Co-utile P2P Ridesharing via Decentralization and Reputation Management

David Sánchez[1], Sergio Martínez, Josep Domingo-Ferrer

UNESCO Chair in Data Privacy, Department of Computer Engineering and Mathematics
Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Catalonia
{david.sanchez,sergio.martinezl,josep.domingo}@urv.cat

Abstract

Ridesharing has the potential to bring a wealth of benefits both to the actors directly involved in the shared trip (e.g., shared travel costs or access to high-occupancy vehicle facilities) and also to the society in general (e.g., reduced traffic congestion and CO_2 emissions). However, even though ridesharing is based on a win-win collaboration and modern mobile communication technologies have significantly eased discovering and managing ride matches, the adoption of ridesharing has paradoxically decreased during the last years. In this respect, recent studies have highlighted how privacy concerns and the lack of trust among peers are crucial issues that hamper the success of ridesharing. In this paper, we tackle both of these issues by means of i) a fully decentralized P2P ridesharing management network that avoids centralized ride-matching agencies (and hence private data compilation by such agencies); and ii) an also decentralized reputation management protocol that brings trust among peers, even when they have not previously interacted. Our proposal rests on the recently proposed notion of *co-utility* (essentially, self-enforcing and mutually beneficial collaboration), which ensures that rational (even purely selfish) peers will find no incentives to deviate from the prescribed protocols. We have tested our system by using data gathered from real mobility traces of cabs in the San Francisco Bay area, and according to several metrics that quantify the degree of adoption of ridesharing and the ensuing individual and societal benefits.

Keywords: Ridesharing, P2P, Privacy-protection, Reputation, Trust, Co-utility

[1]Corresponding author. Address: Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Catalonia. Tel.:+34 977 559657; Fax: +34 977 559710. E-mail: david.sanchez@urv.cat

1. Introduction

Ridesharing is a mode of transportation in which several travelers share a vehicle (typically a private car) for a trip and split the trip costs. In this manner, they can enjoy the convenience and speed of private car rides without paying much more than if using public transportation [17]. Moreover, ridesharing is often seen as a promising means to reduce CO_2 emissions [8] and use finite oil supplies in a wiser way. For end users (drivers and passengers), ridesharing mainly saves travel costs, but it may also reduce travel time in case high-occupancy vehicle (HOV) facilities are available for vehicles carrying two or more people (special lanes, toll booths or parking spaces) and, if widely adopted, by mitigating traffic congestion [9, 24]. In fact, the benefits of congestion mitigation can be very significant and extend far beyond the ridesharers themselves: the annual cost of traffic congestion in the US in terms of wasted time and fuel was estimated at $124 billion in 2013 and, unless significant actions are taken to alleviate it, this cost is expected to rise by 50% by 2030 [20].

Due to the potential benefits for participants, ridesharing naturally defines a win-win scenario whereby the involved parties maximize their outcomes as a result of their collaboration. This mutually beneficial collaboration has been formalized and termed *co-utility* [15]. *Co-utile* protocols are those in which helping other (rational) agents to increase their utilities is also the best way to increase one's own utility. Ridesharing is potentially co-utile because, in terms of cost and time, it is the best option for most agents, even for purely selfish ones.

Ridesharing has been used on a regular basis since the 1970s by means of *carpools*. Carpooling is usually understood as an organized and regular ridesharing service (e.g., employees taking turns to drive each other to work). With the advent of the Internet and the enormous adoption of mobile communication technologies, modern ridesharing is also characterized by a more dynamic (or even real-time) scheduling of rides [9]. By dynamic ridesharing we refer to an automated system that matches drivers and travelers on very short notice or even en route [3]. Current ridesharing platforms are Internet-based agencies (e.g., Carma, BlaBlaCar, CarpoolWorld, etc.) that match drivers' availabilities with passengers' needs in a centralized way [17]; in this scenario, the main challenges are to find optimal matches and immediately satisfy on-demand requests to form ridesharing instantaneously [3].

