Cooperative Social System based on Trust for Carpooling

Pino Caballero-Gil, Francisco Martin-Fernandez and Candido Caballero-Gil
Department of Computer Engineering
University of La Laguna
38271 La Laguna Tenerife, Spain
Email: {pcaballe, fmartinf, ccabgil}@ull.edu.es

***Abstract*—One of the worst traffic problems today is caused by the high percentage of commuters who use private vehicles and generate huge traffic jams. This issue has led to an increase in research on optimization of vehicle occupancy in urban areas because at rush hour in large towns most vehicles only carry one passenger. The solution of sharing cars, known as carpooling, is already being successful in major cities. However, carpooling is not yet considered a safe and reliable solution by many users. With the wide spread of mobile technology and social networks, it is possible to create a trust-based platform to encourage carpooling through a convenient, fast and secure system. The main objective of this work is the design and implementation of a carpool system that can be used to improve previous systems by focusing on the security aspect. The new system guarantees the privacy of users, and measures the trust levels through a reputation algorithm. The proposal has been developed as a mobile application according to the Android Open Source Project initiative. Currently, the application has received thousands of downloads in the Google Play Store.**

I. Introduction

The recent and rapid increase in the number of vehicles has raised pollution, which is one of the main factors of the global warming. This fact has led governments to take steps to try to decrease pollution in urban centres, through measures such as encouraging the use of public transport. However, public transport is not always the most convenient and/or affordable solution for everybody.

Another practical solution to the problem would be to avoid the low occupancy of most vehicles in cities, through sharing cars so that empty seats are used in most trips. This modality is known by the term carpooling and has been proposed as an effective way to reduce pollution and spending. A related but different approach, known as carsharing, is based on collective fleets of cars with multiple users, but such a solution does not solve as many problems as carpooling. Moreover, ridesharing is the general term used to refer to solutions for sharing the use of a car with other people in order to travel to a given destination. Besides carsharing and carpooling (also known as real-time or instant or dynamic or ad-hoc ridesharing), ridesharing also includes other versions known as slugging, lift sharing and covoiturage. However, this work does not address other ridesharing solutions different from carpooling.

Both types of collaborative solutions are increasingly used since the beginning of the economic crisis, thanks to technology 2.0. They are applicable in almost any environment, but are especially useful in situations like universities, holidays, long journeys and urban centres. This is because in these situations both the owners and passengers of vehicles have the same motivations to consider the carpooling solution. Usually, their main goal is to share fuel cost, but there may be other reasons, such as to try to avoid parking problems, to want to meet new people or to make a contribution to the environmental protection, etc.

The main problem of carpooling is reliability because the service requires users to be confident that the driver will drive them to their destinations. An improvement of existing carpooling solutions is proposed here, based on the use of the latest technological advances of smartphones and social networks, to increase reliability. On the one hand, the described solution allows the establishment of trust and reputation accountability between drivers and passengers, while on the other hand, it protects the privacy of all users.

This paper is organized as follows. Section 2 summarizes several related works. Section 3 introduces the general design of the proposal, which is mainly based on the reputation algorithm explained in Section 4. Section 5 briefly analyses the security of the proposal. The developed Android application is described in Section 6. Finally, some conclusions and open problems can be found in Section 7.

II. State of the Art

Car sharing is very popular in the United States, where its practice is regulated by road signs. In other countries like United Kingdom and Italy, many websites exist that offer carpooling services.

The first carpooling projects emerged in the late 1980s [1], but in those days, without the technology available today, they had to face many difficult obstacles such as the need to develop a user network and convenient means of communication. Gradually, the media used to organize the trips changed from telephone to other more flexible tools such as the Internet, email and smartphones. Nowadays, many different carpooling platforms and services exist, but even today, they may be considered in their early stages because none of them has reached a critical mass of users [2].

Table I shows several features of different existing carpooling systems, including the most relevant security-related ones. In particular, we have chosen for this comparative analysis the representative systems: Amovens [3], Blablacar [4], CarPooling [5], Compartir.org [6], and ZimRide [7].

