy and failure recovery approaches and they conclude the paper by listing the survivability mechanisms available like proactive protection mechanisms such as Ad-Hoc On-Demand Distance Vector Multipath (AODVM) routing or Multi-path Dynamic Source Routing (MP-DSR). They also list the available reactive protection mechanisms where multiple paths are known in advance before the communication session is started like Split Multipath Routing (SMR) and coding-based protection mechanisms where some of the path or transmission redundancy is eliminated by introducing error detection and correction codes which can reconstruct corrupted data saving bandwidth and energy.

In the thesis of El Maliki [59], the author states that conventional security protocols cannot deal with dynamic attacks as they tackle problems in a rather static way. Also, this comes as a disadvantage as they limit the efficiency of the resources present in sensors and overall network efficiency gets diminished. The key problem identified by the author is that there is a lack of security adaptation protocols that are general enough to deal with extremely dynamic security conditions in the context of mobile wireless networks, where reliability and survivability are critical.

To solve these issues he proposes a Security Adaptation Reference Monitor (SARM) for wireless and mobile environments that aims to offer a framework able to integrate new and emerging security mechanisms to deal efficiently with the security of complex systems. SARM is based on an autonomic computing security feedback loop that improves in each round the present security means by monitoring the current context, including the user environment and the energy consumption, resulting in a self-managed and self-optimized framework.

In order to validate his results, the author has carried out extensive simulations using an agent-based approach, in order to verify the performance of SARM in the presence of attackers. His results prove that SARM is efficient in terms of survivability, overall network utilization and power consumption. Also, as part of his thesis work, the author has implemented and evaluated SARM by attempting to select the best access point through the use of trust and reputation paradigms taking into account factors such as energy performance and comparing his solution against the consumption of different security methods used in his framework. His results and the analysis of the data obtained from the simulation experiments indicate that the security management works as intended and that security is provided efficiently for different situations.

To finalize, Xing et al. [60] focus on the analysis of network survivability in the presence of misbehaving nodes and failures. In order to tackle this problem they propose a novel semi-Markov process model to study the evolution of nodes' behaviours and as an immediate application of the proposed model they investigate the problem of node isolation where the effects of Denial-of-Service (DoS) attacks are considered. The authors also find that the network survivability degradation is directly proportional to the increase of misbehaving nodes and that moreover DoS attacks have a significant impact on the network survivability, especially in dense networks. To finalize, they validate their proposed model and their analytical results using numerical analysis and showing the effects of node misbehaviours on both topological survivability and network performance.

3.4 Identifying the GAP

After having reviewed the most relevant literature on the three main fields related to this thesis, the purpose of this last section is to compare them over the three key points that our framework aspires to achieve in order to evaluate their completeness and their compliance with these characteristics so we can identify the common gap to all of them.

Table 2 presents the comparison of all the frameworks and metrics analysed in the previous sections regarding three main features:

- Whether the framework or metric is fully decentralized or can be applied in a fully decentralized environment.
- If the framework or metric is resistant against the most common attacks affecting these environments, most precisely the Sybil attack.
- If the evaluated frameworks or metrics provide some sort of incentive to foster cooperation and enforce good behaviour among the participants of the system.

Table 2. Frameworks and Metrics Comparison

Framework/Metric	Fully Decentralized	Sybil Resistant	Incentives
Certain Trust [33]	✗	✗	✗
Martucci et al. [34]	✗	✓	✗
Appleseed [35]	✓	✗	✗
Ingram et al. [37]	✗	✓/✗[1]	✗
Eigen Trust [39]	✓	✗	✓
Sybil Conf [40]	✓	✓/✗[1]	✗
Quercia et al. [41]	✓	✗	✗
Fallah et al. [42]	✓	✓/✗[1]	✗

[1] While it provides means to greatly reduce Sybil attacks close to a full extent, it is not 100% Sybil resistant (i.e. there are mechanisms to try to deter Sybil nodes or to detect them, but those not deterred or detected still are able to cheat).

Sybil Defender [43]	✓	✓/✗[1]	✗
Gate Keeper [45]	✗	✓/✗[1]	✗
Feldman et al. [47]	✓	✗	✓
Credible Recommendations [2]	✓	✗	✓
Sprite [48]	✗	✓/✗[1]	✓
Weijie et al. [49]	✓	✗	✓
Zhang et al. [50]	✓	✗	✓
Rise [51]	✗	✓/✗[1]	✓
Aldini et al. [52]	✓	✗	✓
Nuglets [53]	✗	✓/✗[1]	✓
Pirzada et al. [54]	✓	✗	✓
Mitchell et al. [55]	✓	✓/✗[1]	✗
SECK [56]	✓	✓/✗[1]	✗
DED [57]	✓	✗	✗
Xing et al. [60]	✓	✗	✗
SARM [59]	✓	✓/✗[1]	✗
HotSpot Trust [46]	✗	✓/✗[1]	✓

As can be seen from the previous table comparing and summarizing the main characteristics of the metrics and frameworks that have been analysed, none of them comply at the same time with the three basic characteristics that we are aiming for in our framework.

First of all, one of the pitfalls common to some of the analysed frameworks or metrics is that even if they have some elements that can be applied or used in a fully decentralized way, they rely in other elements that need to be central to the framework, thus not fully complying with the fully decentralized paradigm. This in turn means that they cannot adapt themselves so easily in dynamic and constantly changing environments and that these central entities might become a bottleneck in the system, both affecting the survivability and the resilience of the system.

One of the main concerns when talking about decentralized user-centric environments or networks is the fact that this lack of central element or authority often introduces the opportunity of several identity related attacks such as the ones explained in Section 2.1.4. From those attacks, the one of most concern is the Sybil attack, as it is based on controlling multiple identities in the system in order to be able to take coordinated actions that can unbalance the system in an unfair manner. Given that the aim of the thesis is to provide a fully decentralized framework, with no centralized authority or trusted third-party, it is inherently difficult to perform authentication and identity validation leaving such systems open for attacks. This fact determines that one of the characteristics we are aiming for when designing our framework and when comparing the surveyed frameworks and metrics is whether they are resistant or not to Sybil attacks. So far, as can be seen in **Table 2**, there is no framework or metric which is fully resistant to Sybil attacks, even though there are a couple of them which provide the means to greatly reduce Sybil attacks close to a full extent. Nevertheless, they are not 100% Sybil resistant as they present mechanisms to try to deter Sybil nodes or to detect them, but those not deterred or detected still are able to cheat.

Finally, the third characteristic for which we search in the reviewed literature is whether it provides some sort of cooperation incentive or not, as this fact is key both for the survivability of the network and to encourage good behaviour and deter malicious nodes from acting as such. As can be seen from the comparison table, many of the surveyed frameworks (approximately 50%), do not provide any type of incentive to the participating entities or nodes, greatly diminishing the cooperative behaviours in between them and affecting the stability and resilience of the network or system in the long run.

While the literature reviewed presents frameworks which always comply with one or more of the desirable characteristics that we deem as desirable for such decentralized user-centric network environments, none of them is truly resistant to Sybil attacks, and some frameworks or metrics, even when applied to decentralized wireless environments, still rely in some centralized elements or centralized bootstrapping steps. Moreover, as shown in Chapter 3 and to the best of our knowledge, there is no or little work done to study how trust management can be coupled with cooperation incentives in order to empower the latest, in a fully decentralized way and being fully Sybil attack resistant.

Chapter 4. Framework Model and Design

After having identified and precisely defined the gap, we are now going to present and thoroughly describe our framework, which covers this previously identified gap by bringing together an attack resistant trust metric and a cooperation incentives schema in the form of points as a form of reward in a fully decentralized and user-centric fashion.

Our fully decentralized framework has three main core component or building blocks in each of its nodes as depicted in **Figure 2**:

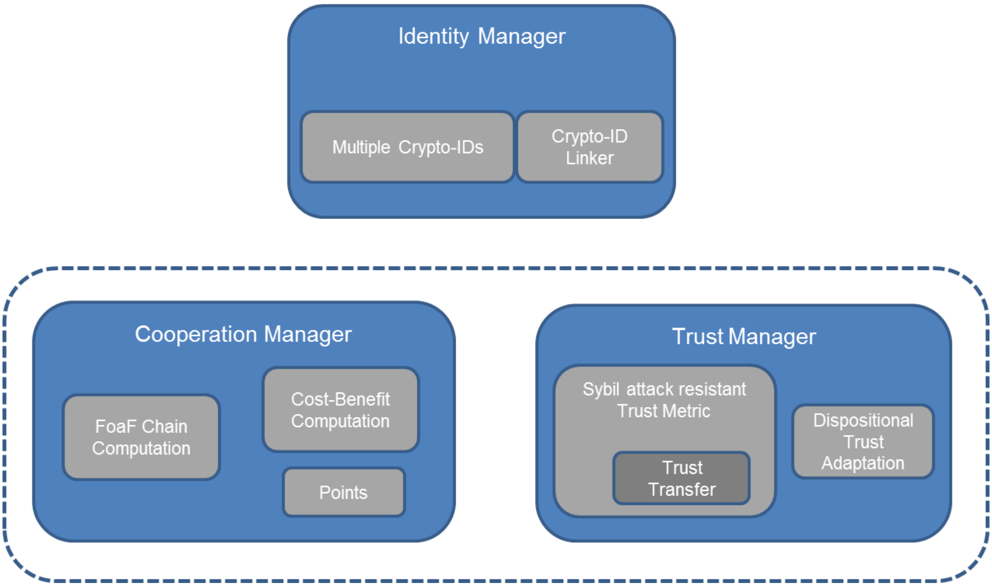


Figure 2. Trust and Cooperation Framework high-level view.

In each of the nodes, the trust manager is in charge of computing trust levels for the different nodes that the user has to interact with, the cooperation manager is dedicated to providing and computing incentives and rewards and finally the identity manager is responsible for generating virtual identities in the form of crypto-IDs for the node.

In the following sections we will first introduce the formal definitions tied to this framework and afterwards we will describe each of the main building blocks and their subcomponents in detail as well as their functionality.

4.1 Component Definitions

This section introduces the formal definitions for each of the components that are related and/or constitute the building blocks of our framework.

4.1.1 User

We define a user on our framework as any entity, be it human or not, which controls and is able to use one or more devices or nodes to interact with the system.

Users can be of two kinds, honest or malicious, which are defined like follows:

- An ***honest user*** is a user that interacts with the system through its device(s) or node(s) in a rightful way, behaving properly, not taking any advantage from whatever flaw the system might have and always providing a service with equal or higher quality than agreed with the counter party.
- A ***malicious user*** in the other hand, is a user which when interacting with the system through its device(s) or node(s) tries to take unfair advantage from the system, exploits it and/or provides services with lower quality than agreed or doesn't provide the service at all.

A user is represented by one or more virtual identities in the form of crypto-ids as per the definition in the Section 4.2.

4.1.2 Device or Node

We define a device or node in our framework as any element that is owned by a user, as defined previously, and that enables a user to interact with the system. The term device and node are interchangeable in our framework and can be used indistinctly.

A device or node can be seen as honest or malicious depending on how it interacts with the system, but in reality it is the user owning and controlling it who is malicious or not, as the devices only act as the mean to interact with the system for a given user, being the latter the responsible for the good or the bad use of them. A node can be of two types:

- **Requester:** a node requesting a service.
- **Provider:** a node providing a service.

4.1.3 Manager

We define a ***manager*** as a core element to our framework, which has a well-defined purpose and a set of basic operations and functionalities. A manager can be of three types, which will be defined and explained later in the next sections, ***trust manager***, ***cooperation manager*** or ***identity manager***.

Each of the managers in the framework is responsible for the part that gives its name and all of them are necessary for the well-functioning of the framework.

4.1.4 Attack

We define an attack in our framework as any intent of directly or indirectly exploiting a vulnerability on the system to gain an unfair advantage over it. We focus mainly on the Sybil attack, but we also consider the attacks listed in Section 2.1.4.

4.2 Identity Manager

The first main building block in our framework is the identity manager, as well present in every node. The identity manager is in charge of creating virtual identities for the user and managing them in order to be able to link trust levels to identities and to combine if needed identities in order to be able to exhibit a higher trust level by linking those virtual identities as can be seen in **Figure 3**.

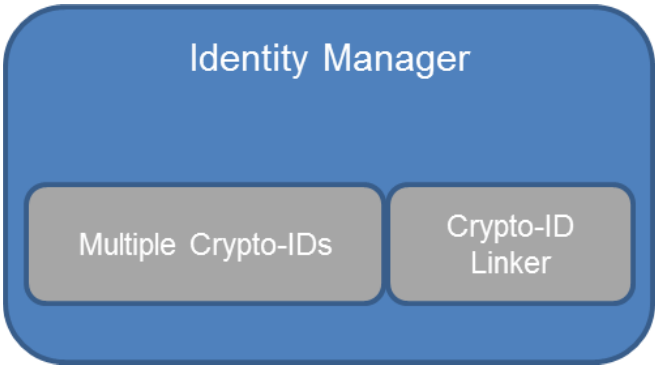


Figure 3. Identity Manager Components.

By design choice, we are going to use in our framework the concept of ***Crypto-ID*** as virtual identity as introduced in Seigneur's PhD thesis [1]. Simply put, a crypto-ID is a piece of cryptographic material belonging to a user, which is then hashed to produce a unique identifier that can represent that user. As cryptographic piece we are going to use the public key of a public-private key pair, which can be generated at will by each identity manager belonging to each of the users in the system, and then hash that public key in order to produce the crypto-ID itself.

The interaction in between the components contained within the trust manager is depicted in the following **Pseudo-code block 1**:

Pseudo-code block 1. Identity manager high level code.

```
CHECK crypto-id
IF needed OR requested THEN
        VERIFY crypto-id linking message
        LINK crypto-ids
REQUEST trust level
END
```

Following, we will explain in detail each of the components of the identity manager.

4.2.1 Crypto-ID

In our framework, a crypto-id is defined according to the following general **Formula 4**:

$$CryptoID_A = f_{HASH}\ (PubKey_A)$$

Formula 4. Crypto-ID computation.

Where f_{HASH}() represents any available hashing function or algorithm such as SHA-1/2/3, MD5, etc., and $PubKey_A$ is the public key from the private-public key pair from node or device A.

4.2.2 Multiple Crypto-IDs

Given that strong enrolment in a centralized authentication manner is not available in our framework, as we want it to be fully decentralized, we cannot rely on each of the users to have a validated and unique identity. In the other hand, the usage of multiple crypto-IDs facilitates attacks at the identity level on trust management, for example, as said before voting several times with different virtual identities owned by the same user, namely Sybil attacks. In order to solve this issue, rather than trying to forbid the users to create multiple virtual identities, our framework allows them to create many virtual identities based on crypto-IDs and mitigates potential attacks by providing an attack-resistant trust metric as introduced in previous sections, Trust Transfer. Although this approach is more difficult to achieve than unique crypto-IDs at the trust management level, it allows for the creation of fully a decentralized user-centric network environment framework, and also the improvement of privacy protection by default, as the users can choose to split their actions among different pseudonyms or crypto-IDs, making it more difficult to have a complete view of the actions executed by one user and find her real-world identity through action linking or extensive data collection or mining.

To create multiple crypto-IDs, the user can generate multiple key-pairs that will correspond to different pseudonyms that she can use in different situations during her interaction with the decentralized environment. Those key pairs would be used to sign requests and messages and to identify herself.

4.2.3 Crypto-ID Linking

Still according to Seigneur's PhD thesis [1], in order to implement a mechanism balancing trust with privacy, as we said in the previous section we allow users to freely create pseudonyms identified by the crypto-id, i.e., the hash of the public key of a locally generated asymmetric cryptography key pair. Then, depending on the context, one or another pseudonym can be used to carry out actions logged as events signed with the private key of the pseudonym.

If needed, one or several pseudonyms could also be linked together in order to increase the number of known actions and potentially increase the trust in the linked entity assuming that all these actions had a positive outcome. As each crypto-ID is able to sign, two crypto-IDs can both sign a special message, called “crypto-IDs linking message”, saying that they are linked together meaning that they have the same owner. In this way, the trust level linked to each of the crypto-IDs is proven to belong to the very same user, allowing her to carry out an action that perhaps couldn't be taken with the trust level linked to one of her single crypto-IDs alone.

4.3 Trust Manager

The second main building block in our framework is the trust manager, which is present on each node. The trust manager is in charge of managing the dispositional trust adaptation for the user's device(s) and for providing trust computation in order to assign trust levels to other nodes in the system using a Sybil resistant trust metric as can be seen in **Figure 4**.

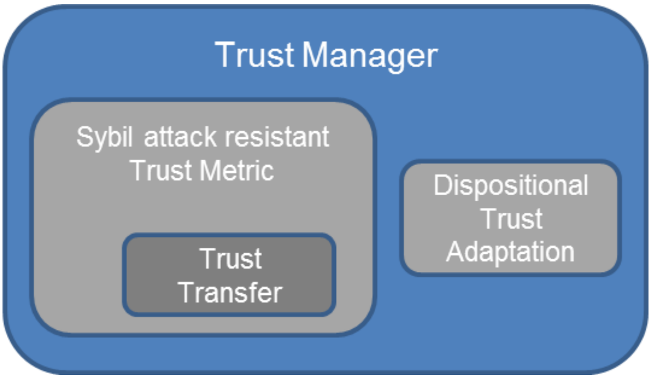

Trust Manager
Sybil attack resistant Trust Metric
Trust Transfer
Dispositional Trust Adaptation

Figure 4. Trust Manager Components.