A recent study [1], however, reports a significant and continuous decline of ridesharing (i.e., ridesharing accounted for 19.7% of work trips in 1980 whereas it fell down to 9.4% in 2013). Why the immediacy and ubiquity of Internet-enabled mobile communications have not shifted people's choices of transportation has been recently studied in [3, 17]. Furuhata et al. [17] identify several difficulties that prevent ridesharing from being widely adopted. Building trust among travelers that do not know each other stands out as a crucial issue. It is well known that most people are reluctant to travel with complete strangers (e.g., according to the survey in [10], only 7% of respondents would accept rides from strangers, whereas 98% and 69% would accept rides from a friend and the friend of a friend, respectively). The user's experience is usually the most reliable foundation of trust, and it can be spread through the network using a reputation system by which users (both drivers and passengers) provide feedback on each other. Even though some matching agencies (such as Carma or Carpool World) already implement reputation systems, there is another key issue that severely hampers the ridesharing model offered by these agencies: users' privacy [17]. Indeed, a significant loss of privacy occurs as a result of matching agencies systematically collecting travel data (this also happens in intelligent transportation systems in general [11, 16]) and reputations of their customers [4], which is an information that may be sold or used for other purposes such as marketing products or personal profiling [32].

To neutralize the above trust-related disincentives to ridesharing without running into privacy-related disincentives, we propose in this paper a fully decentralized P2P ridesharing system with the following features:

- Being fully decentralized, it circumvents central matching agencies that may compile, aggregate and exploit individuals' data (location, travel habits, reputations) or that may bias ride matches because of commercial interests. Moreover, decentralization eliminates a central point of failure that might be the target of external attacks.

- For the sake of user privacy, we maintain information disclosure at a minimum, so that only the driver and the passenger(s) whose trips match learn each other's identity, desired trip and reputation.

- To address the reluctance of users to share trips with strangers, we implement a decentralized reputation management mechanism whereby i) drivers and passengers can rate each other after a shared ride, and ii) peers can learn the reputation of others (resulting from aggregation of ratings of past rides) in a trustworthy way before deciding whether to share or not a future trip. Even though decentralized reputation may look more complex and less reliable than centralized reputation maintained by a trusted party [17], our reputation mechanism is designed in such a way that rational agents are interested in honestly cooperating to maintain it. As a result, it is robust against a number of tampering attacks (e.g., fake reputation reporting, creation of multiple accounts, etc.), because agents cannot derive any benefit from such attacks.

- Our system rests on the theoretical foundations of co-utility [15] to characterize the collaboration between users and ensure that rational users (even purely selfish ones) will follow the proposed protocols. An interaction is co-utile if the best way for an agent to increase her own outcome is to help peers increasing theirs. Thus, if agents are rational, a co-utile system is collaborative in a self-enforcing way, because collaboration is mutually beneficial.

As far as we know, ours is the first fully decentralized P2P ridesharing system that incorporates an also decentralized reputation mechanism [17]. The technical emphasis of our work is thus on the decentralized, privacy-preserving and efficient management and matching of drivers' offers and passengers' requests, as well as on the distributed calculation of peers' reputations. We do not deal with the ride-matching optimization problem (see [3] for a recent survey on that topic). By designing the protocols in our system to be co-utile, we make sure the system will work as intended even if there is no central entity controlling its operation.

The rest of this paper is organized as follows. Section 2 the co-utility of ridesharing and lists the assumptions we make on dynamic ridesharing. Section 3 describes the types of ridesharing, proposes a fully decentralized P2P ride management system and ride-matching protocol, and argues under which circumstances co-utility holds. Section 4 discusses how the lack of *trust* may hinder the co-utility and hence the adoption of ridesharing; then it reviews a decentralized reputation management mechanism that is itself co-utile and can be used to enforce trust (and hence co-utility) in the ridesharing community; finally, it shows how to incorporate reputations in the ride-matching protocol presented in Section 3. Section 5 reports on experiments carried out with real travel data, and evaluates the results according to several metrics that quantify the adoption of ridesharing and the ensuing individual and societal benefits. Finally, Section 6 contains the conclusions and sketches several lines of future research.