TABLE I. CARPOOLING PLATFORMS

Platform	Social Network	Privacy	Reputation System	Phone Cert.	Trust Alg.
Amovens [3]	yes	yes	yes	no	no
Blablacar [4]	yes	yes	yes	yes	no
CarPooling [5]	yes	no	yes	no	no
Compartir [6]	no	yes	no	no	no
ZimRide [7]	yes	yes	no	no	no
Proposed System	**yes**	**yes**	**yes**	**no**	**yes**

BlaBlaCar can be considered one of the largest carpooling networks in Europe. It is a service focused on long distance trips, which uses social networks for registration and feedbacks as a guarantee, and an enabler of real connections between users. One of the biggest carpooling service in the United States is ZimRide, where payments are made through credit cards account and PayPal.

The main trust enforcing system in all these platforms is based on points given by users. However, to bypass this security system is quite easy because those users who obtain a negative score can create a new profile with new credentials and no points.

Apart from these practical platforms already in operation, there are several papers that propose different solutions. The work [8] shows an integrated system for the organization of carpooling service by using different technologies such as web, GIS and SMS. The authors of [9] propose a web platform for carpooling. The paper [10] presents a carpooling architecture that uses a credit mechanism to encourage cooperation between users. A more recent work is [11], where an algorithm to encourage carpooling is proposed based on assigning priorities to users with positive feedbacks through a fuzzy logic scheme. Another paper, [12], defines a push service to promote carpooling through instant processing. Finally, another interesting proposal is [13], based on a secure multi-agent platform focused on several security services that allow both the mutual authentication between the users, and between the application components and the system.

The main aspect of our work is different from the aforementioned because it deals with the trust aspect of carpooling services through a combination of reputation measurement with privacy protection.

III. PROPOSED SYSTEM

The main objective of the proposed design is the increase of both usability and security because its key factors are user-friendliness and privacy. One of its main features is that users who publish their trips have their privacy fully protected. Unlike other carpooling platforms, in the described system, no user is allowed to access data such as email, phone or full name of others, unless he/she is authenticated on the platform and the algorithm for checking mutual trust returns a valid permission for him/her. In this case, the interested user can see all the data in detail. Otherwise, he/she can only send a request so that the receiver can decide whether the applicant is to be trusted or not.

The algorithm is based on trust relationships so that people who want to use the platform first need to be authenticated in the platform through social networks such as Facebook, Twitter or Google+. In this way, the algorithm checks the existence of some chain of trust between the applicant and other users, based on the so-called rule of six degrees of separation [14]. Six degrees of separation is the theory according to which everyone is six or fewer steps away from each other in the world, so that a chain of 'a friend of a friend' statements can be made to connect any two people in a maximum of six steps.

Besides, the reputation gained through the use of the application is an influential factor, which is considered in the decision on whether carpooling with a person or not. In order to do this, after every shared trip, the application asks both driver and passengers to score the other users. Such scores are used in future trips so that seats offered by drivers with good scores appear in better positions than others with lower scores. Also well-scored passengers have higher probability to have access to more details of drivers.

The overall system architecture uses an application model known as client-server (see Figure 1). Its main elements are:

- Client: Mobile device used by the system.
- Server: Hosted in the cloud, and divided into two parts. On the one hand, the Google Cloud Messaging server is the GCM server that handles all the notifications and is responsible for sending the notification when the receiver clients are alive. On the other hand, the server that stores in its DataBase all the data related to users and system is the DB dedicated server. It also serves as a gateway for sending notifications between the client and the GCM server.

The proposed scheme protects user privacy through limited and controlled access to user data, according to the trust level stated for the relationship between each pair of users. This trust level is reached through the combination of direct scores and trust networks so that it provides the system with enough data to deduce whether people can trust each other or not. In this way, privacy is dealt with as one of the most important aspects of the proposed carpooling system.

A first approach to the development of a trust measurement algorithm that provides a value to each pair of users is based on the use of the PageRank algorithm to predict whether two people can rely each other or not. However, since this algorithm does not meet the requirements for the morphology of the specific problem, a second approach is also being used to complement it, based on Bayesian networks to know whether people can trust others or not. The refined algorithm for trust measurement is available in the Android application.

IV. THE REPUTATION ALGORITHM

No existing carpooling proposal offers the user a quantitative method that can be used to decide whether another user is trustful to share a vehicle or not. Some of the proposals even do not allow users to decide who may or may not apply their routes. A few proposals use a quantitative method based on similarities among users [15] [16]. Their main problem is that they have to collect information about the attributes and characteristics of each user.