The interaction in between the components contained within the trust manager is depicted in the following **Pseudo-code block 2**:

Pseudo-code block 2. Trust manager high level code.

```
CHECK crypto-id database
COMPUTE trust level
IF trust chain is needed THEN
        ASK cooperation manager
ELSE
IF trust level > threshold THEN
        COMPUTE profit
        CALL cooperation manager
ELSE
        DENY service request
END
```

Following, we will explain in detail each of the components of the trust manager.

4.3.1 Dispositional Trust Adaptation

User-centric networks are supported both by static, fully dedicated nodes as well as by nodes provided by end-users on the fly. Since some nodes are carried by Internet end-users, their networking composition, surrounding environment and organization can rapidly change.

Dispositional trust reflects the disposition of a certain individual to trust or not "per-se", and it is mostly represented by a fixed value which doesn't change over the time.

We define dispositional trust $D_T \in [-1, 1]$ where:

-1 = full distrust and

1 = full trust

Contrary to this, we think that dispositional trust is a value that, as far as it is modelling a part of human trust, it is also subject to changes. When trying to copy human behaviour and translating it into computational trust notions and representations, we think that it is important to take into account, that the disposition to trust usually changes over the time according to the situation and surrounding environment and the interactions with it. As such, a hostile environment might turn an open-to-trust individual into a distrustful one, and in the other hand, a reliable environment can turn a distrustful individual into a more willing-to-trust one. The aim of adapting dispositional trust is twofold:

- Firstly, it can be helpful when trying to protect a node in such dynamic and highly changing environments, where often nodes have not had yet an interaction with many other nodes in the system and thus the uncertainty is high.
- Secondly it can help the metric to converge faster to realistic trust values in extreme environments, i.e. environments with a big majority of misbehaving nodes, as dispositional trust will lower the overall trust level of a given node faster than when dispositional trust has a fixed value.

Dispositional trust adaptation is based on the adaptation rate and adaptation step intervals defined following. Adaptation rate defines how many malicious or honest interactions a node needs to have in order to re-adapt its dispositional trust and adaptation step defines the numeric amount in which the dispositional trust level is adapted, i.e. how much it decreases or increases after a certain amount of malicious or good interactions with other nodes in the system. We define adaptation rate A_R and adaptation step A_S as:

$$A_R \in \{1, 2, 3, 4\} \text{ and } A_S \in \{0, 0.2, 0.3, 0.4\}$$

We normalize D_T as D_{TN} into [0, 1], when needed, according to the following **Formula 5**:

$$D_{TN} = 0.5 + \frac{D_T}{2}$$

Formula 5. Dispositional trust normalization.

Finally, we adapt dispositional trust as shown in **Formula 6**; every time a requester accumulates a number of malicious interactions M_I or good interactions G_I then:

$$D_T \begin{cases} if\ M_I \geq A_R \quad then\ D_T = D_T - A_S \\ if\ G_I \geq A_R \quad then\ D_T = D_T + A_S \end{cases}$$

Formula 6. Dispositional trust adaptation.

4.3.2 Trust Level

We define trust level from A → B as T_{AB}, direct trust from A → B (direct observations) as TD_{AB}, recommendations as TR and trust threshold T_T, all $\in$ [0, 1].

4.3.3 Trust Metric

We define trust metric as any function or formula that computes the trust level of a given node. In our framework, we have used Trust Transfer [1] as it is proven to be Sybil resistant, but it would be possible to use any other existent or newly developed metric as far as functionality is concerned.

4.3.4 Trust Transfer

Trust transfer, a trust metric by Seigneur [1], has been proven to protect against Sybil attacks when pieces of evidence are limited to direct observations and recommendations based on the count of event outcomes. Trust transfer implies that recommendations move some of the trust level of the recommending entity to the trust level of the trustee. This approach is particularly efficient for our system as, besides assessing trust, we can use the metric to reward in the form of trust points the agents that share their Wi-Fi connectivity. This effectively combines trust management with cooperation incentives as will be explained in following sections.

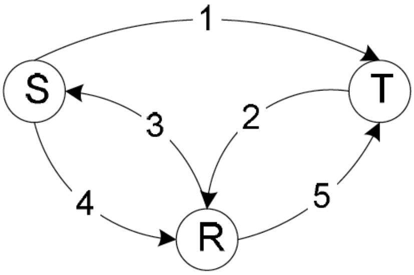


Figure 5. Trust Transfer process [1].

Based on **Figure 5**, Trust Transfer works in the following manner:

1. The subject (S) requests an action, requiring a certain amount of positive event outcomes - trustworthiness is based on event outcomes count in Trust Transfer – in order for the request to be accepted by the trustor (T).
2. If S has not enough trust, T queries its contacts to find recommenders (R) willing to transfer some of their positive event outcomes count to S.
3. If the one or more contacts have interacted previously with S and the contacts' trust balance with T allows it to permit to transfer an amount of the recommender's trustworthiness in S, the contact agrees to recommend the subject. It queries the S on whether it agrees to lose that same amount of trust on the recommender's side.
4. Subsequently S returns a signed statement, indicating whether it agrees or not.
5. Finally, R sends back a signed recommendation to T, indicating the trust value it is prepared to transfer in behalf of S, including the signed agreement of S.

4.4 Cooperation Manager

The third main building block in our framework is the cooperation manager, which is also present in every node. The cooperation manager is in charge of computing the cost and benefit balance of a given action or service exchange, managing the node's points balance and searching and computing friend-of-a-friend chains in order to find potential point lenders as can be seen in **Figure 6**.

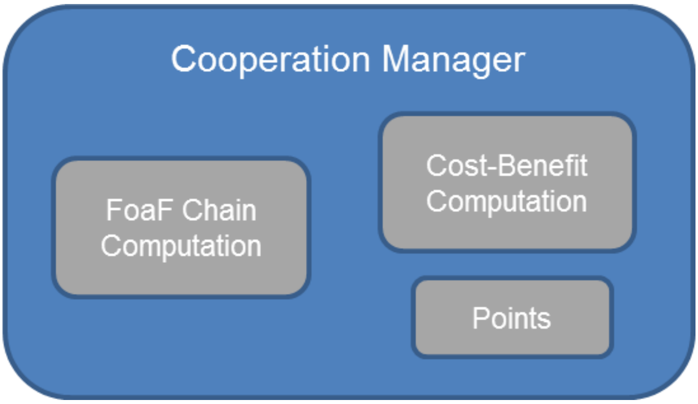


Figure 6. Cooperation Manager Components.

The interaction in between the components contained within the trust manager is depicted in the following **Pseudo-code block 3**:

Pseudo-code block 3. Cooperation manager high level code.

```
CHECK points for action
COMPUTE profit
IF profit > 1 THEN
        COOPERATE
        ACCEPT service request
ELSE
        DENY service request
END
```

Following, we will explain in detail each of the components of the cooperation manager.

4.4.1 Cooperation Incentives: Points and other Rewards

In order to foster interaction amongst users in a collaborative environment such as the one described in this thesis, there is a need to offer incentives to the users besides providing them with the appropriate safety features such as a solid trust metric.

Trust Transfer can effectively be used as a cooperation incentive enabler, by using its trust points as the de facto “currency” in order to be able to use the services other users have to offer, in this case Wi-Fi connectivity sharing. By awarding trust points to the service provider proportionally to the duration of the Wi-Fi sharing period, we foster cooperation among users as not only the trust points reflect the good behaviour of the user giving her a good reputation, but also enable her to in turn obtain Wi-Fi connectivity when roaming or being out of data by using those trust points earned previously in order to pay for the service. The more you share in the system, and the more different users you share with, the easiest will be to in turn find another user which will accept your trust points as payment, be it because of having interacted directly with her or using trust transfer mechanisms to find another user who can lend the service requester those needed points as explained in the previous section. We reckon that these incentives are limited by your own circle of direct interactions and acquaintances inside the system, and this is why we exploit another capability of trust transfer, which is being able to transfer trust points through chains of trust with multiple hops, as explained in the next section.

Cooperation Points

We define cooperation points as the points given, in the case of a service requester, or gained, in the case of a service provider, when a service exchange happens in between two nodes.

Points are defined in units in the range of [1, +∞[, and being 1 the minimum amount of points able to be gained or paid for a given service.

Modelling Cooperative Behaviours

For a cooperation incentives schema to work, a basic premise that needs to hold is that the potential benefits obtained from behaving good is greater than the potential costs incurred when performing an action in the system. In our collaborative sharing service, we model our cooperation incentives schema in form of credits, which can be obtained when sharing a Wi-Fi access and spent when using other user's Wi-Fi access. Then, we compute the profit P of cooperating according to the following utility function shown in **Formula 7**:

$$\mathrm{P} = \frac{B_p}{C_p}$$

Formula 7. Profit computation formula.

In this equation, B_p stands for the potential benefit a user can obtain when behaving good and C_p stands for the potential cost of performing a certain action, where *P* is contained in the interval [0, +∞[and represents whether the action to be taken will yield a profit or not for the entity performing the action.

From the general **Formula 7** we can devise different cost-benefit computation formulas for our Wi-Fi sharing service and their corresponding strategies for potential attackers. The most straightforward approach is to define the cost-benefit in terms of trust points, given that they are the main currency of our framework. In this case, the cost or benefit that a certain action would yield would be defined by **Formula 8**:

$$\mathrm{P} = \frac{Points_{gained}}{Points_{needed}}$$

Formula 8. Point-based profit.

In this formula, $\text{Points}_{\text{gained}}$ would be the points awarded when providing the service and $\text{Points}_{\text{needed}}$ would be the amount of points needed in average to buy a similar quality service in the system. In this case, the most efficient strategy for an attacker would be to maximize profit by overpricing the service and requesting a higher amount of points that what the service is worth, in order to be able to buy much bigger amounts of service from other providers with the points earned.

Nevertheless overpricing the service would lead to service requesters searching for cheaper service providers, and assuming a general fairness of the nodes participating in the system, similar services could be found with lower point prices. The only case in which the strategy could work would be in the case that there is no other service provider nearby, in which case the higher price would be tied to the offer and demand law, and the overpricing would be justified given the scarcity of the service.

Another way to compute the cost benefit from a Wi-Fi sharing perspective would be computing the profit in terms of data volume or bandwidth. Similarly to the previous formula, the only viable strategy for an attacker would be to give a way lower bandwidth or data volume than advertised, in order to profit the system with the points obtained by providing the service, but this in turn would drain down its trust level rapidly, making the strategy unviable.

After computing the profit of an action using the utility of choice (points, bandwidth, etc.), we then take the decision D on whether to cooperate or not according to **Formula 9**:

$$\mathrm{D}\begin{cases} \quad\quad if\ P > 1, \quad\quad cooperate \\ else, non - cooperative\ behaviour \end{cases}$$

Formula 9. Decision computation.

4.4.2 Small World Networks

To empower the cooperation incentives provided by Trust Transfer and the trust points, some other mechanism in order to extend the usefulness of those points needs to be introduced, as Trust Transfer contemplates mainly that trust points are to be used “one-to-one”, or as most with one degree of indirection. This means that in a scenario where several strangers are supposed to cooperate and to share services, it would be difficult to spend those points as the likeliness of finding in the same environment another user which one has already interacted with, or as most within one degree of separation is highly unlikely.

In order to overcome this limitation, we have explored the probabilities of finding longer “friend-to-friend” chains applying the principles of small worlds, first stated by Milgram [61], and degrees of separation. For the sake of simplicity, we assume that most of the system’s users come from networks which are already highly connected, such as Facebook.

Social Networks like Facebook have been proven to have a degree of separation of around 4.76 to 6 with almost a 100% of probabilities [62], [63]. The problem of finding the probabilities for a subset of a small world network to find a chain of 6 degrees of separation or less can be modelled as random node failures (different from targeted attacks) in the complete network until we are left with the desired amount of nodes, which would be our subset of the small word network. In order to model a social network like Facebook, we need to use a scale-free network which exhibits both short paths and high clustering degree. Such a network can be modelled by using a Klemm and Eguíluz (KE) [64] Network, which is a type of scale-free network which complies with both properties.

Degrees of Separation

We define degrees of separation in our framework as it is defined in small world networks' mathematical models of Watts et al. [65], being it the distance in between two given nodes in the user-centric network environment of our system.

The formal definition of distance (***d***) can be seen in the following **Formula 10**:

$$d = 1 + \frac{\ln N}{(\ln k + (\ln k - 1))}$$

Formula 10. Distance computation.

Where ***N*** is the number of nodes conforming the user-centric network, and ***k*** is the number of links per node to other nodes in the network.

4.4.3 Friend-of-a-Friend Chains

While the most used metrics to determine the properties of a network are L (characteristic path length) and C (clustering), those can produce misleading results when used to re-evaluate such properties when eliminating large portions of random nodes, as disconnected or isolated users or small unreachable clusters can skew the results. It is thus a better estimate of the properties of a network, as stated in Crucitti et al. [66], the one produced by the global and local efficiency (E_{glob} and E_{loc}). The efficiency of a network is defined as the effectiveness of the network to propagate information both globally and locally, meaning the probability of finding a path in between two nodes of that network for the information to propagate. Those definitions can be modelled mathematically as seen in **Formula 11** and **Formula 12**.

$$E_{glob}(G) = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}}$$

Formula 11. Global efficiency formula.

$$E_{loc} = \frac{1}{N} \sum_{i \in G} E(G_i)$$

Formula 12. Local efficiency formula.

Taking this formula into account, and applied over a network inducing random failures and targeted attacks, the authors have come up with the results that can be seen in **Figure 7**.

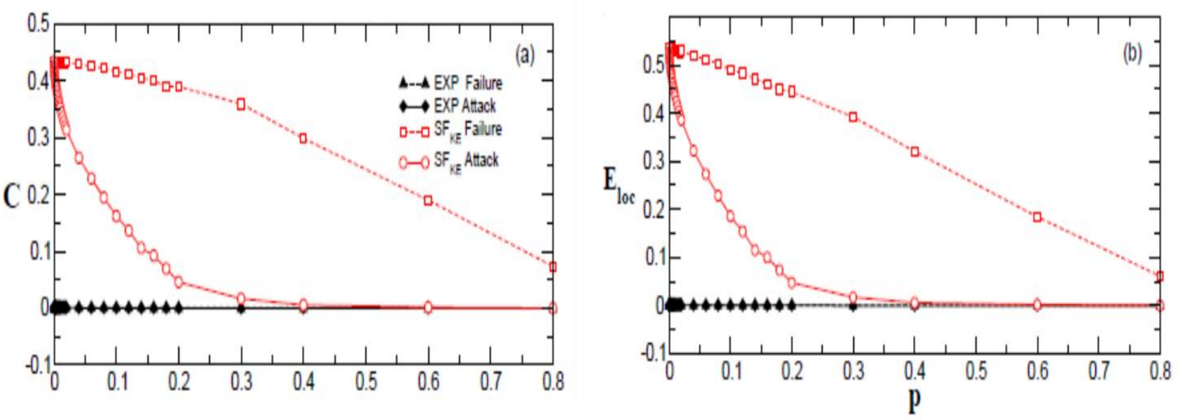


Figure 7. Clustering and efficiency for percentages of random failure in nodes and targeted attacks [66].

As we have seen in the previous graphs, until the network is not at least a 20% of the original (less than 80% of the nodes have failed due to random failures), the efficiency or clustering size is not big enough to even consider it a functioning network, as the probability of finding a path to transmit information is lower than 10%. Nevertheless, there are other aspects that have not been taken into account in the purely mathematical demonstration:

- Facebook is especially high clustered (much more than any of the networks in the previous results), to which one could argue that the removal would not impair the network as badly as that.
- When users decide to adopt system which is collaborative and based in friendships, most likely it will be adopted in an «epidemic» way, on which friends and friends of friends would install it, leading to an also highly clustered and connected sub-network.
- The interactions between disconnected users while using our system, would in the long run create a small world by itself.

In our simulations, we apply these same principles and we calculate for a given user base population, how quick the full system would bootstrap and which is the minimum amount for such a user base which would enable reasonable probabilities of finding such FOAF chains so the cooperation incentives are more useful and in turn, encourage the users to cooperate and behave properly.

Implicit Social Network

By virtue of trust and friend-of-friend chains, and driven by continuous interaction in between the users or nodes taking part in our framework, an implicit social network is created in our system. Applying the principles of small world networks described previously, the bigger the user base of our system is, the easier is to find a chain connecting two given nodes. Besides that, the constant interaction between nodes while using our Wi-Fi sharing service makes the network more connected and thus increasing the chances of finding such chains, effectively building a new kind of trust-exchange social network on top of our system.

4.5 Framework Components Interaction

After having presented each of the individual components and subcomponents that integrate our framework, we are going to finally define how they interact together in the framework when a node receives a service request. The main interaction diagram can be seen in **Figure 8**.