2. Co-utility and ridesharing

Co-utility models a kind of interaction between rational agents in which the best option for each agent to reach her own goal is to help other agents to reach theirs. Since we are dealing with rational agents, game theory is a natural framework to formalize this concept. We define co-utility [15] for scenarios that can be represented as perfect-information games; these are games in which each agent making a decision knows the payoffs of all agents under the various possible actions (or sequences of actions), plus any previously made decisions [23]. We represent these games in the so-called extensive form, which is a tree where: (i) nodes are the points where decisions are made, (ii) each node is labeled with the name of the agent making the decision, (iii) outgoing edges in a node represent the available choices (actions) at that node, and (iv) each leaf node is labeled with the tuple of payoffs that agents obtain when the node is reached.

By using this extensive form, we can view a *protocol* (i.e., the actions needed for the completion of a task) as a path that traverses the tree representing the game.

We focus on self-enforcing protocols, that are those from which agents have no rational incentive to deviate. That is, no agent can increase her utility by deviating from the protocol, provided that the other agents stick to it. In game-theoretic terms, this means that, at each successive node of the protocol path, sticking to the next action prescribed by the protocol (taking the next edge in the path) is an *equilibrium* of the remaining subgame (the subtree rooted at the current node), that is, a *subgame perfect equilibrium* of the game.

We say that a self-enforcing protocol is *co-utile* if it results in mutually beneficial collaboration between the participating agents. More specifically, a protocol $\mathcal{P}$ is co-utile if and only if *the three* following conditions hold:

1. $\mathcal{P}$ is self-enforcing;

2. The utility derived by each agent participating in $\mathcal{P}$ is strictly greater than the utility the agent would derive from not participating;

3. There is no alternative protocol $\mathcal{P}'$ giving greater utilities to all agents and a strictly greater utility to at least one agent.

The first condition ensures that, if participants engage in the protocol, they will not deviate. The second condition is needed to ensure that engaging in the protocol is attractive for everyone. The third condition can be rephrased in game-theoretic terms by saying that the protocol is a Pareto-optimal solution of the underlying game.

To assess the co-utility of the dynamic ridesharing scenario considered in this paper, we make the following assumptions that are supported by the figures about ridesharing presented in the introduction:

1. Potential passengers obtain a better utility (i.e., lower cost and/or time, and better convenience) by sharing a ride than by using a private transport or a taxi (which is more costly) or the public transportation (which, even if perhaps cheaper than ridesharing, is slower and less convenient because of the waiting times, crowded/unpleasant trips and the fixed drop-in/drop-off locations).

2. Drivers also obtain better utility by sharing a ride due to the split travel costs; specifically, the higher the occupancy of their vehicles, the lower the individual costs. Moreover, if high-occupancy vehicles, carrying two or more passengers, are favored by the administration, costs may be further reduced (e.g., thanks to reduced highway tolls and/or parking spaces at reduced fees). In our system, no route detours/waiting times are offered by drivers to pick up/drop passengers; hence, sharing the ride should not incur significant time overheads for drivers.

3. Most peers are reluctant to travel with strangers and, in most cases, this reluctance dominates the intrinsic utility benefits of ridesharing. Even so, peers are willing to share the ride with those that have shown reliable behaviors in past rides (i.e., those having a good *reputation*); that is, reputation management is a trust-enabling mechanism.

4. Potential passengers and drivers are concerned about their privacy, both regarding central matching agencies compiling their personal details, reputations and locations, and regarding other peers in the network. In some cases, these concerns may also dominate the utility benefits of ridesharing.

5. The dynamic ridesharing scenario we consider usually focuses on short trips (typically within urban areas), which are decided and organized online and on short notice [2]. This contrasts with long/medium-range ridesharing, in which rides are agreed upon *days* in advance and peers may also communicate face-to-face before the trip to organize it and/or build *trust* among them (which is not possible in dynamic ridesharing).

Notice that, even though ridesharing also provides environmental benefits likely to improve the welfare of peers in the medium term (e.g., reduced CO_2 emissions, reduced traffic congestion, etc.), we do not consider such benefits during the short-term decision making of dynamic ridesharing because the immediate individual cost and time benefits discussed above are arguably the most powerful incentives to collaborate.