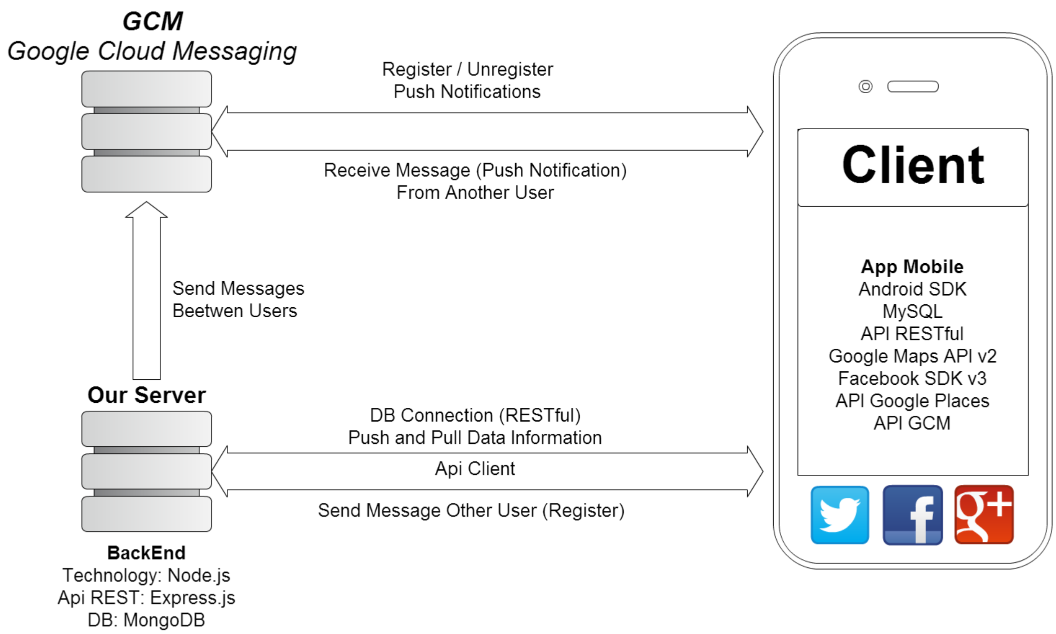

Fig. 1. Carpooling System Architecture

The proposal here described aims to be simple for the user, so that he/she has not to fill out any information about his/her attributes. With a simple click to enable social login on the network after a previous registration, the user may enter the system and start using the platform.

The main feature of our scheme is the proposed reputation algorithm because thanks to it, the user can use a quantitative measure to decide whether trusting another user or not. The algorithm is based on the theory of six degrees of separation and individual scores within the platform. The number of steps may be reduced significantly by introducing the concept of social networks. The application uses social networks when logging into the application to create network users to be used to interconnect with each other and provide a reliable measure of confidence. Through the use of social networks we can ensure that the six degrees of separation are reduced to only four. In particular, according to several researches on Facebook [17] [18] [19], the obtained average distance was 3.9, corresponding to intermediaries or 'degrees of separation', what shows that the world is even smaller than expected.

The reputation measurement is expressed as a number between 0 and 10, and is computed from the values given by each pair of users to inform about reliability on each other. This measure is calculated by taking into account the two parameters mentioned above: degree of friendship and appreciation of other users in the platform. If the Facebook social network is used, the score given by the system with the degrees of friendship is as follows:

- 7 points if a user has only one degree of separation (i.e. it is a direct friend).
- 6 points if both users have a mutual friend.
- 4 points if between those users there is a chain of friendship of more than two friends.
- 0 points if there is no friendship chain.

Other users obtain the remaining scores from the assessments after sharing routes. When a route is completed, users who participated in it, can vote giving a score between 1 and 5 stars. Each passenger individually assesses the driver, and the driver individually assesses passengers. The weight is higher from driver to passengers than from passengers to driver, as the driver puts his/her vehicle available to the users. In order to apply the different ratings on a user, we use a simple arithmetic average. The metric taken into account for these ratings is as follows:

- 1 star will impact the score in -3 points if that rating was given to the passenger by the driver. -2 points if the passenger was who rated the driver.
- 2 stars will impact the score in -1 points if that rating was given to the passenger by the driver. 0.25 point if the passenger was who rated the driver.
- 3 stars will impact the score in 1.5 points if that rating was given to the passenger by the driver. 0.5 point if the passenger was who rated the driver.
- 4 stars will impact the score in 2.5 points if that rating was given to the passenger by the driver. 2 points if the passenger was who rated the driver.
- 5 stars will impact the score in 3 points if that rating was given to the passenger by the driver. 3 points if the passenger was who rated the driver.