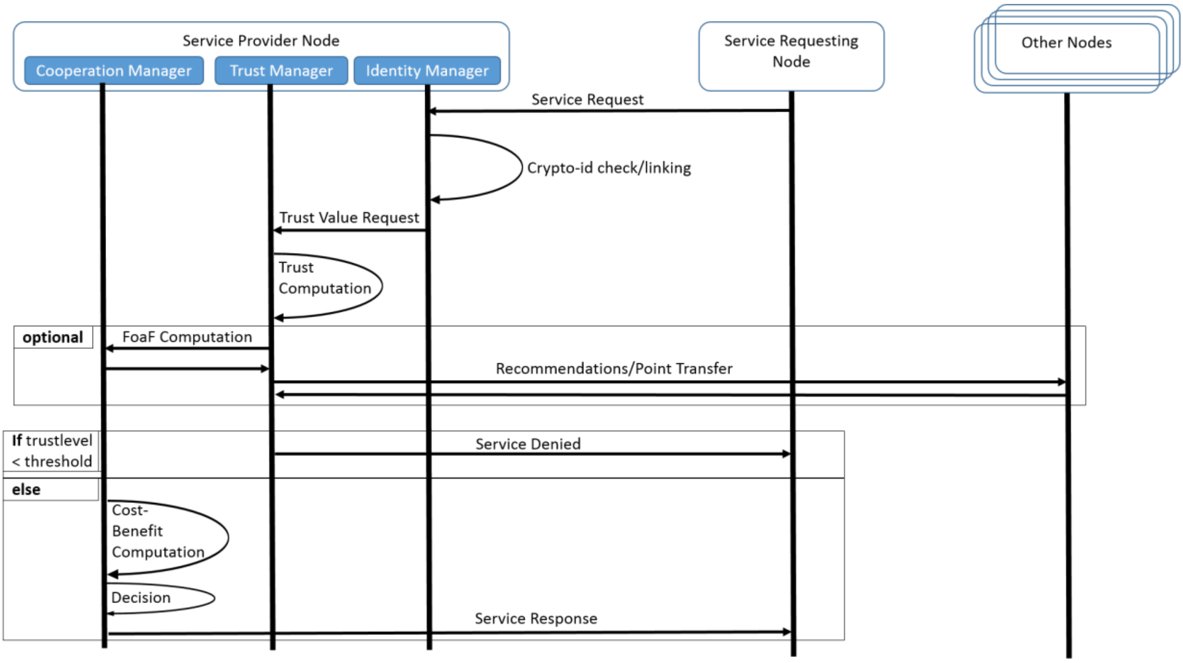


Figure 8. Framework component interaction.

As shown in the previous figure, a regular service request interaction would be as follows:

1. The service request coming from another node arrives at a given service provider node. It contains the crypto-id of the requesting node signed with his private key for authentication purposes.
2. The identity manager does the crypto-id authentication and checks if there are other crypto-ids that can be linked to this one.
3. A trust value request is initiated.
4. The trust manager checks the balance of trust points of the requesting node. If it is sufficient, no other action has to be taken and trust computation can proceed. If it is not enough, the trust manager will try to find through FoaF chains if there are other nodes that can transfer some

trust points in behalf of the requesting node in order to fulfil the required trust balance to access the service.

5. A cost benefit analysis is performed in order to know if there are enough incentives to provide the service, i.e. the benefit is higher than the cost.
6. A decision, taking into account the trust computation and the cost-benefit analysis, is taken.
7. The service response is transmitted back to the requesting node.

In order to give a more detailed view of the behaviour both of a service requester and a service provider node, we have summarized the main action flowchart in the following **Figure 9**.

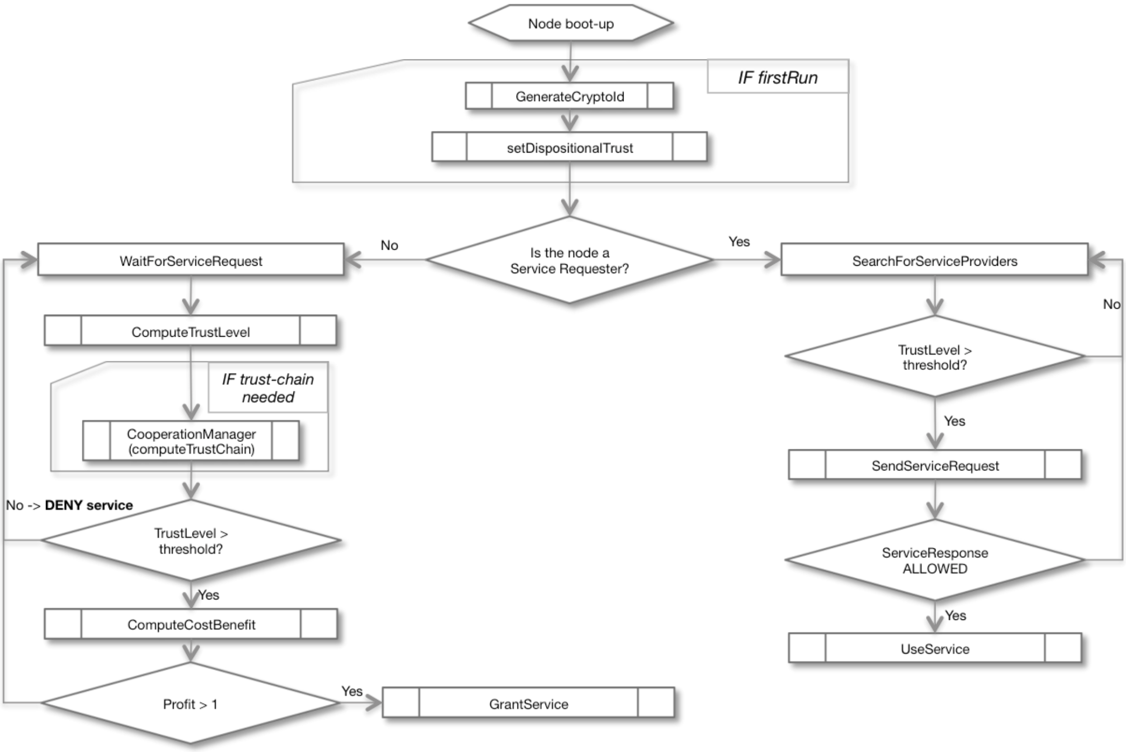


Figure 9. Framework flowchart.

Chapter 5. Implementation and Evaluation

After defining our framework both formally and technically in the previous chapter, we need to validate it in order to be sure it works as intended, is meaningful and actually improves from the current state of the art evaluated in Chapter 3.

5.1 Tools

In order to validate our framework as stated before, we have used different sets of tools and methods that have allowed us to test it, simulate it and collect meaningful data. In this section we introduce and describe the tools and methods employed for validation, so the reader can have a better understanding on what has been done in these regards when going through the results attained.

5.1.1 AnyLogic

AnyLogic [17] is a simulation framework that supports all the most common simulation methodologies in place today, namely System Dynamics, Process-Centric (Discrete Event) and Agent Based modelling. It is based on an object-oriented model design paradigm, which provides for modular, hierarchical, and incremental construction of large models, being able to hold thousands of agents at the same time.

We have chosen AnyLogic as our main simulation tool because of several characteristics that make it appealing for our simulation purposes:

- Object libraries provide the ability to quickly incorporate pre-built simulation elements
- Given its object-oriented structure, reusability of any given component, be it pre-build or self-coded, is possible throughout all the model
- It supports the combination of discrete and continuous simulations
- The native Java environment supports limitless extensibility including custom Java code, external libraries, and external data sources
- An extensive statistical distribution function set provides an excellent platform for simulating the uncertainty inherent in all systems
- Its animation functions allow the development of visually rich, interactive simulation environments
- The native Java environment provides multi-platform support: both the AnyLogic IDE and models work on Windows, Mac and Linux

All of these reasons make AnyLogic the perfect tool for us to simulate large scale models with thousands of agents, being able as well to integrate random patterns and behaviours, which otherwise would be complicated to simulate. Also, the visual representation offered by AnyLogic allows us to better represent our framework and model for others to understand, without having to focus only on a particularly technical audience.

5.1.2 Empirical Testing

Empirical testing consists on actually implementing a prototype or a basic working version of a given system or framework in order to obtain results as close as possible to real-life ones. By using empirical testing, we are able to actually experiment how the framework works and responds for real instead of basing our assumptions on simulation results, which while valid and usually accurate, many times include randomization of certain values to better simulate reality and possible outcomes after a given action and sometimes rely on statistical or probabilistic approaches in order to infer some of the values used.

In order to carry out an empirical testing of our framework, we have developed a basic prototype of it in Android OS using the Eclipse framework and the Android SDK libraries.

5.2 Simulations with AnyLogic

In this section we are going to explain in detail each of the simulations carried out using AnyLogic. These sets of simulations have different purposes, and each of them targets one or several characteristics of the framework, sometimes in isolation and sometimes combined with the rest of the framework.

The aim of these simulations is twofold:

- To determine how much a single characteristic or property of the framework can actually improve the framework overall
- To test how the framework reacts and works as a whole

In the next sections we will present the scenarios and the results of each of the simulations done with AnyLogic.

5.2.1 Connectivity Performance Results in Highly Dynamic Environments

Given that our framework allows for service sharing in between nodes connected to the system, it is important to initially study whether it is feasible to share such services in a way that are still usable taking into account service's setup time and the time window where two or more nodes are close enough to be effectively connected to each other.

The aim of this simulation is to measure how effective a framework like ours can prove to be in a highly dynamic environment such a ski slope or in general in any situation where a high degree of node mobility is expected, as if our framework is functioning well in such an environment, it will also work in a more static environment such as an airport or a train station.

To this end, we have chosen to simulate a ski slope scenario where users carrying nodes or devices ski freely while automatically sharing connectivity as a service. In order to study the technical feasibility of our approach from a connectivity perspective, we measure for each foreign skier his or her connectivity duration time and their connectivity status, be it “connectivity setup” or “connected”.

Scenario

The experiment shows the simulation of a ski slope, with local skiers which have connectivity through 2G, 3G, 4G, HSDPA etc. and foreign skiers which in principle do not have connectivity of any kind. The simulation has been carried out using AnyLogic's agent based simulation capabilities, assigning real values and proportions to the scenario.

When the simulation is started, an introduction screen that depicts the scenario and asks for the skier concentration rate is shown. The concentration rate defines how busy the ski slopes are, and thus influences the availability of connectivity for foreign skiers. It ranges from 1 (not busy), which will deploy zero to one skier, either foreign or local, for each ski lift arrival, to 4 (very busy), which will always deploy four skiers (maximum ski lift capacity) per ski lift arrival. The foreign-to-local skier's ratio is 30%-to-70%, in order to reflect the real statistics of Megève, first smart ski resort [67].

The ski lifts are modelled with a discrete-event approach, having the appropriate inter-arrival time and speed. Every time one ski lift arrives to its "sink element", 3 skiers are deployed according to the concentration rate, as explained previously. We have chosen 3 as the concentration rate as it reflects a moderately full ski resort, which is the case we want to analyse. When the skiers are deployed, they are assigned an initial random speed, which ranges from the slowest to the highest average speed of a regular downhill skier, which ranges from 25 to 40 kilometres/hour, and a final destination at the end of the slope. In order to make the simulation more realistic, the trajectory between the deployment point to the end point is not set as a straight line but as a sinusoidal function which imitates the real movement of a skier. The ski slope's length is set to 800 metres and the width to 80 metres, as it is a good estimate of the length of a typical ski slope according to Megève Tourism board and Megève ski lifts' company.

Foreign skiers' terminals scan for access points, and when they are in range of one (or many) of them, they try establishing a connection. The range of the portable access points has been set to 40 meters in the simulation, as it is the typical average value of that of a mobile device such as an Android or an iPhone terminal.

Once connected, the foreign skier proceeds to upload and/or download useful data such as skiing stats (speed, position, etc…), slope maps and the like for the duration of the connection (assuming regular 3G/4G/HSDPA rates), and when she or he is disconnected the process starts over again until the skier reaches the end of the slope and goes into a ski lift. Connectivity setup time, the time that it takes for one skier to stablish a connection to another skier's access point, has been set up to 50 seconds as it is the average time shown in the real performance test results.

Results Analysis

In order to study the feasibility of a real application on the ski slopes, we have measured during simulation experiments 3 sets of different data. For all the graphs, the X axis represents the simulation time:

- Global amount of foreign and local skiers in the ski slope at any given time of the simulation.
- Amount of foreign skiers in “connection setup” state vs. the amount of foreign skiers in “connected” state at any given time of the simulation.
- The maximum amount of time a skier has been connected at any given point of the simulation time and the global average time a skier is in “connected” status.

The first graph, depicted in **Figure 10**, shows the statistics for the global amount of local and foreign skiers present in the simulation slope at any given point of the simulation time, being the X-axis the time and the Y-axis the amount of skiers. As can be seen in the graph, the amount of local skiers ranges in average from 30 to 40 while the amount of foreign skiers ranges from roughly 10 to 20. This gives a ratio of approximately 67% of local skiers and 33% of foreign skiers, which adequately reflects the real ratio between French local skiers and foreign skiers in Megève ski resort according to Megève Tourism board. As can be inferred from this numbers, the total amount of skiers in the slope at any given point of time ranges from 50 to 60, which is a realistic measure for what a moderately busy ski slope would look.

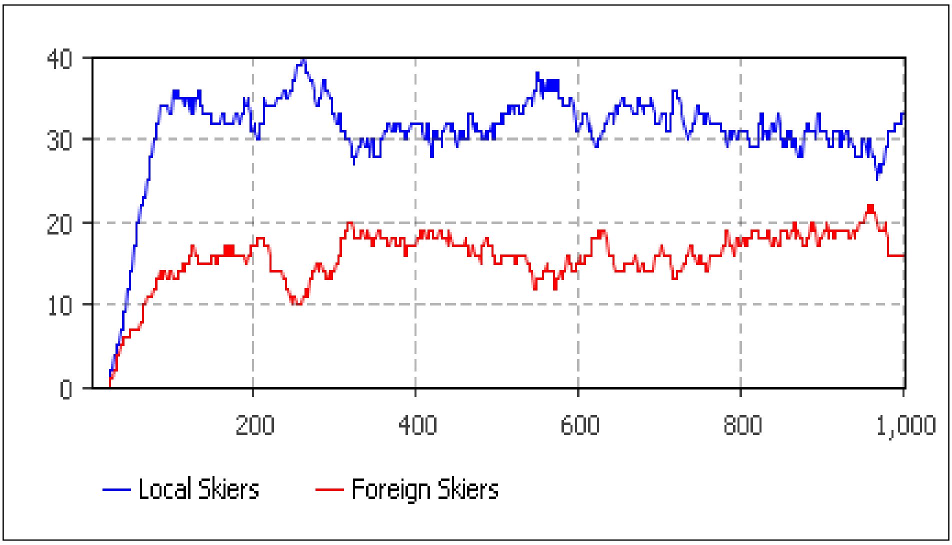


Figure 10. Amount of local and foreign skiers in the slope at any given point of the simulation time.

The second graph in **Figure 11** shows the amount of foreign skiers that are setting up a connection with a local skier versus the amount of foreign skiers that are already connected and thus, transmitting data, being the X-axis the simulation time and the Y-axis the amount of skiers in each of the two states. As can be seen in the graph, from the 10 to 20 foreign skiers present in the slope at any time, around a 20%-25% have an established connection, while 70%-75% are into “connection setup” state. The remaining percentage accounts for those who have no local skier to try to connect to. This data correlates properly both with the total amount of foreign skiers present in the slope at the given simulation time and with the expected results of the simulation, as the connection setup phase takes 50 seconds, thus allowing only skiers that have been for at least in connection for 50 seconds with a local skier in the slope to have an established connection.

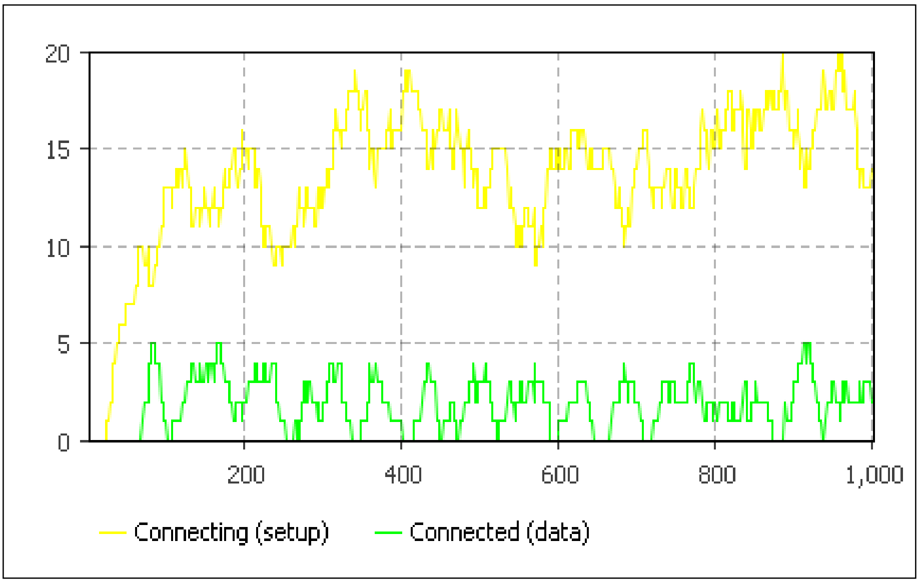


Figure 11. Amount of foreign skiers in connection setup and connected states at any given point of the simulation time.

Finally, the third graph in **Figure 12** shows the maximum amount of time (in seconds) a skier has been already connected in a given point of time, and the global average of the time (in seconds) skiers in general are in a connected state, being the X-axis the simulation time and the Y-axis the amount of seconds of achieved connectivity. We can see in the graph that in average, skiers are connected during 10 seconds to a local skier, reaching maximum connection times of over 20 seconds. This fits into the expected results, as the speed of a foreign skier is in the 25 to 40 kilometres/hour range as previously said before, which makes an average speed of 32.5 kilometres/hour (9 metres/second). Taking into account that the length of the slope is 800 metres, we can establish that it takes nearly 88 seconds to complete the full length of the slope.