According to Assumptions 1 and 2 above, we can formalize the utility u_i of a ridesharing peer $\mathcal{P}_i$ (either a driver or a passenger) as follows:

$$u_i = cs_i + \alpha_i^{ts} \cdot ts_i + \alpha_i^c \cdot c_i, \tag{1}$$

where: cs_i corresponds to the cost savings that peer $\mathcal{P}_i$ derives from sharing the ride w.r.t. traveling alone (for drivers) or taking a taxi (for passengers); ts_i corresponds to the time the passengers save w.r.t. traveling by public transport (ts_i is zero for drivers, but never negative, since we do not consider detours in ridesharing); c_i is the 'convenience' of ridesharing (for passengers, c_i will be always positive w.r.t. public transportation because they avoid waits and crowded trips and gain flexibility; for drivers, c_i could be zero or negative because of the burden of sharing their vehicles with other people); and the α_i^x coefficients weight the importance attached by each peer to each utility dimension and also allows aggregating their heterogeneous magnitudes.

According to Expression (1), we have that ridesharing is co-utile if both drivers and passengers obtain a positive u_i, which means that the utility they get is strictly greater than what they get from not sharing the ride and/or using an alternative transportation means; in that case, ridesharing provides mutual benefits and is a Pareto-optimal traveling solution. For passengers, this easily holds as long as traveling by taxi is significantly more expensive, and public transport is significantly slower and less convenient. For drivers, it also holds as long as they are not reluctant to share their vehicles. Therefore, ridesharing will be adopted by *rational* agents in a self-enforcing way.

However, as we discussed in the introduction, the concerns and reluctances identified in Assumptions 3 and 4 can affect the utility outcomes as follows:

$$u_i' = u_i - \alpha_i^p \cdot p_i - \alpha_i^r \cdot r_i, \tag{2}$$

where: p_i is the privacy loss the peers incur when they are required to reveal their identities and personal data to a central matching agency (in a centralized setting) or to make these data uncontrollably available to other peers in the network (in a decentralized setting); r_i measures the reluctance to share a ride with complete strangers due to lack of trust.

From the figures discussed in the introduction and the conclusions reported in ridesharing surveys [1, 17], the influence of the negative utilities p_i and r_i tends to dominate the positive utility gain u_i brought by ridesharing; if this is the case, the final utility outcome u'_i becomes negative and ridesharing is no longer Pareto-optimal (and, thus, co-utile), and the adoption of ridesharing is severely hampered.

The decentralized ridesharing system we present in the following sections is specifically designed to neutralize these two negative utilities: we ensure that the privacy of peers is preserved during the ride-matching process and we build trust among peers by means of a decentralized reputation mechanism. In this way, we aim at retaining only the positive utility described by Expression (1) and, thus, at making ridesharing co-utile.

3. Decentralized P2P ridesharing

Our system follows the *organized ridesharing* paradigm, by which ride-matching can be prearranged between participants (unlike hailing a cab or hitchhiking, which are *ad hoc*). However, it does not necessarily assume previous involvements between the participants [12] (in other words, passengers and drivers do not need to know each other).

Organized ridesharing has been operated by agencies that match passengers and drivers according to a specific matching algorithm [17]. Ridesharing matches are two-sided, since they are performed according to the offers and requests received from drivers and passengers, respectively. Our system provides the same type of ridesharing matching service, but without relying on a central matching agency. Specifically, we offer a fully decentralized P2P network in which passengers and drivers act as peers that dynamically enter the network to offer or request rides. Ride matches are managed by the peers themselves in such a way that only those peers whose ride offers/requests match disclose their personal details (i.e., specific locations and identities) to each other. This feature overcomes the privacy concerns that have hampered ridesharing services thus far [17], because users are no longer forced to disclose all their ride details to a centralized agency.

3.1. Ridesharing types

Ridesharing matching depends on the spatial and temporal features of ride offers and requests. In our system, each driver D has a trip route $R(D) = \{(l,t)_0^D, \ldots, (l,t)_n^D\}$ that is defined by an origin location and time $(l,t)_0^D$, a destination location and time $(l,t)_n^D$ and, optionally, an ordered set of the intermediate positions and associated (estimated) times $(l,t)_i^D$. Even if intermediate positions are optional, the more detailed the route (i.e., the more numerous and precise the intermediate points), the higher the chance to successfully match drivers' routes and passengers' requests, as it will be discussed below. Each passenger P, on the other hand, states his ride request by just specifying an origin location and desired pick-up time, and a destination, that is,

$R(P) = \{(l,t)_0^P, (l)_1^P\}$. We assume passengers have no specific route constraints between their origins and destinations (the specific path followed to reach destination does not matter to them).