The maximum reputation score that a user can get is 10 points, which corresponds to the situation when the user's direct friends have rated him/her with the highest scores. A user can have total null valuation if he/she is starting to be known, does not have any degree of friendship, and/or has had negative reviews. In this case, the system does not use a score below 0, so 0 is the minimum score for any user.

The reputation valuation is dynamically calculated as a function of the friendship degree that a user has. This helps users to have a reliability measure about whether to trust another user of the platform or not. Besides, only users who have a valuation higher than 7.5 points and/or users who have been accepted by the driver to make the route can see certain route data, such as the phone number or other confidential data.

Thus, the mathematical expression applied to the calculation of the reputation score in the algorithm is shown in Equation 1:

$$\left(lvFs + \frac{\sum_{i=1}^{n} rat[i]}{n} \right) \quad (1)$$

where:

- $lvFs$: Friendship level measured in points between 0 and 7, depending on the level of friendship each user has with the other user, as explained above.
- $rat[]$: Each of the ratings a user has received both as driver and passenger, on the used routes. The points at which the rating of 1-5 stars are mapped.
- n: Total number of ratings that the user has received.

V. Security of the Scheme

Regarding the safety of the platform, Sybil attack is a notorious attack in traditional carpooling systems [20]. These types of attacks are hacking attacks on peer-to-peer networks [21] where a malicious device illegitimately takes multiple identities by forging them. Due to the privacy-preserving environment of carpooling schemes, Sybil vulnerability is generally hard to defend against.

In a Sybil attack, the attacker subverts the reputation system of a peer-to-peer network by creating a large number of pseudonymous identities to gain a disproportionately large influence. Vulnerability of a reputation system to a Sybil attack depends on how cheaply identities can be generated, the degree to which the reputation system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically.

An entity on the analysed peer-to-peer network is a piece of software that has access to local resources. It advertises itself on the peer-to-peer network by presenting an identity. More than one identity can correspond to a single entity. In other words, the mapping of identities to entities is many to one. Entities in peer-to-peer networks use multiple identities for purposes of redundancy, resource sharing, reliability and integrity. In peer-to-peer networks, the identity is used as an abstraction so that a remote entity can be aware of identities without necessarily knowing the correspondence of identities to local entities. By default, each identity is usually assumed to correspond to a different local entity. In fact, many identities may correspond to the same local entity.

A dishonest member or adversary user may present multiple identities to act as multiple nodes. After becoming part of the network, he/she may eavesdrop on communications and/or act maliciously. By masquerading through multiple identities, the attacker can control the network.

Considering, for example, the following scenario (see Figure 2): A 'bad' user ($B0$) sets up several bogus accounts in social media and the proposed system ($B1$, $B2$, $B3$, etc.). He/she then advertises a possible trip from $X \mapsto Y \mapsto Z$. For the first part ($X \mapsto Y$) he/she uses the bogus accounts to claim that his/her vehicle is full and all these 'passengers' state that they get off at Y. He/she can then get excellent scores and increase his/her reputation. At some point, this will be so high that other users will be able trust him/her from $Y \mapsto Z$.