A perfect descent would imply that the foreign skier is in “connection setup” state for 50 seconds, having 38 extra seconds to be in a connected status. Taking into account that the trajectories and speeds of the skiers in the slopes are not uniform, that connectivity setup can start later in time than when the skier first reaches the slope and that the connectivity setup phase can be broken if the foreign skier goes out of the area of coverage of the local skier’s hotspot, we assume that the results obtained in the simulation adjust to the reality of the situation well.

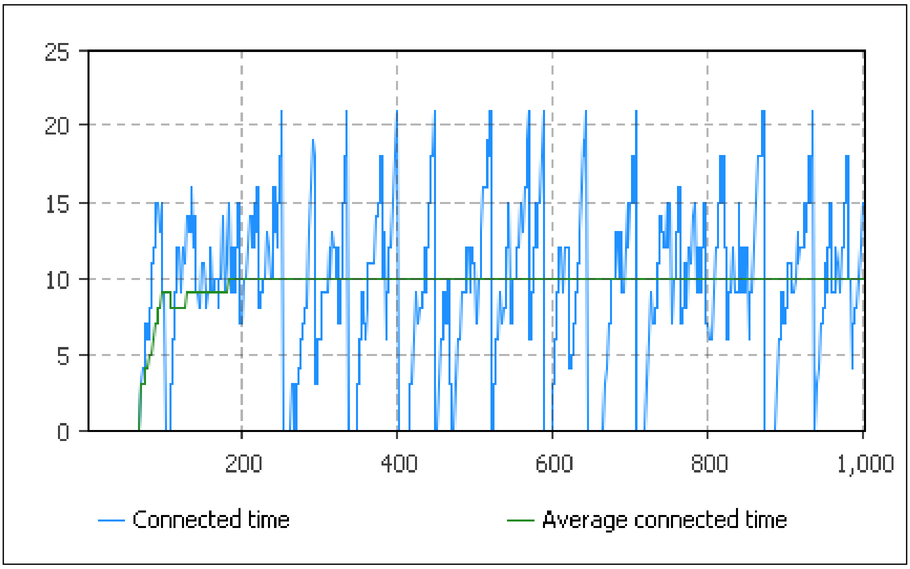


Figure 12. Average connected time for any foreign skier during the simulation time.

All in all, the results obtained in the simulation phase imply that foreign skiers have enough time to at least do some light data exchange, as the average 3G/4G connection averages real data rates from 200Kbps to 1Mbps [68]. This light data exchange can consist for example in updating slope maps and status in real time, which can take approximately the order of 0.98 to 2.86Mb with a decent zoom level [69] and/or uploading meaningful statistics about speed, distance and any other valuable data to a server when using a smart application which offers this capabilities. All of this without having to take care to connect manually or use their smartphone actively.

Also, these results imply that in a static situation, such a local cafeteria, restaurant or shop, connectivity would be obtained without any significant problem, as the situation is much less demanding than that of a ski slope.

5.2.2 Dispositional Trust Adaptation Results

The aim of this simulation is to determine whether using dispositional trust adaptation, which is a new technique developed for this framework, can improve the overall reliability of a given trust metric when it comes to detect or avoid malicious nodes. In this thesis, we define trust metric as the equation or function used to compute and to update the trust level of a given node. Note that the metric choice here is independent from the one used in our framework, as the objective is to check if this technique can improve in general any given framework, in order to justify the inclusion of it in our own framework. Thus, we have chosen very generic trust metrics in this simulation in order to see whether dispositional trust adaptation really makes a difference in general or not.

To this end, we need to also know which dispositional trust adaptation rate is the optimal in order to obtain a balanced trade-off between security and network utility. Following, we proceed first to describe the metrics used in each simulation scenario as well as our definition of network utility.

Metrics

In ***Metric 1*** only the distance is taken into account in order for the requester to attach to a provider, as the closer the latter is, the stronger the signal will be and (theoretically) the better the service. This metric is interesting for our simulation in order to see how a basic and free environment in which no trust levels are used performs, and how this impacts both in terms of amount of requesters connected to malicious providers and in network usability.

Metric 2 combines dispositional trust, which remains constant with a value of 0 (neutral) throughout all the simulation, with direct observations and recommendations. In this metric, the trust level is computed as shown in the following **Formula 13**:

$$T_{AB} = \frac{D_T}{2} + \alpha * \ TD_{AB} + (1 - \alpha) * TR$$

Formula 13. Trust level computation.

Where: $TD_{AB} = \frac{\sum_{i=1}^{n} TD_{AB_i}}{i}$, being TD_{AB_i} one particular direct observation and *n* the total amount of direct observations.

$TR = \frac{\sum_{j=1}^{n} TR_{jB}}{n}$, being TR_{jB} the computed recommendation from node *j* towards B and n the total amount of nodes recommending, received by all the other nodes and discarded if the recommender's trust value is below 0.5.

And $\alpha = 1 - D_{TN}$

As can be seen, in this metric D_T is fixed to a neutral value of 0, and it is only used as D_{TN} when there is not any direct observation yet, in order to supply an initial TD_{AB} value. Obviously, $T_{AB} > T_T$ in order for the requester to attach to the requestee.

Metric 3, which combines dispositional trust, which can self-adapt over the time depending on the rest of the entities in the system, with direct observations and recommendations. Trust level is computed as in Equation 2, and D_T adapts according to the following procedure:

Every time a requester accumulates a number of malicious interactions M_I or good interactions G_I then we update the dispositional trust value using **Formula 14**:

$$D_T \begin{cases} if\ M_I \geq A_R \ \ then\ D_T = D_T - A_S \\ if\ G_I \geq A_R \ \ then\ D_T = D_T + A_S \end{cases}$$

Formula 14. Dispositional trust adaptation.

As can be seen from this formula combined with the previous one and its sub-equations, the dispositional trust adaptation affects both to the weight of the recommendations and to the overall T_{AB} computed.

Network Utility

The network utility for a given scenario is calculated as follows in **Formula 15**:

$$Network_{Utility} = \frac{QoS_{RECEIVED}}{QoS_{ANNOUNCED}} * \frac{USERS_{CONNECTED}}{USERS_{COULD\_CONNECT}}$$

Formula 15. Network utility computation.

Where $QoS_{RECEIVED}$ is the real quality of service given by the service provider, $QoS_{ANNOUNCED}$ is the quality of service that the provider offers to give to a given service requester, $USERS_{CONNECTED}$ are the amount of users that have actually connected to a service provider and $USERS_{COULD\_CONNECT}$ are the amount of users that potentially could have connected to a service provider regardless of whether they did so or not.

The scenarios used for the simulations are described in the following section.

Scenarios and Results

In each scenario there are always 20 service requesting nodes and 10 service provider nodes. The trust threshold is always fixed at 0.5, which is a neutral value. Each scenario runs with each different metric with the same path followed by the users the first time the scenario is run. The amount of malicious providers and requesters varies from scenario to scenario. In each scenario the simulation is carried out using all the available metrics, one at a time, and also in the case of Metric 3, for different adaptation rates and adaptation steps of dispositional trust (refer to Section 4.3.1).

After carrying out the simulation experiments, we have obtained four sets of results, one for each of the simulation scenarios. Each set contains four pairs of graphs. From each pair of graphs, the leftmost one displays the amount requesters using malicious providers at any given time, and the rightmost shows the average QoS delivered by the system (in green) and the network utility (in blue).

First Scenario

- Requesters: 20, Malicious Requesters: 0;
- Providers: 10, Malicious Providers: 0;
- 1 simulation run with Metric 1;
- 1 simulation run with Metric 2;
- 2 simulation runs with Metric3:
    - 1 with A_R = 4 and A_S = 0.2 (slow adaptation)
    - 1 with A_R = 2 and A_S = 0.4 (fast adaptation)
- The results of the simulation can be found in **Figure 13**, **Figure 14** and **Figure 15**.

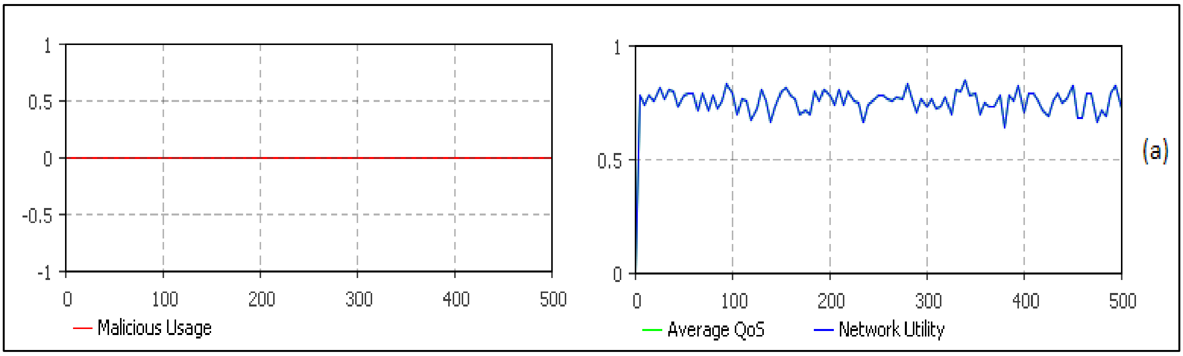


Figure 13. Scenario 1 results with Metric 1.

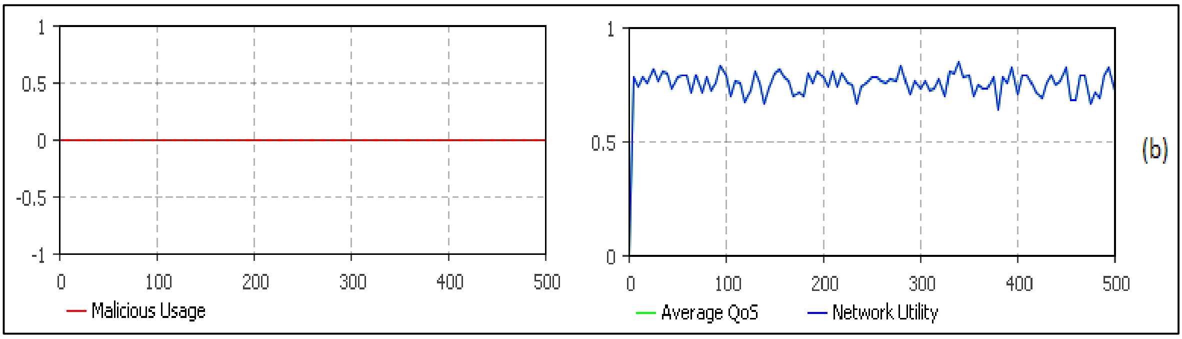


Figure 14. Scenario 1 results with Metric 2.

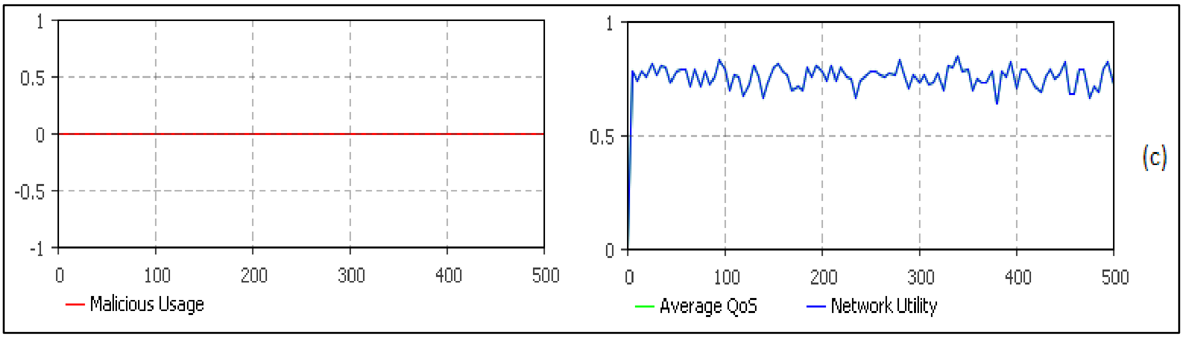


Figure 15. Scenario 1 results with Metric 3.

As can be seen, in the first scenario all the pairs of graphs are equal one to the other. Moreover, there are only three pairs instead of four as in the rest of the sets. This is because a) When all the requesters and providers in the system are well-behaved, the three metrics behave exactly the same, as all the providers will have a trust value computed bigger than the trust threshold and b) because the two runs for the third metric yield the same graph, thus they have been combined into one. Also, the average QoS is equivalent to the network utility as all the service providers are well behaved thus giving the same QoS they advertise and all the users that actually could connect to a service provider do so. Hence the green and blue graphs overlap.

So far, the first scenario doesn't provide any basis to discriminate if our metric is better or not than the other two as can be seen in **Figure 13**, **Figure 14** and **Figure 15**.

Second Scenario

- Requesters: 20, Malicious Requesters: 6;
- Providers: 10, Malicious Providers: 3;
- 1 simulation run with Metric 1;
- 1 simulation run with Metric 2;
- 2 simulation runs with Metric3:
  - 1 with A_R = 4 and A_S = 0.2 (slow adaptation)
  - 1 with A_R = 2 and A_S = 0.4 (fast adaptation)
- The results of the simulation can be found in **Figure 16**, **Figure 17**, **Figure 18** and **Figure 19**.

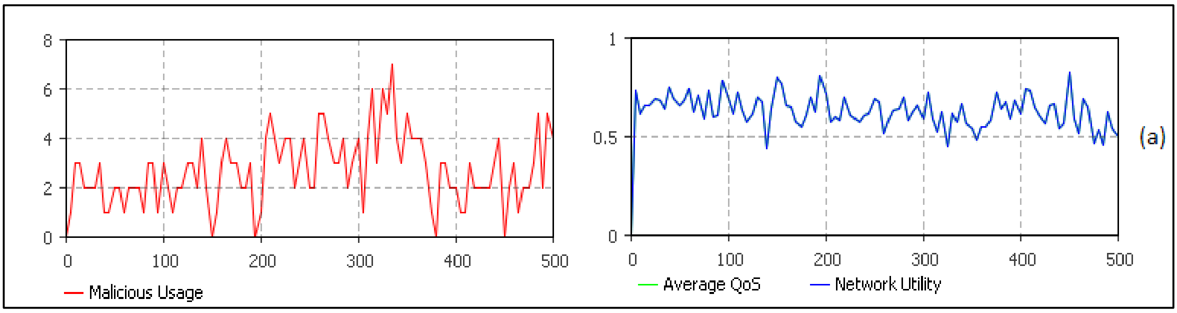


Figure 16. Scenario 2 results with Metric 1.

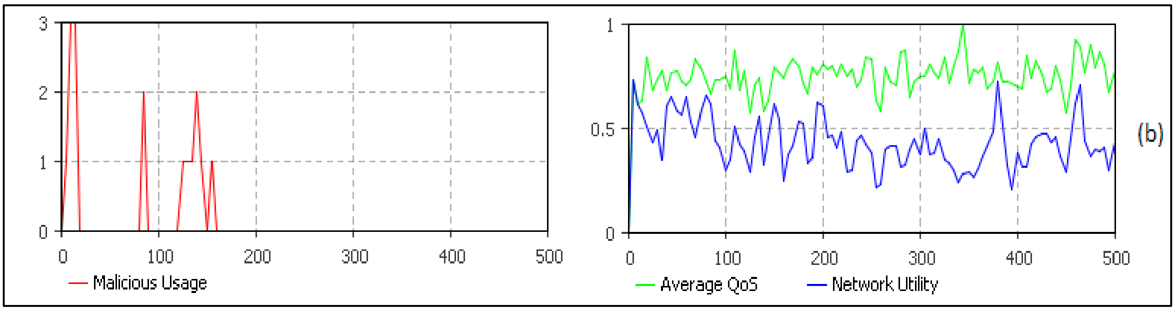


Figure 17. Scenario 2 results with Metric 2.

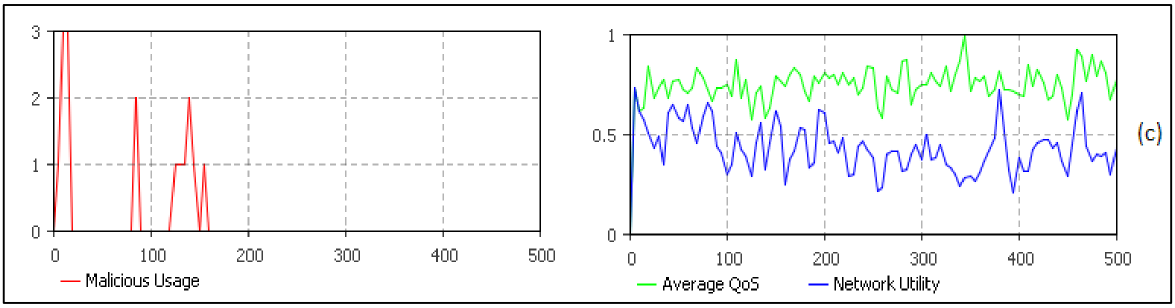


Figure 18. Scenario 2 results with Metric 3 with $A_R = 4$ and $A_S = 0.2$.

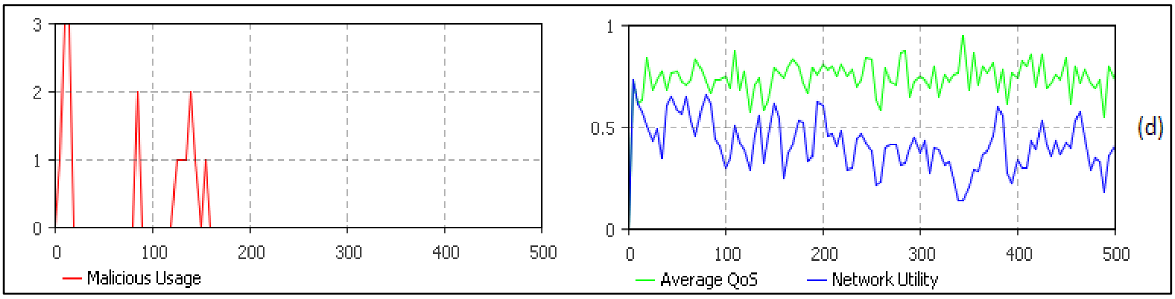


Figure 19. Scenario 2 results with Metric 3 with $A_R = 2$ and $A_S = 0.4$.

The second scenario includes already some malicious requesters and providers. In this scenario, we can already see clearly that the first metric, based solely on the distance to providers doesn't protect requesters in the system properly, as throughout all the simulation time there are requesters connected to malicious providers, as can be seen in **Figure 16**. The second and third metric, as can be seen, perform exactly the same regarding the amount of requesters connected to malicious providers. This happens because as there are not many malicious nodes in the system, the adaptation rate and step are seldom used, as just by the mere recommendations of other requesters and by direct observations, most of the malicious providers are avoided already. From this scenario, we conclude that D_T adaptation has little or no impact in scenarios where there are not many malicious nodes, our metric performing the same as a normal metric with direct observations and recommendations does. This can be seen in **Figure 17**, **Figure 18** and **Figure 19**. In regards to network utility and QoS, we can see that the Metric 1 provides a better network utility, as all the requesters are connected if they have the possibility no matter what, but metric 2 and 3 provide a better QoS in average for the nodes connected, as they avoid effectively requesters from connecting to malicious providers.

Third Scenario

- Requesters: 20, Malicious Requesters: 12;
- Providers: 10, Malicious Providers: 6;
- 1 simulation run with Metric 1;
- 1 simulation run with Metric 2;
- 2 simulation runs with Metric3:
    - 1 with $A_R = 4$ and $A_S = 0.2$ (slow adaptation)

  - 1 with $A_R = 2$ and $A_S = 0.4$ (fast adaptation)
- The results of the simulation can be found in **Figure 20**, **Figure 21**, **Figure 22** and **Figure 23**.

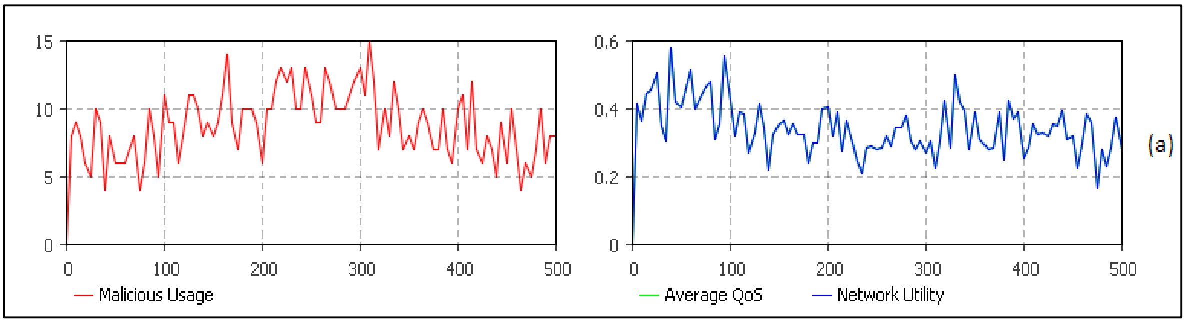


Figure 20. Scenario 3 results with Metric 1.

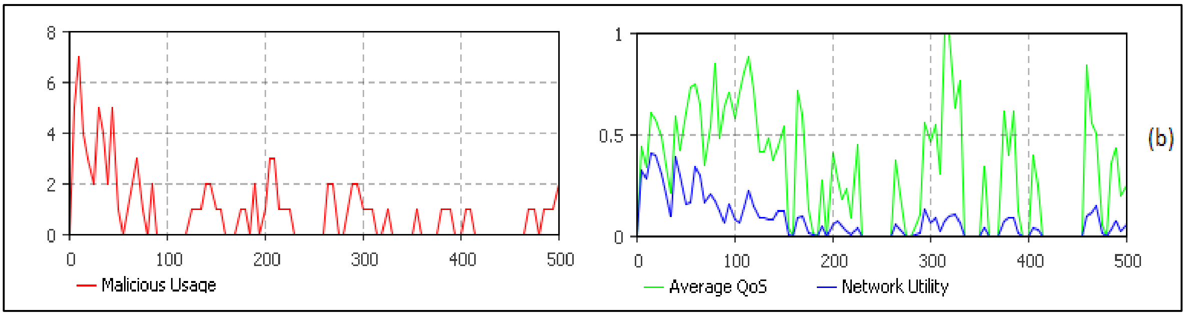


Figure 21. Scenario 3 results with Metric 2.

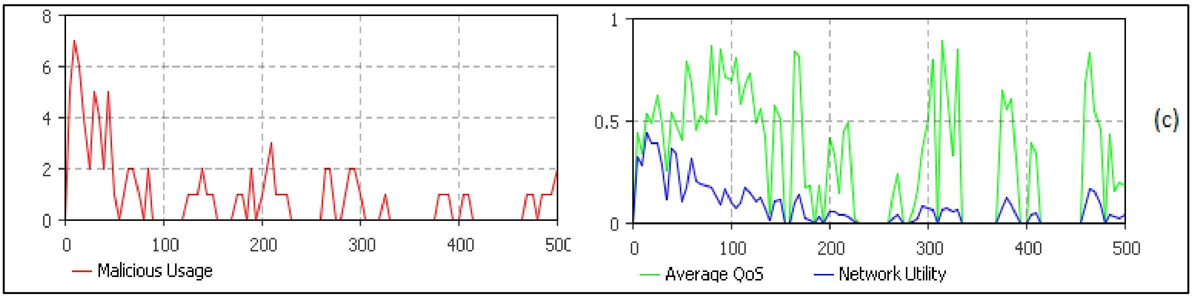


Figure 22. Scenario 3 results with Metric 3 with $A_R = 4$ and $A_S = 0.2$.

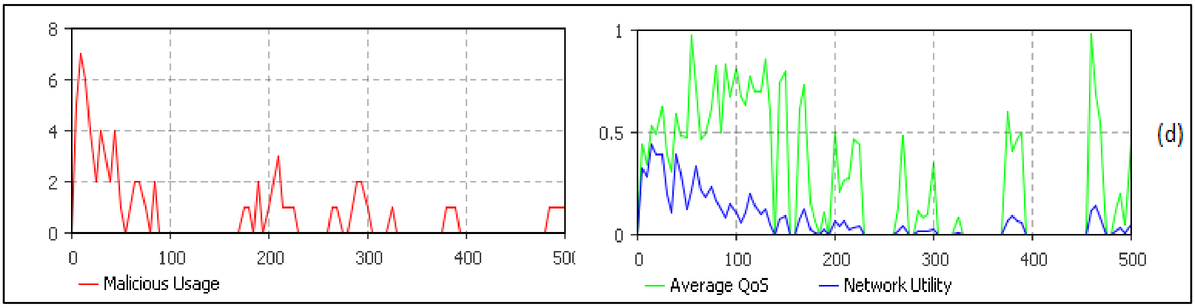


Figure 23. Scenario 3 results with Metric 3 with A_R = 2 and A_S = 0.4.

The third scenario has already more than a half of malicious requesters and providers. As can be seen in **Figure 20**, metric one performs poorly in regards of avoiding malicious nodes as expected. Metric 2, as shown in **Figure 21**, copes pretty well concerning the avoidance of malicious providers, even though there are still some requesters using them throughout all the simulation time. Lastly, metric 3 with a slow adaptation rate, displayed in **Figure 22** manages to cut off some of those malicious connections and with a fast adaptation rate, shown in **Figure 23**, performs even better avoiding even more requesters to connect to malicious providers. It can be effectively seen that D_T adaptation already plays a role in environments where more than half of the nodes are malicious, performing better than the other two metrics. Concerning QoS and network utility, it is obvious that a more aggressive metric will avoid more malicious connections, which even though they provide a lower QoS they provide some, impacting directly the network utility as the usage of providers gets diminished. In the other hand, it can be seen in that the average QoS provided is higher when a requester is connected, this resulting from the avoidance of malicious providers which provide lower QoS.

Fourth Scenario

- Requesters: 20, Malicious Requesters: 20;
- Providers: 10, Malicious Providers: 10;
- 1 simulation run with Metric 1;
- 1 simulation run with Metric 2;
- 2 simulation runs with Metric3:
    - 1 with A_R = 4 and A_S = 0.2 (slow adaptation)
    - 1 with A_R = 2 and A_S = 0.4 (fast adaptation)
- The results of the simulation can be found in **Figure 24**, **Figure 25**, **Figure 26** and **Figure 27**.

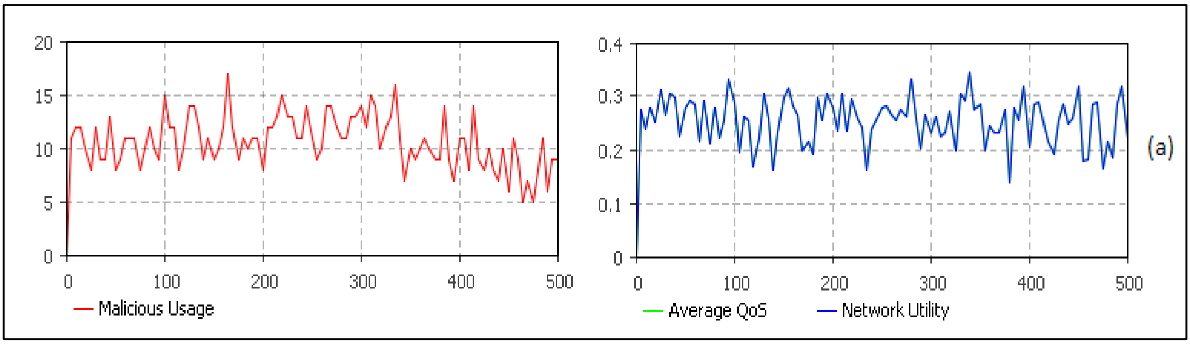


Figure 24. Scenario 4 results with Metric 1.

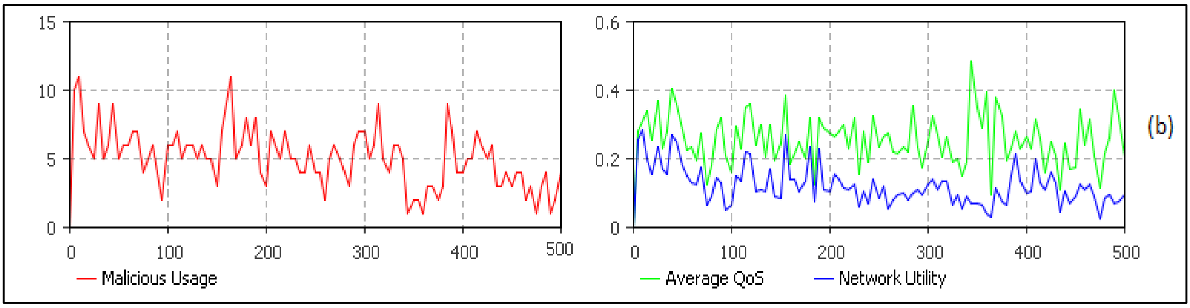


Figure 25. Scenario 4 results with Metric 2.

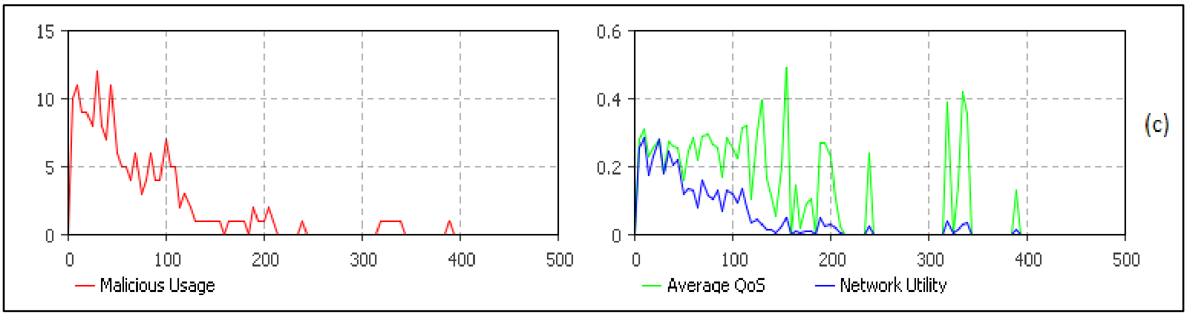


Figure 26. Scenario 4 results with Metric 3 with A_R = 4 and A_S = 0.2.

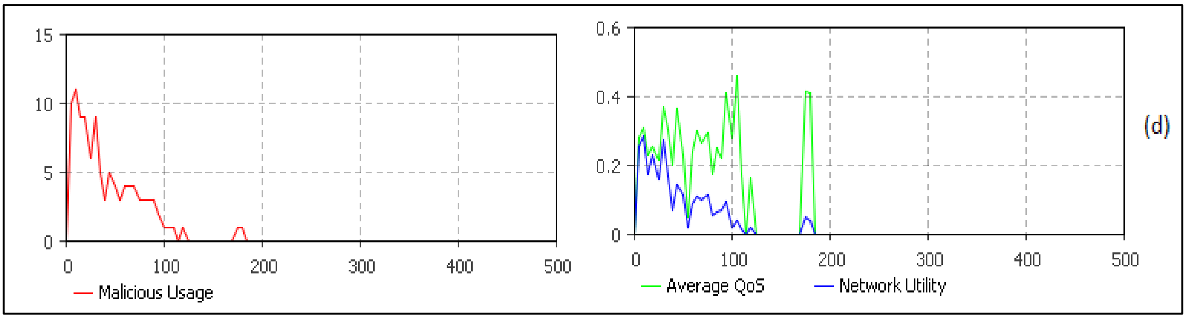


Figure 27. Scenario 4 results with Metric 3 with A_R = 2 and A_S = 0.4.

Finally, in the fourth scenario we present an (unlikely) extreme situation, where all the nodes in the system (requesters and providers) are malicious. This scenario is not likely to happen in any UCN environment, but it still is a valid test case in order to understand how trust metrics perform under extreme adverse conditions. As displayed in **Figure 24**, metric 1 performs very poorly in terms of impeding requesters to connect to malicious providers as expected already from the previous simulation scenario. Metric 2 performs better than metric one as could be anticipated, but in such hostile environments it is incapable to cope and to converge to accurate trust values as shown in **Figure 25**, as also all requesters are lying when giving recommendations. In the other hand, metric 3 displayed in **Figure 26** with a slow adaptation rate performs already much better, as the bad interactions also diminish the impact of bad recommendations, resulting in a noticeable drop in the amount of requesters connecting to malicious providers over the simulation time. Metric 3 with a fast adaptation rate performs even better and before half of the simulation time has elapsed it manages already to prevent any requester from connecting to malicious providers, proving very effective in making the trust metric to converge to accurate trust computation values. This can be seen in **Figure 27**. Regarding QoS and network usability, as metric 3 effectively prevents from malicious connections, it also impacts network utility, as in an environment where all providers are malicious, there will not be requesters using the network at all.

Results Analysis

As can be seen from all the previous graphs resulting from each of the simulation scenarios, dispositional trust adaptation presents a trade-off; in one hand, it proves very effective the more hostile the environment is, effectively preventing requesters from connecting to harmful providers, which are easily detected as they provide a lower QoS than advertised. In the other hand, it also decreases extensively the network utility, as it prevents malicious connections, rendering the network unusable when all the providers are malicious.

This said, the latest is very unlikely to happen in a real UCN, and if it was to happen our metric will actually prevent users from connecting to other nodes that potentially could result in a harmful interaction. In these simulations we have defined a malicious provider as the one who provides less QoS than announced, which in principle is not a harmful interaction unless the service to be used is of a critical importance and it becomes unusable under certain QoS threshold. In general, a malicious provider could be defined by any other parameter or action, which actually could be of a real harm for a requester if to ever interact with such a provider. It is our belief that dispositional trust adaptation can help anticipating malicious nodes in highly hostile environments and that, all in all, it is an effective measure in order to protect nodes from such interactions.

5.2.3 Cooperation Incentives Results

In order to test the robustness of our trust metric and the effectiveness of our cooperation incentives schema, we have created a simulation experiment and we have compared our metric to the one of Koutrouli et al. [2], which also is used in P2P environments and which also introduces the notion of incentives. In this metric, the authors provide a crediting mechanism in order to incentivize the fairness of the recommendations made by the users of the P2P service or system. Users with better reputation can buy recommendations at a lower cost, and get paid more for their recommendations. Meanwhile, users with a bad reputation should pay a higher amount of credits when buying recommendations but they receive a lower amount, if any, when being asked for recommendations, eventually running out of credits and being excluded of the system. The computation of the reputation of a given user is done in a traditional fashion, taking into account direct observations and third party recommendations.