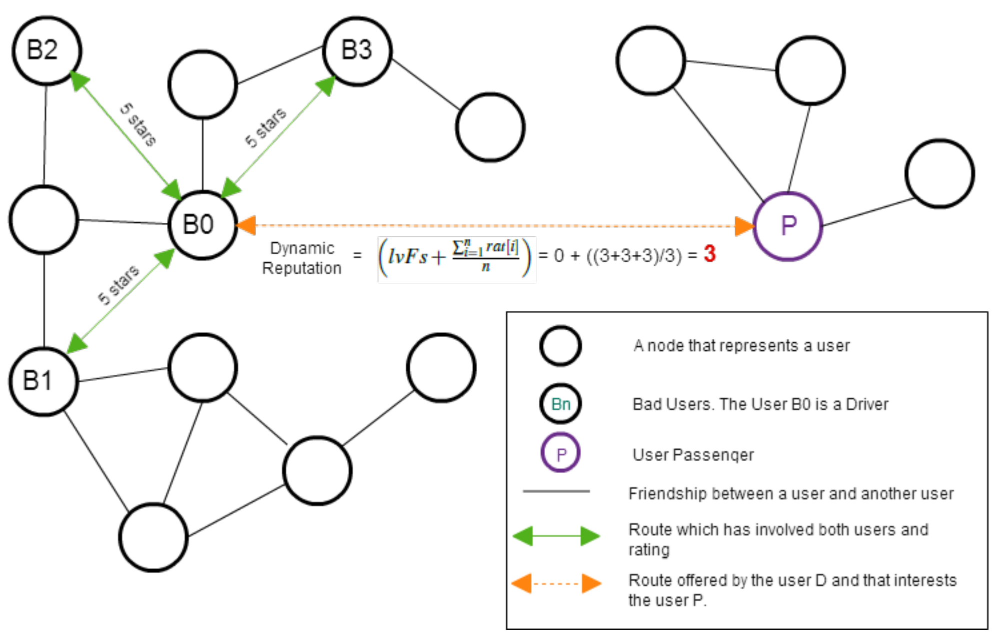

Fig. 2. Insufficient Dynamic Reputation Rating

The proposed algorithm reduces significantly the vulnerability to the attack described above because most of the score of the algorithm is preceded by confidence in degrees of friendship that binds each user to another user. Thus, if a user does not know (at all) another user, very high ratings of the latter in the system are not reliable enough for the former. It is remarkable that in our system the minimum reliability to trust another user is 7.5 points. At most, a user can have up to 3 points related to ratings for the done routes (see Figure 2) , far from the 7.5 points that are needed at least for that user to be reliable.

VI. The Android Application

The proposed design has been included in an Android application that is already published in the Google Play Store under the name 'Carpoolap' (see Figure 3).

The Android application is developed for the versions 3.0 or higher of the operating system. APIs like Google Maps v3.0, Google Places, Google Cloud Messaging, etc., and Facebook SDKs 3.0 and libraries like Action Bar Sherlock for the functionality of the new versions of Android on older versions, were used. Autocomplete in address searches, Google Maps 3D Technology, design based on the latest versions of Android, push notifications with requests or responses of passengers or drivers, etc. are among the features of the Android Application.

Each user can see the routes he/she proposes as driver, and whether potential passengers exist for those routes. Besides, with colour codes, he/she can know the routes that each user has already made and the routes that have been confirmed by users. For the assessment of users participating in a route, after finishing it, each one can give a score. In order to deploy the carpooling platform, a server is also needed, so we developed one using JavaScript technologies by frameworks like 'node.js' and 'express.js'. As a database for all the data centralized on this server, we decided to adopt a No SQL database, such as MongoDB [22]. We deployed our server on a micro instance of Amazon Web Services, specifically under Ubuntu machine with Amazon EC2 account.