Scenario

The simulation scenario is a populated area of an imaginary city. There are several streets on which users are randomly placed each time the simulation is run. All agents acting as clients and local users move randomly, but with a pre-set random seed so that the paths followed by them are always the same for each simulation run. When they are in range of one or more hotspot, be it a fixed one in a shop or restaurant or a mobile one carried by a local user, they will try to connect to them. For this purpose, both parties involved will compute the trust level of each other and will also calculate the cost-benefit function of behaving either rightly or wrongly according to the given incentives.

In every simulation run, there are 20 agents acting as clients (foreigners roaming), 10 agents acting as fixed hotspots and 10 acting as mobile hotspots (local users). In order to test whether the trust metric and the cooperation incentives are effective in this kind of scenario, we have assumed a very high amount of potentially malicious agents (agents prone to act in self-interest and to display selfish behaviours), accounting for the 70% of the total agents.

Results Analysis

We have plotted for each of the metrics two graphs, one regarding the effectiveness of the trust metric itself, and the other concerning the mitigating effects of the cooperation incentives. The results shown by the graphs are averages for 50 runs of the same experiment with each metric.

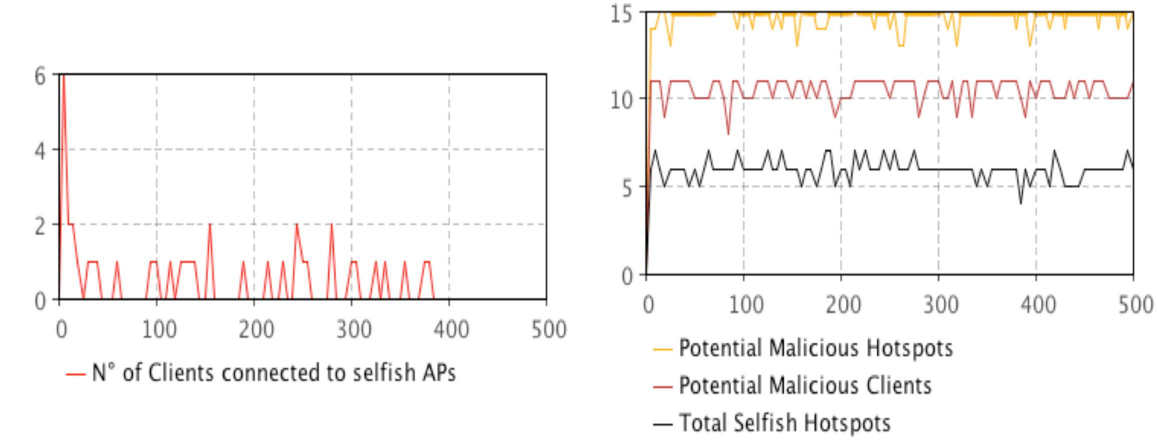


Figure 28. Our metric results.

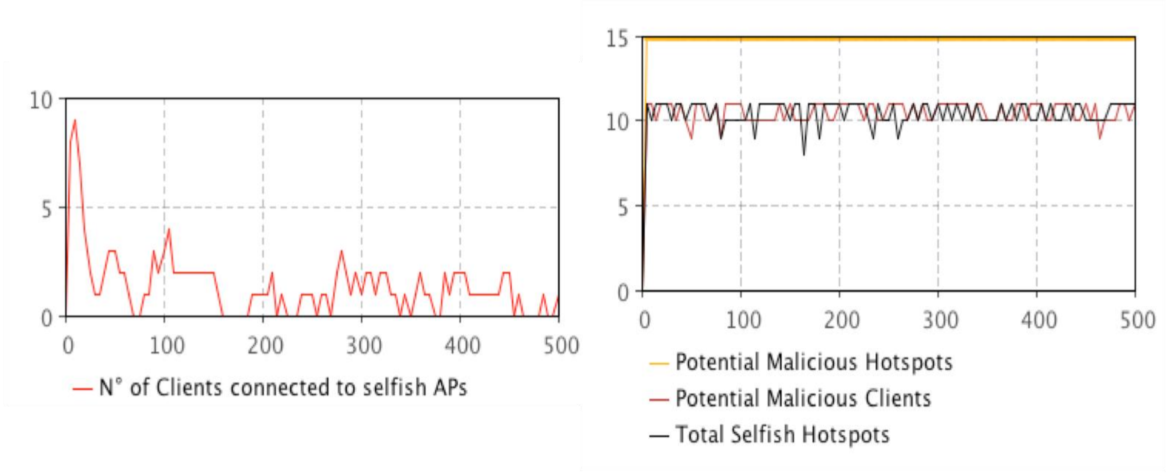


Figure 29. Koutrouli et al. "*Credible Recommendations*" [2] results.

As can be seen from **Figure 28** and **Figure 29**, our trust metric proves to be more effective when dealing with selfish (non-trustful) users as the simulation time evolves. With our metric, we achieve that even though selfish nodes are present in the system, no client will connect to them. In the other hand, with *Credible Recommendations* clients are connecting to selfish nodes during all the simulation, even though the numbers are quite minimized towards the end.

Regarding the cooperation incentives, with our system, only up to 33% of the potential malicious hotspots actually behaves selfishly at certain points of time, while with *Credible Recommendations*, this value goes up to a 70% of selfish nodes in some cases. This is due to the fact that our incentives are focusing on the uppermost level of the system's functionality, which is sharing and getting connectivity, making it worthy to behave correctly as it comprises a bigger benefit, while the compared metric's incentives are targeting a mid-layer of the system, which are the recommendations, not making it worthy as much of the time to be well behaved, as not that big benefit is gained from it.

5.2.4 Framework Feasibility and Survivability Simulation

This last simulation deals with the framework as a whole, with all its elements in place, in order to evaluate its feasibility from different perspectives. Firstly, we aim to evaluate whether the required time to bootstrap the framework is reasonable enough to make it usable. Secondly, from a survivability point of view we evaluate whether data depletion, users running out of available monthly data in order to share services with other users, can impact our framework and how long does it take for a user to be depleted of data.

Scenario

The simulation environment corresponds to a real world area, which is the airport of the city of Geneva, Switzerland. The environment has been modelled respecting the real dimensions of the airport, and also the real proportions of both local and foreign travellers and permanent workforce of the airport [70]. The exact details of the simulation are as follows:

- 450 meters long and 150 meters wide, spanning 3 floors of this same size
- Around 13 million passengers in 2012, from which 55% are foreigners and 45% are locals.
- 840 staff and permanent workers (working in shifts).

Taking into account this previous data, each of the simulation runs has been done with 3000 agents, which simulate passengers (both local and foreign in the proportions previously mentioned) and 280 workers (assumed always locals) at any time, included in those numbers. To make the scenario as realistic as possible, agent renewal happens with a normal distribution with an average of 2 hours in order to simulate the passengers leaving and new ones arriving. Workers are also renewed in 8 hour shifts. We assume that locals have an average of 15-20 friends (acquaintances or previously interacted users) and foreigners an average of 2. All local workers are known to each other.

In order to study the feasibility of the system, we have run several simulations each with a different user base for the system. This user base is a key point, as it will determine the threshold from which the system might be usable both from the bootstrapping point of view and from incentives perspective.

Note that when we talk about user base, we are not talking about the amount of agents in the simulation, which are fixed according to the criteria mentioned in the previous section, but to the total amount of users in the world using this system. This user base is what enables the probabilities of finding long FOAF chains in order to enhance the cooperation incentives provided by Trust Transfer.

Each simulation runs for a real-world whole day, measured in seconds (86400 seconds)

Bootstrapping Results

For the system to be usable, the bootstrapping time needs to be as low as possible in order for the foreigner passengers to be able to connect to locals while in their short time at the airport.

We consider that the system is bootstrapped if half of the agents that can provide connectivity have at least shared once their Wi-Fi with a foreign, or a local that might have run out of data. For each of the graphs presented below, the Y axis represents the amount of agents and the X axis the simulation time, measured in seconds.

We have run the simulation for different sizes of user base population, ranging from 2 million system users to 200 million system users with an intermediate simulation accounting for a 20 million system user base. The results can be seen in **Figure 30**, **Figure 31** and **Figure 32**.

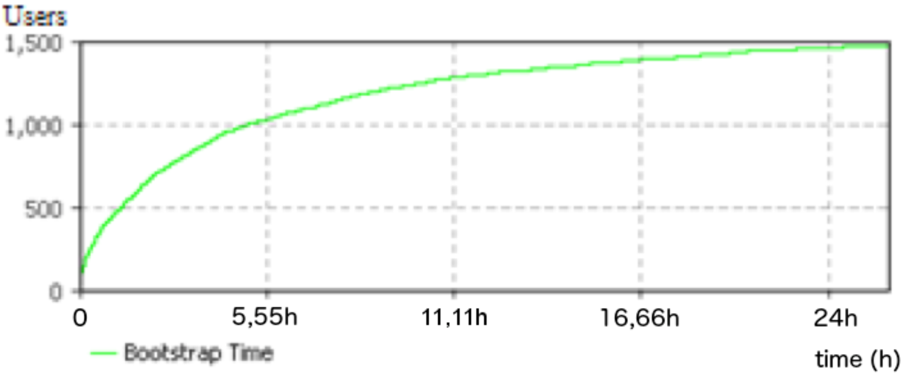


Figure 30. Bootstrap time with 200 million system users worldwide.

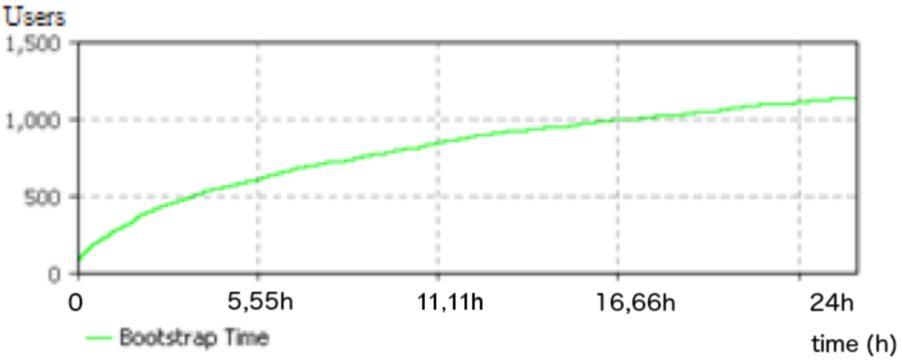


Figure 31. Bootstrap time with 20 million system users worldwide.

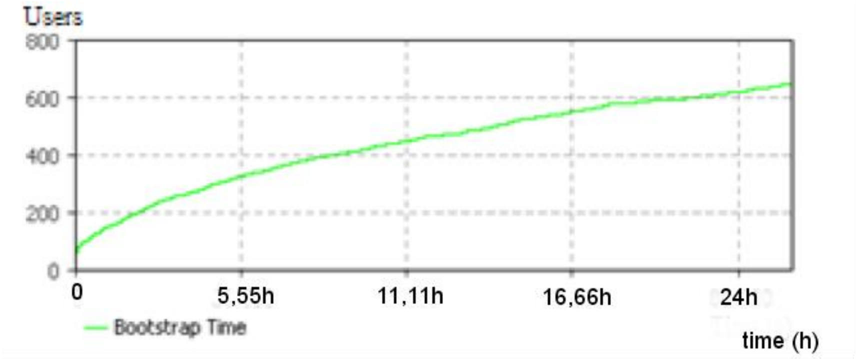


Figure 32. Bootstrap time with 2 million system users worldwide.

As can be seen from the results, if we want to achieve the aforementioned objective of half the agents having shared their Wi-Fi with foreigners in a reasonable time, the best configuration achieving this is the one with 200 million system users, even though with a 2 million system users base we can still achieve a proper time for the system to be bootstrapped. This accounts for 750 agents in roughly 7,500 to 8,000 seconds, which is close to the average time for agent renewal in the simulation, making it a feasible time for the system to be bootstrapped.

Data Depletion Results

Another interesting measurement for us is how quick users run out of data capacity, i.e. they go over the amount of data their telecomm company allows for their contract, and which is the average time that it takes for a given user to be depleted of her data capacity, as this has a direct relation with the survivability of the system. If users run out of data to quickly, the system would became unusable and thus there would be a vast amount of nodes that could not request any service from any service providing node.

We have run the simulation for different sizes of user base population, ranging from 2 million system users to 200 million system users with an intermediate simulation accounting for a 20 million system user base. For each of the figures, the left-hand graph represents the amount of users that experience data depletion in a given point of time, being the Y axis the amount of users and the X axis the time in seconds, and the right-hand graph represents the average time that took for those users to be depleted of their data capacity, measured in seconds. The results can be seen in **Figure 33**, **Figure 34** and **Figure 35**.

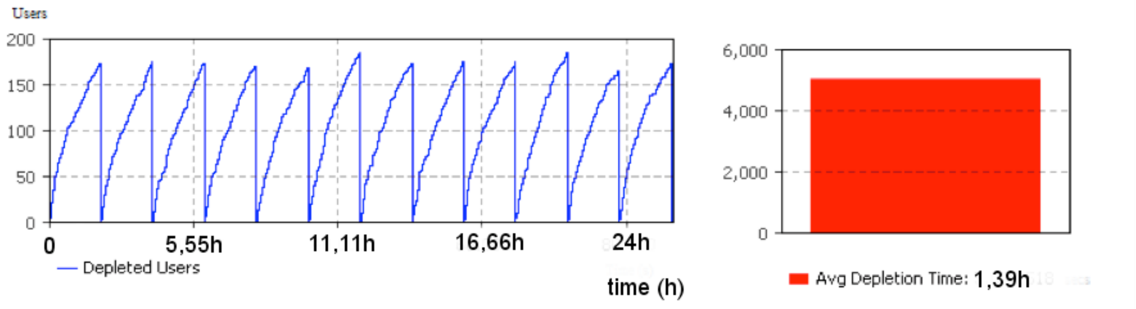


Figure 33. Amount of depleted users and average depletion time with 200 million system users worldwide.

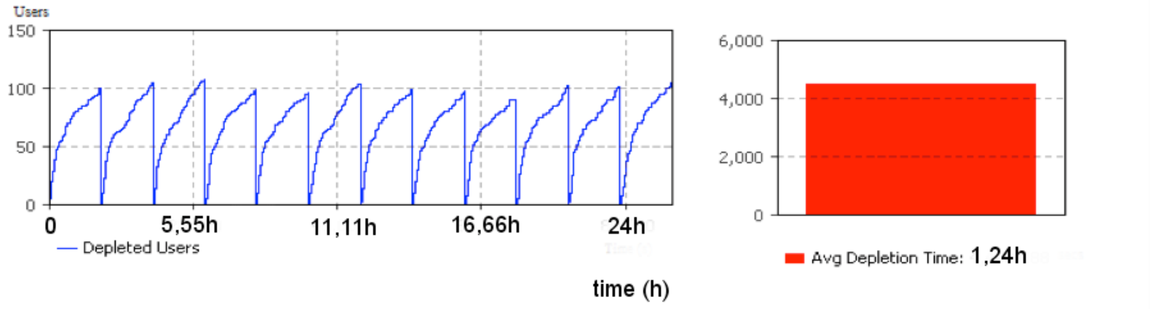


Figure 34. Amount of depleted users and average depletion time with 20 million system users worldwide.

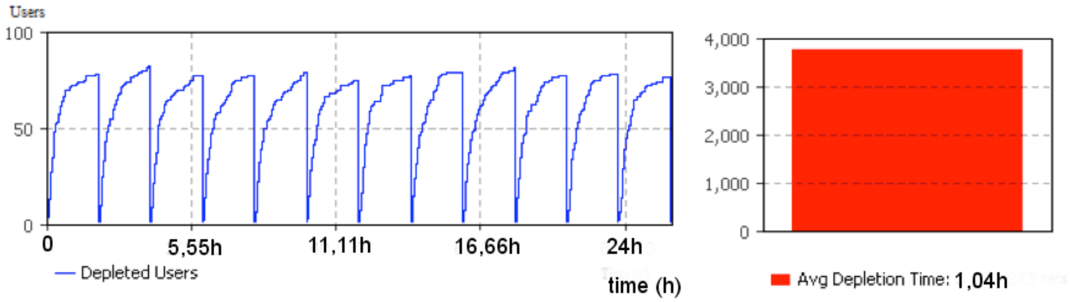


Figure 35. Amount of depleted users and average depletion time with 2 million system users worldwide.

As can be seen from the results, with a smaller system user base the amount of depleted users in each renewal period is also smaller, while the average depletion time for each of those agents is lower as well. The implications of this will be discussed in the next section.

Results Analysis

From the previous simulation runs, we can summarize the results in **Table 3** as follows:

Table 3. Summary of results

User Base (in millions)	**Bootstrap time (in hours)**	**Number of depleted users per renewal period**	**Average time before depletion (in hours)**
200	2.26	175-185	1.39
20	7.87	95-105	1.24
2	> 24	75-85	1.04

As can be seen from the summary in the previous table, the bigger the system user base, the better the results, both in terms of bootstrapping time and depletion measurements.

Regarding bootstrapping results, with a bigger user base it is more probable to find a chain connecting a service requester with a service provider, accounting for the shorter bootstrapping time, as it is more likely to find users who can transfer some trust points from one end to the other and thus enabling cooperation in between the two users. It is also worth to note that with the use of the system the probabilities of finding users from which to get points increases as the interactions in between agents increase. This translates into an increase of the probabilities of finding a chain of agents from which to get points lent from one end to the other by 0.1% per interaction per agent. Arguably, it could be said that a 20 million user base could be enough to obtain a reasonable bootstrapping time (~7.8 hours), but with a user base closer to 200 million we can achieve times which are closer to the agent renewal time in our scenario, making it closer to the ideal situation.