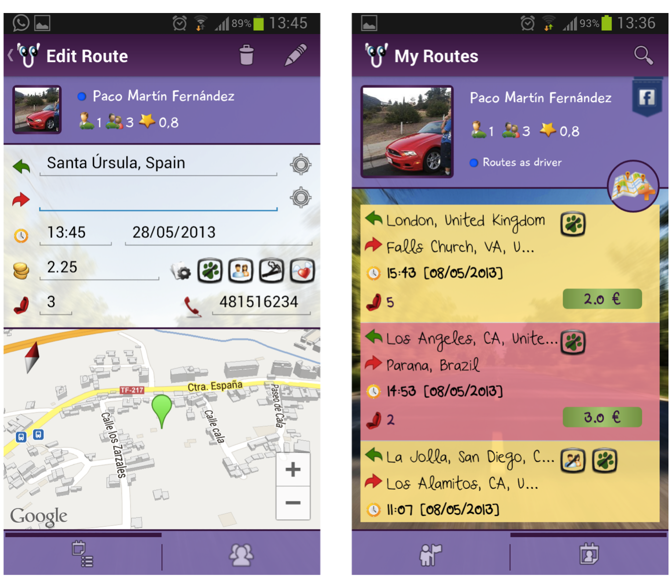

Fig. 3. Carpoolap Screens: Route Edition & Routes List

VII. Conclusions

This work proposes a new carpool system that can be used to increase safety and reliability of existing systems. After reviewing existing literature on this type of systems, several improvements are described here. In particular, the proposed system includes both a strong social component that encourages trust among users, and a progressive access to user data based on confidence levels estimated by the system. This work not only includes the description of the system design, but also its development and implementation in an application for mobile devices under the platform Android Open Source Project. The application that uses this research to create a carpool platform has been recently published in the Google Play Store and has already thousands of downloads. There are some open issues and challenges to address, such as the development of the application for other platforms, or the creation of an API for its use in third party applications.

Acknowledgment

Research supported by the Spanish MINECO and the European FEDER under projects TIN2011-25452, IPT-2012-0585-370000 and RTC-2014-1648-8, and the FPI scholarship BES-2012-051817.

References

[1] R. Teal, *Carpooling: Who, how and why*. Transportation Research, v. 21A pp. 203-214, 1987.

[2] F.E. Prettenthaler, K.W. Steininger, *From ownership to service use lifestyle: the potential of car sharing*. Ecological Economics 28, pp. 443-453, 1999.

[3] Amovens: https://www.amovens.com, March 2015.

[4] Blablacar: https://www.blablacar.es, March 2015.

[5] Carpooling: http://www.carpooling.es, March 2015.

[6] Compartir.org: http://compartir.org/, March 2015.

[7] Zimride: http://www.zimride.com/, March 2015.

[8] R.W. Calvo, F. De Luigi, P. Haastrup, V. Maniezzoi, *A distributed geographic system for the daily car pooling problem*. Computers and Operations Research 31, v. 13, 2004.

[9] A. Roberts, H. Pimentel, S. Karayevm, *CabFriendly: A Cloud-based Mobile Web App*. Amazon's EC2, 2011.

[10] J. Ferreira, P. Trigo, P. Filipe, *Collaborative Car Pooling System*. International Conference on Sustainable Urban Transport and Environment, 2009.

[11] M. Collotta, G. Pau, V.M. Salerno, G. Scata, *A novel trust based algorithm for carpooling transportation systems*. IEEE International Conference on Energy Conference and Exhibition, pp. 1077-1082, 2012.

[12] A.J. Fougeres, P. Canalda, T. Ecarot, A. Samaali, L. Guglielmetti, *A Push Service for Carpooling*. IEEE International Conference on Green Computing and Communications, pp. 685-691, 2012.

[13] C. Bonhomme, G. Arnould, D. Khadraoui, *Dynamic carpooling mobility services based on secure multi-agent platform*. IEEE Global Information Infrastructure and Networking Symposium, pp. 1-6, 2012.

[14] D.J. Watts, S.H. Strogatz, *Collective dynamics of small-world networks*. Nature 393, pp. 440-442, 1998.

[15] S. Cho, A. Yasar, L. Knapen, T. Bellemans, D. Janssens, G. Wets, *A Conceptual Design of an Agent-based Interaction Model for the Carpooling Application*. 1st International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications, pp. 801-807, 2011.

[16] N.V. Pukhovskiy, R.E. Lepshokov, *Real-Time Carpooling System*. International Conference on Collaboration Technologies and Systems, pp. 648-649, 2011.

[17] E.Y. Daraghmi, S.M. Yuan, *We are so close, less than 4 degrees separating you and me!*. Computers in Human Behavior, pp. 273-285, 2014.

[18] J. Ugander, B. Karrer, L. Backstrom, C. Marlow, *The Anatomy of the Facebook Social Graph*. Cornell University Library, November 2011.

[19] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, S. Vigna, *Four Degrees of Separation*. Cornell University Library, November 2011.

[20] J.R. Douceu, *The Sybil Attack*. 1st International Workshop on Peer-to-Peer Systems, March 2002.

[21] A. Oram, *Peer-to-peer: harnessing the benefits of a disruptive technology*. O'Reilly Media, 2001.

[22] MongoDB: http://www.mongodb.org, January 2015.