Regarding data depletion, as true as it is that with smaller number of users amounts there are less agents that get depleted from their daily quota allowance, this is due to the fact that also there are less agents being able to connect and to allow connections in order to share Wi-Fi as it is more difficult to find a longer user chain to transfer trust points. In the other hand, it can also be seen that the average time taken to deplete a user from her daily data quota is higher the bigger the user base is, meaning that even though more users are depleted in each agent renewal period, those users take longer to be depleted due to the higher amount of agents being able to share their Wi-Fi connection. It is also worth to note that even with a higher number of depleted users, those account only for ~10% approximately of the total amount of agents being able to share their Wi-Fi connectivity (175-185 out of 1500).

5.3 Performance Results in Real Tests

In this section we present the performance results obtained from testing a prototype Android application both in a dynamic scenario such as ski slopes and in a static scenario such as a cafeteria, plus some preliminary tests about Android portable hotspot connectivity range.

This tests were of crucial importance in order to assess the feasibility of our system in the real world and to check how good it performs and which are the performance indicators of it.

5.3.1 Client and Hotspot Measurement Results

In order to determine the required parameters for both the simulation and the real application, we have carried out [18] a few simple tests to determine the range of a portable hotspot or access point (AP) such as the one present in Android phones. **Table 4** shows the results from these first tests.

Table 4. Preliminary client-to-hotspot distances and signal strength.

Distance to AP (m)	RSSI
5	-29
10	-57
15	-57
20	-57
25	-57
30	-57
35	-57
40	-57
45	-65
50	-62
55	-57

75	-70
100	-74

As shown in the table, we have tested how the signal strength of the portable hotspot evolves for a set of distances, ranging from 5 to 100 meters, being lower values a better signal strength. As can be seen, the coverage area of such a device goes well up to 100 meters, even though the signal strength is already too low in order to achieve a meaningful and reliable data transmission. Thus, according to this results, we have considered for both the simulation experiment and the real application tests a range of 40 meters maximum, as we think it is a realistic approach on what the capabilities of a portable hotspot are.

5.3.2 Ski Slope Test Results

For these tests, we had four users with our android application implementing the connectivity sharing service installed. They were split into two groups of two, where one user will be sharing while the other will be connecting to the service. They would try to ski fairly close to each other for the duration of the tests.

In order to perform the tests on the ski slopes [18], we first divided them into two sets of tests, carried out in two consecutive days. The results of the first day of testing can be seen in **Table 5**, while the results from **Table 6** correspond to the second day of testing.

In the first day of testing, we focused more actively in testing the overall performance and response of the application than in performing intensive data consuming operations (heavy download, HD video streaming, etc.), which was left for the second testing day. As can be seen in Table II we underestimated slightly the connection setup time in the simulation, being sometimes higher than the previously 50 seconds used. Nevertheless, we obtained good results while skiing inside the appropriate distance which the hotspot covers, achieving connections lasting up to 5 minutes and a good amount of data download and upload.

The short lived connections present in the **Table 5** account for the cases where either the connection setup phase broke due to surpassing the adequate distance in between the skiers or due to the local skier not having mobile data connectivity at the moment, or due to automated sharing protection mechanisms not being established properly during the setup phase, or due to HSDPA to 3G failover or vice versa.

Table 5. Upload, download, connection setup time and connected time in the ski slopes, 1st day.

Data upload (Bytes)	Data download (Bytes)	Connection Setup	Connected Time
634744	3719567	1 min, 6 sec	5 min, 18 sec
0	10171	1 min, 24 sec	0 min, 6 sec
641559	11364516	0 min, 40 sec	5 min, 59 sec

0	0	0 min, 55 sec	0 min, 0 sec
203325	2287543	1 min, 40 sec	0 min, 59 sec
6338	24506	0 min, 50 sec	0 min, 55 sec
144730	1151956	0 min, 50 sec	2 min, 49 sec
889	13057	0 min, 54 sec	0 min, 37 sec
1131749	39404562	0 min, 50 sec	4 min, 57 sec
0	3099	0 min, 40 sec	0 min, 19 sec
0	0	1 min, 5 sec	0 min, 6 sec
10580	28864	0 min, 42 sec	1 min, 9 sec
0	72	1 min, 14 sec	0 min, 6 sec
80058	1470702	0 min, 49 sec	0 min, 47 sec
210412	6311026	0 min, 44 sec	0 min, 55 sec

We dedicated the second day of testing to perform more controlled and more data intensive experiments, trying to stay at all times inside the appropriate range to the local skier while performing data consuming tasks such as HD streaming and the like. We aimed to maintain long lived connections (as long lived as can be while in a ski slope and a ski lift) to see how the application would perform, even though we still got some short or non-existent connections due to the facts mentioned previously for the first testing day. The results can be seen in **Table 6**.

Table 6. Upload, download, connection setup time and connected time in the ski slopes, 2nd day.

Data upload (Bytes)	Data download (Bytes)	Connection Setup	Connected Time
0	11215	0 min, 41	0 min, 6 sec
713780	24391110	0 min, 47 sec	7 min, 56 sec
3161848	102125792	1 min, 18 sec	13 min, 18 sec
1234	9215	0 min, 45	0 min, 8 sec
644697	22112428	0 min, 44 sec	6 min, 39 sec
0	0	0 min, 56 sec	0 min, 0 sec
148083	1547139	1 min, 32 sec	29 min, 18 sec

As shown in the table, we successfully achieved to maintain quite long connections up to almost 30 minutes, while skiing and in the ski lifts. Also, it was possible to upload and download a good amount of data without further problem while being connected, streaming HD video and browsing internet.

5.3.3 Static Environment Results

To finalize the set of tests, we carried out a session in a less dynamic place than the ski slopes. We tested the application in less demanding and more static conditions by performing some tests inside a cafeteria of the ski resort while not moving any of the terminals at all. This test involved two users sitting nearby in the cafeteria, one sharing and one connecting to the service. The results of these last set of tests can be seen in **Table 7**.

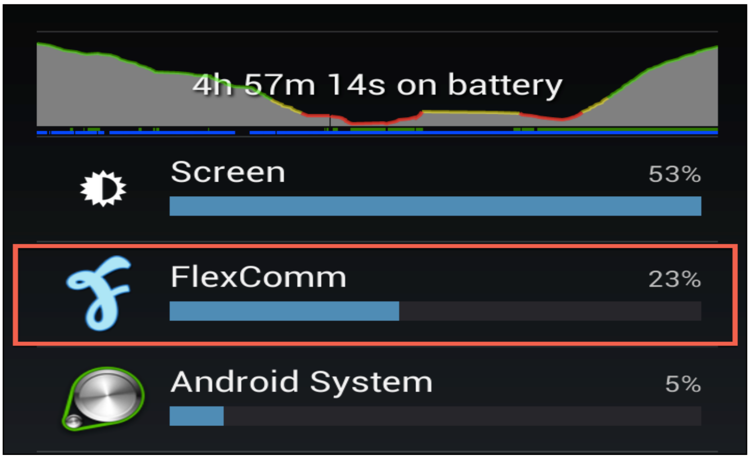
Table 7. Upload, download, connection setup time and connected time in a cafeteria.

Data upload (Bytes)	Data download (Bytes)	Connection Setup	Connected Time
551012	2611660	0 min, 46 sec	11 min, 40 sec
590794	2528282	0 min, 46 sec	8 min, 27 sec
0	5460	0 min, 53 sec	0 min, 2 sec
468920	1916792	0 min, 47 sec	4 min, 25 sec
1227882	4608589	0 min, 52 sec	6 min, 16 sec
0	3465	0 min, 59 sec	0 min, 0 sec
1321702	3829081	1 min, 30 sec	21 min, 27 sec
1118089	3206335	0 min, 57 sec	55 min, 30 sec
265781	1844596	0 min, 48 sec	19 min, 7 sec

As displayed in the table, we were able to achieve long lived connections, up to 55 minutes without it breaking, and successfully doing light browsing and casual social network and app use. Again, some of the shorter or broken connections account for the cases where either the local phone lost connectivity or switched from one type of network to another, deeming impossible to perform the appropriate steps in order to protect the user sharing her or his mobile connection, or to establish a connection at all.

5.3.4 Battery and CPU Usage Results

In order to assess the performance of the application regarding battery consumption and CPU usage, we measured those values using the built-in functionality to monitor per application battery and CPU usage that can be found in any Android phone. High battery consumption values could deter users from adopting our application in the future, and thus were of a high concern for us.



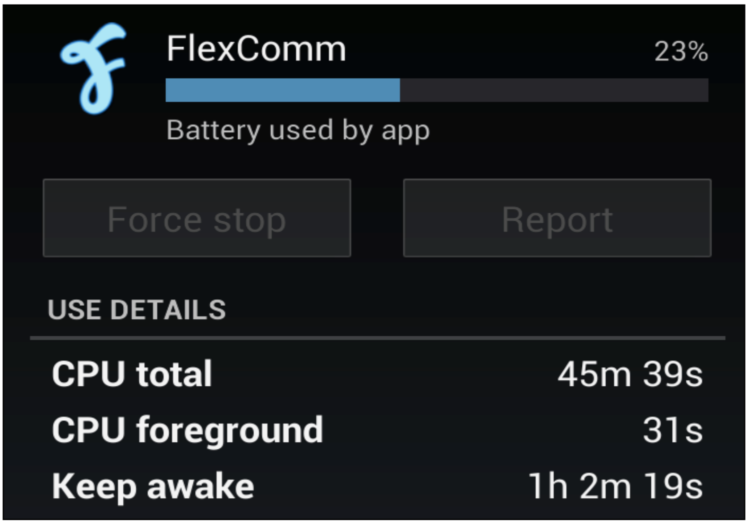
4h 57m 14s on battery

Screen 53%

FlexComm 23%

Android System 5%

Figure 36. Percentage of battery consumed by the Android application service.



FlexComm 23%

Battery used by app

USE DETAILS

CPU total 45m 39s

CPU foreground 31s

Keep awake 1h 2m 19s

Figure 37. CPU statistics for the Android application.

As can be seen in **Figure 36** and **Figure 37**, the battery usage during the testing sessions falls inside an acceptable range, accounting for the 23% of the total battery use, while the CPU usage both while performing active operations and just in a keep awake status are also inside acceptable values. This said, the battery usage level could and should be improved not to impact the overall user experience and this is one of the points we will work in following versions of the application.

5.4 Result Synthesis by Category

In order to give a better overview and comprehension of the results achieved in this section, we present following a summary of the results grouped by category.

5.4.1 Functionality

Regarding functionality, we have three main functional areas on our decentralized framework, namely trust management, cooperation incentives and survivability. Besides those, an inherent functionality of our framework is service sharing in the form of a Wi-Fi connectivity sharing service.

In general, from a trust management point of view and its subcomponents, we can see that dispositional trust increases the attack resistance of the trust manager overall as shown in Section 5.2.2. Trust transfer as a metric is used in the simulations concerning bootstrapping and data depletion on Section 5.2.4.

Regarding cooperation incentives, our cooperation approach in the form of points and rewards is implemented and simulated in Section 5.2.3 and the FoaF functionality of the cooperation manager is again implemented in the simulations in Section 5.2.4.

Concerning survivability, the bootstrapping and data depletion simulations in Section 5.2.4 cover this functional area results. Finally the last functionality which is the Wi-Fi sharing service itself it is covered in Sections 5.2.1 and 5.3.

All these simulations and its results prove that each of the functionalities work and serves its purpose in the overall framework. All the results from the previous subsections of this Chapter are related to those functional areas and they are discussed following classified into more concrete characteristics.

5.4.2 Attack Resistance

Regarding attack resistance, first and foremost our decentralized framework is attack resistant to Sybil attacks by the virtue of using Trust Transfer as the trust metric of choice. This fact has not been proven by simulation, as it has been extensively demonstrated in the work of Seigneur [1].

The other two methods that we use in our framework to increase its attack resistance against general node misbehaviour, i.e. not providing the requested service, cheating and the like, are dispositional trust adaptation and cooperation incentives. In one hand, as we have demonstrated in the simulations, dispositional trust adaptation enables faster convergence of the trust values computed by a given metric to values that accurately represent the reality of the environment. This acts as a pre-emptive measure to avoid misbehaving nodes as explained in Section 5.2.2.

In the other hand, by introducing the adequate cooperation incentives in the form of rewards and points, misbehaving often results in a lower benefit than behaving properly, thus indirectly enforcing good behaviour via reward and deterring malicious nodes. This results in a lower amount of nodes that are willing to actively misbehave, as shown in the simulations in Section 5.2.3.

5.4.3 Usability

Regarding usability, we have proved that our framework is both viable from a bootstrapping point of view and from a data depletion point of view, which is also closely related with survivability.

From a bootstrapping point of view, our framework is able to be fully initialized in a time deemed reasonable to be usable as shown in the simulations on Section 5.2.4.

Another main concern for usability is the nodes which provide services running out of allowed monthly data usage, as many telecom operators place restrictions as has been evaluated in Section 2.3.3 and

Table 1. Our simulation results have proved that our framework is also usable despite some nodes reaching data depletion limits as explained in Section 5.2.4.

5.4.4 Performance

Regarding performance, we have tested our framework's performance both with simulations and with a real prototype implementation in Android of the Wi-Fi service sharing functionality.

Through simulation we have proved theoretically that our framework can work in highly dynamic environments such a ski slope in terms of achieving connectivity from a service requesting node to a service provider node as shown in Section 5.2.1.

Through the real prototype implementation we have demonstrated that such a service can really work both in highly dynamic environments and in static environments such as a cafeteria, both in terms of connectivity setup times and data transmission volume as well as from CPU usage and battery depletion points of view. These results are shown in Section 5.3.

Chapter 6. Conclusion and Future Work

In this Chapter that finalizes this Thesis, we first discuss and summarize the main contributions and conclusions of this thesis. Then we give a brief overview into the future work that still needs to be done and that can improve the present work.

6.1 Contributions Discussion

The aim of this section is to present an overview of the main results obtained in the previous Chapter 5 and to discuss them from a global perspective. To this end, we have summarized the most important results and characteristics of our framework in the following table.

Table 8. Overview of the main results.

Characteristic	Value
Sybil-Attack Resistance as Enforced by Trust Transfer [1]	✓[2]
Dispostional Trust adaptation effectiveness over time	~60-70% less nodes connecting to malicious nodes in very hostile environments[3]
Cooperation Incentives effectiveness over time	~40% improvement vs. *Credible Recommendations* [2]
Survivability Measures(measures to control data/battery/other resources depletion as explained in Section 2.3)	✓
Bootstrap time (biggest user base)	Allows the full system to be bootstrapped in 2.26 hours as explained in Section 5.2.4, being this enough for the system to be usable in real situations
Data Depletion time (biggest user base)	Data depletion for a given terminal in a given day takes approximately 1.39 hours in the worst case scenario, which is an acceptable time not to render the system unusable.
Average Connection Setup time (Android prototype)	Establishing a connection in between two devices takes in average 57 seconds, which is an acceptable time not to render the system unusable.

[2] For positive recommendations

[3] With the highest A_R and A_S

Maximum Connection time (Android prototype, dynamic situation)	Allows approximately a long lived connection of 29 minutes without interruption, which is more than enough to render the system functional.
Maximum Connection time (Android prototype, static situation)	Allows approximately a long lived connection of 55 minutes without interruption, which is more than enough to render the system functional.

Previously in this thesis, in Section 3.4 we identified the gap related with the analysis of the state of the art and we stated the desirable characteristics that a fully distributed user-centric network environment framework should possess. Namely, those characteristics were Sybil attack resistance, a solid Cooperation Incentives schema and survivability mechanisms in order to ensure that the nodes in the system do not run out of valuable resources such as battery life or data allowance.

The results listed in the previous table underline that our framework fills the identified gap as follows:

1. We have designed, developed and implemented a fully decentralized trust management and cooperation incentives framework for User-Centric Network (UCN) environments integrating the listed below contributions.
2. Our framework, by using Trust Transfer, is resistant to Sybil attacks and we have further investigated how Trust Transfer can be used as a solid schema of Cooperation Incentives using its trust points as rewards in order to incentivize collaboration in between service requesters and providers in the application domain of wireless UCN.
3. We have validated that the possibility of transferring points not only directly in between peers but also using friend-of-a-friend chains makes our schema much more attractive the bigger the system grows in user base. Our probabilistic study on chains of trust and friend of a friend (FOAF) chains in small-world network subsets shows that it improves the effectiveness of cooperation incentives in the form of points thanks to their possible propagation through the UCN.
4. We have implemented a large scale simulation of the framework using AnyLogic on real life environments (i.e. airport, train station, ski resort, etc…)
    a. Proving its feasibility and performance from:
        i. Bootstrapping point of view
        ii. Resource depletion point of view
    b. Leading to the comparison through simulation of the trust metric combined with a cooperation incentive schema against other related trust metric using AnyLogic simulation tool, which has never been done before for this metric, taking into account:
        i. Number of users connected to selfish or malicious APs

   ii. Number of potential malicious APs vs. those who actually turn to be selfish regardless of the offered incentives to behave correctly.

5. We have studied the effects of data depletion in our framework applied in this application domain of wireless UCN, proving that with the right user base and agent renewal rate, the rate in which nodes are depleted of their monthly data allowance is compatible with a survivable and sustainable systems.
6. We have introduced dispositional trust adaptation in order to make the framework even more resilient to malicious interactions, whose feasibility and performance have been evaluated with agent-based simulations and real-life validation with the implementation of an Android prototype. Simulations have shown that it can improve the avoidance of malicious nodes as much as 60%-70% in very hostile environments, which are defined as environments where the vast majority of the nodes are malicious either by not providing the promised service or by carrying out attacks such as Sybil attacks, whitewashing, self-promoting and the like, while still being useful with milder improvements in less hostile scenarios. We have demonstrated through simulation that our incentive schema improves malicious conversions in near a 40% more than the compared metric “Credible Recommendations”.
7. Through real testing with the Android prototype we have assessed the feasibility of such a UCN connectivity sharing service, obtaining average connection setup times which are acceptable most of the times and connection times which are long enough to even be able to stream quality video or perform time consuming operations. Even in the shortest of the connection times, it is usually enough to at least upload and download required small amounts of data in order to perhaps sync or upload statistics to a given service or application. Our real life experimental study using this Android prototype implementation contributed accurate and meaningful results on the performance in regards of:

   a. Uploaded and downloaded data
   b. Connection setup times
   c. Connection length times
   d. Battery and CPU use

All in all, we think that our framework, its implementation through simulation and its prototype testing have achieved satisfactory results and have proven beyond a reasonable doubt that we have achieved results that can be considered beyond the current state of the reviewed literature for the related fields of this thesis.

How does this thesis answer the important questions listed at the beginning:

- *Can current trust management and cooperation incentives frameworks/metrics be applied in a fully decentralized manner in user-centric wireless network environments?* **Yes, as proved by this thesis and by using Trust Transfer in our framework, we can provide both trust management and cooperation incentives in a fully decentralized way.**
- *Is there any framework/metric for user-centric wireless network environments which is fully Sybil attack resistant?* **While, as can be seen in the State of the Art review done in this thesis in Chapter 3, there is no framework fully resistant to Sybil attacks, Trust Transfer is fully Sybil resistant for positive recommendations, being it the closest it can get to the intended result.**
- *Can computational trust management be used to empower cooperation incentives among those users participating in the UCN?* **Yes, as proved in this thesis we can use trust management in order to promote cooperation amongst users by assigning trust points a value that can be exchanged for other services in the system, thus incentivizing the users to behave correctly and to be more trustable.**
- *Which leads us to the key question at the core of this thesis: Can trust management and cooperation incentives be coupled into a unique fully decentralized framework to improve user-centric wireless network environments?* **Yes, as proved in this thesis, our combination of trust management and cooperation incentives with the side improvements of dispositional trust adaptation and data depletion control can work coupled together in a fully decentralized framework, effectively improving the overall user-centric network or system where it has been applied.**

6.2 Future Work

There are many things still not implemented or that can be improved from our current framework's status. Firstly, the Sybil resistance of the framework is limited to positive recommendations, while for negative recommendations there is no guarantee that Sybil resistance will hold. An in depth study and possibly new mechanisms should be devised in order to achieve Sybil protection for negative recommendations.

Another point that can be improved is related to simulations. We haven't simulated and compared our framework regarding attack resistance versus the most relevant and well-known frameworks, due to the difficulty and time-consuming requirements of implementing ourselves the logic behind metrics such as EigenTrust, PowerTrust or Appleseed. While there is at least one readily available simulation tool [71], which has those implemented, extending the tool in order to include our own framework metric proved to be an extremely difficult task as well, and it was not done finally. It would be extremely interesting carrying out one of these two approaches in the future in order to better understand where our framework stands in respect to the others.

Finally, the Android prototype does not implement the full framework, but just the service sharing functionality. While this has allowed us to test in real situations if our approach is feasible from a performance point of view and has enabled us to collect very interesting and meaningful data, it would be desirable to implement the full framework's functionality, including the trust management part and the cooperation incentives part.

Bibliography

[1] J.-M. Seigneur, “Trust, Security and Privacy in Global Computing,” Trinity College Dublin, 2005.

[2] E. Koutrouli and A. Tsalgatidou, “Credible Recommendation Exchange Mechanism for P2P Reputation Systems,” in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2013, pp. 1943–1948.

[3] “The World in 2014: ICT Facts and Figures,” *ITU*. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx. [Accessed: 27-May-2014].

[4] M. Weiser, “The computer for the 21st century,” *Sci. Am.*, vol. 265, no. 3, pp. 94–104, 1991.

[5] J.-M. Seigneur, “The emotional economy for the augmented human,” in *Proceedings of the 2nd Augmented Human International Conference*, 2011, p. 24.

[6] J. R. Douceur, “The Sybil Attack,” in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Springer Berlin Heidelberg, 2002, pp. 251–260.

[7] D. Z. Kevin Hoffman and C. Nita-Rotaru, “A Survey of Attack and Defense Techniques for Reputation Systems,” Purdue University, CSD TR #07-013, 2007.

[8] “European Commission : CORDIS :Home.” [Online]. Available: http://cordis.europa.eu/fp7/. [Accessed: 24-Nov-2014].

[9] “ULOOP | ISS - Institute of Services Science - University of Geneva.” [Online]. Available: http://iss.unige.ch/?q=content/uloop-0. [Accessed: 24-Nov-2014].

[10] C. B. Lafuente, J.-M. Seigneur, R. Sofia, C. Silva, and W. Moreira, “Trust Management in ULOOP,” in *User-Centric Networking*, Springer, 2014, pp. 107–119.

[11] “Tefis.” [Online]. Available: http://www.tefisproject.eu/. [Accessed: 24-Nov-2014].

[12] “Muses.” [Online]. Available: http://www.musesproject.eu/. [Accessed: 24-Nov-2014].

[13] C. B. Lafuente and J.-M. Seigneur, “Dispositional Trust Adaptation in User-Centric Networks,” in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, 2013, pp. 1121–1128.

[14] C. B. Lafuente and J.-M. Seigneur, “Extending Trust Management with Cooperation Incentives: Achieving Collaborative Wi-Fi Sharing Using Trust Transfer to Stimulate Cooperative Behaviours,” in *Trust Management VIII*, Springer, 2014, pp. 157–172.

[15] C. B. Lafuente and J.-M. Seigneur, “Achieving Collaborative Wi-Fi Sharing without Changing Current Technologies,” in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, 2013, pp. 1510–1515.

[16] C. Ballester Lafuente and J.-M. Seigneur, "Crowd augmented wireless access," in *Proceedings of the 3rd Augmented Human International Conference*, 2012, p. 25.

[17] "Multimethod Simulation Software and Solutions." [Online]. Available: http://www.anylogic.com/. [Accessed: 27-May-2014].

[18] C. Ballester Lafuente, J.-M. Seigneur, and T. Lyon, "Collaborative Wireless Access to Mitigate Roaming Costs," in *ICNS 2014, The Tenth International Conference on Networking and Services*, 2014, pp. 107–115.

[19] D. H. McKnight and N. L. Chervany, "The meanings of trust," 1996.

[20] D. M. Romano, "The Nature of Trust: Conceptual and Operational Clarification," 30-Jan-2003. [Online]. Available: http://etd.lsu.edu/docs/available/etd-0130103-070613/. [Accessed: 27-May-2014].

[21] S. P. Marsh, "Formalising Trust as a Computational Concept," Dissertation, Department of Mathematics and Computer Science, University of Stirling, 1994.

[22] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000*, 2000, p. 9 pp. vol.1–.

[23] Z. Yan and P. Cofta, "Methodology to Bridge Different Domains of Trust in Mobile Communications," in *Trust Management*, P. Nixon and S. Terzis, Eds. Springer Berlin Heidelberg, 2003, pp. 211–224.

[24] G. Suryanarayana, J. R. Erenkrantz, S. A. Hendrickson, and R. N. Taylor, "PACE: an architectural style for trust management in decentralized applications," in *Fourth Working IEEE/IFIP Conference on Software Architecture, 2004. WICSA 2004. Proceedings*, 2004, pp. 221– 230.

[25] G. Hardin, "The Tragedy of the Commons," *Science*, vol. 162, no. 3859, pp. 1243–1248, Dec. 1968.

[26] G. M. Greco and L. Floridi, "The Tragedy of the Digital Commons," *Ethics Inf Technol*, vol. 6, no. 2, pp. 73–81, Jun. 2004.

[27] C. Macian and J. Infante, "The tragedy of the commons vs P2P success: An analysis of the conditions for cooperative sustainability in the file-sharing world," in *ITS 2008, International Telecommunications Society, 17th Biennial Conference*, Montreal, Canada, 2008.

[28] L. Buttyan and J. Hubaux, "Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks," in *In Technical Report DSC/2001/001,Swiss Federal Institute of Technology – Lausanne, Department of Communication Systems, 2001. ZUSAMMENFASSUNG*.

[29] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Consulted*, vol. 1, p. 2012, 2008.

[30] K. Wei, A. J. Smith, Y.-F. R. Chen, and B. Vo, "WhoPay: A Scalable and Anonymous Payment System for Peer-to-Peer Environments," in *26th IEEE International Conference on Distributed Computing Systems, 2006. ICDCS 2006*, 2006, pp. 13–13.

[31] A. Diaz-Pines, "International Mobile Data Roaming," OECD Publishing, OECD Digital Economy Paper 180, 2011.

[32] "Measures concerning the European single market for electronic communications and to achieve a Connected Continent, and amending

Directives 2002/20/EC, 2002/21/EC, 2002/22/EC, and Regulations (EC) No 1211/2009 and (EU) No 531/2012.” .

[33] S. Ries, “Certain Trust: A Trust Model for Users and Agents,” in *Proceedings of the 2007 ACM Symposium on Applied Computing*, New York, NY, USA, 2007, pp. 1599–1604.

[34] L. A. Martucci, S. Ries, and M. M&uuml;hlh&auml;ser, “Sybil-Free Pseudonyms, Privacy and Trust: Identity Management in the Internet of Services,” *J. Inf. Process.*, vol. 19, pp. 317–331, 2011.

[35] C.-N. Ziegler and G. Lausen, “Spreading Activation Models for Trust Propagation,” in *the International Conference on e-Technology, e-Commerce, and e-Service*, 2004.

[36] R. Levien, “Attack-resistant trust metrics,” in *Computing with Social Trust*, Springer, 2009, pp. 121–132.

[37] D. Ingram, “An Evidence Based Architecture for Efficient, Attack-Resistant Computational Trust Dissemination in Peer-to-Peer Networks,” in *Trust Management*, P. Herrmann, V. Issarny, and S. Shiu, Eds. Springer Berlin Heidelberg, 2005, pp. 273–288.

[38] C. Bryce, P. Couderc, J.-M. Seigneur, and V. Cahill, “Implementation of the SECURE Trust Engine,” in *Trust Management*, P. Herrmann, V. Issarny, and S. Shiu, Eds. Springer Berlin Heidelberg, 2005, pp. 397–401.

[39] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The Eigentrust Algorithm for Reputation Management in P2P Networks,” in *Proceedings of the 12th International Conference on World Wide Web*, New York, NY, USA, 2003, pp. 640–651.

[40] F. Tegeler and X. Fu, “SybilConf: Computational Puzzles for Confining Sybil Attacks,” in *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, 2010, pp. 1–2.

[41] D. Quercia, S. Hailes, and L. Capra, “Lightweight Distributed Trust Propagation,” in *Seventh IEEE International Conference on Data Mining, 2007. ICDM 2007*, 2007, pp. 282–291.

[42] M. S. Fallah and M. Mouzarani, “A Game-based Sybil-resistant Strategy for Reputation Systems in Self-organizing MANETs,” *Comput J*, vol. 54, no. 4, pp. 537–548, Apr. 2011.

[43] W. Wei, F. Xu, C. C. Tan, and Q. Li, “SybilDefender: Defend against sybil attacks in large social networks,” in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 1951–1959.

[44] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman, “SybilGuard: Defending Against Sybil Attacks via Social Networks,” *IEEEACM Trans. Netw.*, vol. 16, no. 3, pp. 576–589, Jun. 2008.

[45] N. Tran, J. Li, L. Subramanian, and S. S. M. Chow, “Optimal Sybil-resilient node admission control,” in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 3218–3226.

[46] X. Titi, “QoS/QoE-based Wi-Fi Network trust and reputation,” University of Geneva, 2014.

[47] M. Feldman, K. Lai, I. Stoica, and J. Chuang, “Robust Incentive Techniques for Peer-to-peer Networks,” in *Proceedings of the 5th ACM Conference on Electronic Commerce*, New York, NY, USA, 2004, pp. 102–111.

[48] S. Zhong, J. Chen, and Y. R. Yang, “Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003, vol. 3, pp. 1987–1997 vol.3.

[49] W. Wu, J. C. S. Lui, and R. T. B. Ma, “A game theoretic analysis on incentive mechanisms for wireless ad hoc VoD systems.,” in *WiOpt*, 2012, pp. 177–184.

[50] Z. Zhang, S. Chen, Z. Mo, and M. Yoon, “An Efficient Incentive Scheme with a Distributed Authority Infrastructure in Peer-to-peer Networks,” *J Parallel Distrib Comput*, vol. 72, no. 12, pp. 1741–1752, Dec. 2012.

[51] M. E. Mahmoud and X. Shen, “RISE: Receipt-Free Cooperation Incentive Scheme for Multihop Wireless Networks,” in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5.

[52] A. Aldini and A. Bogliolo, “Model Checking of Trust-Based User-Centric Cooperative Networks,” presented at the AFIN 2012, The Fourth International Conference on Advances in Future Internet, 2012, pp. 32–41.

[53] J.-P. Hubaux, L. Buttyán, and S. Capkun, “The Quest for Security in Mobile Ad Hoc Networks,” in *Proceedings of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, New York, NY, USA, 2001, pp. 146–155.

[54] A. A. Pirzada and C. McDonald, “Establishing Trust in Pure Ad-hoc Networks,” in *Proceedings of the 27th Australasian Conference on Computer Science - Volume 26*, Darlinghurst, Australia, Australia, 2004, pp. 47–54.

[55] R. Mitchell and I.-R. Chen, “On Survivability of Mobile Cyber Physical Systems with Intrusion Detection,” *Wirel. Pers. Commun.*, vol. 68, no. 4, pp. 1377–1391, Feb. 2013.

[56] M. Chorzempa, J.-M. Park, and M. Eltoweissy, “SECK: survivable and efficient clustered keying for wireless sensor networks,” in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, 2005, pp. 453–458.

[57] M. M. Hasan and J. P. Jue, “Survivable Self-Organization for Prolonged Lifetime in Wireless Sensor Networks,” *Int. J. Distrib. Sens. Netw.*, vol. 2011, p. e257156, May 2011.

[58] O. M. Al-Kofahi and A. E. Kamal, “Survivability strategies in multihop wireless networks,” *IEEE Wirel. Commun.*, vol. 17, no. 5, pp. 71–80, Oct. 2010.

[59] T. El Maliki, “Security adaptation in highly dynamic wireless networks,” University of Geneva, 2014.

[60] F. Xing and W. Wang, “On the Survivability of Wireless Ad Hoc Networks with Node Misbehaviors and Failures,” *IEEE Trans. Dependable Secure Comput.*, vol. 7, no. 3, pp. 284–299, 2010.

[61] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[62] S. Milgram, “The small world problem,” *Psychol. Today*, vol. 2, no. 1, pp. 60–67, 1967.

[63] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna, “Four Degrees of Separation,” in *Proceedings of the 4th Annual ACM Web Science Conference*, New York, NY, USA, 2012, pp. 33–42.

[64] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, “The anatomy of the facebook social graph,” *ArXiv Prepr. ArXiv11114503*, 2011.

[65] K. Klemm and V. M. Eguíluz, “Highly clustered scale-free networks,” *Phys. Rev. E*, vol. 65, no. 3, p. 036123, Feb. 2002.

[66] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, “Efficiency of scale-free networks: error and attack tolerance,” *Phys. Stat. Mech. Its Appl.*, vol. 320, pp. 622–642, 2003.

[67] “Tefis.” [Online]. Available: http://www.tefisproject.eu/about/tefis-use-cases/smart-ski-resort. [Accessed: 24-Nov-2014].

[68] “Mobile data: how much do you need? | Analysis | Features | PC Pro.” [Online]. Available: http://www.pcpro.co.uk/features/379402/mobile-data-how-much-do-you-need. [Accessed: 27-May-2014].

[69] “Tileset Concepts and Terminology.” [Online]. Available: http://www.microimages.com/documentation/FeatureSummaries/Tileset%20Terminology.pdf. [Accessed: 24-Nov-2014].

[70] “Genève Aéroport - Statistics.” [Online]. Available: http://gva.ch/en/desktopdefault.aspx/tabid-244/. [Accessed: 27-May-2014].

[71] F. G. Marmol and G. M. Perez, “TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks,” in *IEEE International Conference on Communications, 2009. ICC ’09*, 2009, pp. 1–5.

Annex A – Links to Simulations

This annex provides the links to the simulations performed in this thesis in order to allow the reproduction of the experiments carried out.

All the three simulations have been done using AnyLogic, thus the links provided contain each of the AnyLogic project files (.alp) plus the additional resources such as images that are needed to run the simulations.

The links to the simulations can be found in GitHub repositories in the following locations:

1. SmartSki Resort Simulation: https://github.com/carlos-ballester/PhD-Thesis-Simulations/tree/master/SimpleSkiResort
2. Human-AccesPoint Interaction Simulation: https://github.com/carlos-ballester/PhD-Thesis-Simulations/tree/master/AP-HumanInteraction
3. Credit Transfer Airport Simulation: https://github.com/carlos-ballester/PhD-Thesis-Simulations/tree/master/CreditTransferFinal