Similar to the usual practice by matching agencies, we do not require ride offers and requests to perfectly match (regarding locations and times), because this would be too rigid and would severely restrict feasible matches. Instead, each passenger P defines spatial and temporal slacks that specify the maximum distance δ^P he accepts to walk (from his origin location l_0^P to the pick-up location by a driver, say l_i^D, and from the driver's drop-off location, say l_j^D, to his destination l_1^P), and the maximum time τ^P he accepts to wait before being picked up (with respect to his initial preference t_0^P). Temporal and spatial slacks are considered separately, so that the waiting time does not include the walking time to the pick-up location (note that the walking time is actually a distance: the distance that can be covered in a certain time when walking). Even so, the passenger may define both slacks so that walking distances match his time constraint (e.g., 10 minutes may allow him to walk around 500 meters) in order to better accommodate walking times within his time constraints. We do not consider spatiotemporal flexibility for drivers to pick up and drop passengers along the route because, as mentioned in Section 2, the resulting detours and delays may dilute the drivers' cost savings, especially in short routes (the typical ones in urban mobility).

Our system supports the usual types of ridesharing implemented by state-of-the-art matching agencies [17, 24]:

- *Identical ridesharing*: the origins and destinations of driver and passenger match both spatially and temporally (within the passenger's space and time slacks); that is, $|l_0^D - l_0^P| \leq \delta^P \wedge |l_n^D - l_1^P| \leq \delta^P \wedge |t_0^D - t_0^P| \leq \tau^P$.

- *Inclusive ridesharing*: the origin and destination of the passenger are included in the route $R(D)$ of the driver (within the passenger's space and time slacks); that is, $\exists (l,t)_i^D \in R(D) \mid |l_i^D - l_0^P| \leq \delta^P \wedge |t_i^D - t_0^P| \leq \tau^P \wedge \exists l_j^D \in R(D) \mid |l_j^D - l_1^P| \leq \delta^P$. With this ridesharing type, the driver does not need to deviate from her route, but she must stop at the pick-up and drop-off locations agreed with the passenger (l_i^D and l_j^D, respectively). As stated above, the more numerous and precise the intermediate positions/times of the driver's route, the higher the chance of finding an inclusive match with a passenger.

The above ridesharing types also support multiple passengers per vehicle (up to the maximum capacity c^D specified by each driver D for her vehicle). In inclusive ridesharing, the only difference between carrying one or several passengers is that in the latter case the driver must stop several times for pick-ups and drop-offs.

The literature also considers more complex ridesharing types. For example, in *partial ridesharing* the shared ride covers only a part of the passenger's trip, so that additional shared rides or transportation means need to be concatenated to complete the trip; in *detour ridesharing*, the driver may deviate from her route to accommodate the passengers' requests, at the expense of increased travel costs. Since these complex ridesharing types hinder instantaneous decision making by drivers and passengers and/or may require adding compensations for detouring, they are not currently offered by matching agencies [17] and we leave them for future work.

3.2. The P2P ride management network

Rather than using a central matching agency, our system consists of a fully decentralized P2P network of drivers and passengers. Given that decentralized P2P networks lack a trusted authority and a common legal framework binding all peers, our challenge here is to design a co-utile protocol for ridesharing (which in turn will rely on a co-utile protocol for reputation management), that is, a ridesharing protocol that is followed by rational peers because of the benefits it brings to them.

There are several alternative communications technologies to implement decentralized P2P applications (for example, CAN [13], Chord [30], Pastry [7], etc.). Among these, Pastry [7] is especially interesting because of its proximity-based routing. It consists of a self-organizing overlay network that provides a fault-tolerant and load-balanced distributed hash table (DHT) for large-scale P2P applications. Each node (i.e., a P2P user) is identified by a unique, randomly generated *nodeId* from a 128-bit identifier space, which acts as a pseudonym that hides the real identity (IP address) of the user. The DHT is internally used to map an existing *nodeId* to its associated IP address. In this manner, a message can be anonymously routed to any network user (via anycast) by forwarding it over a bounded number of hops, concretely, $< \log_{2^b} N$, where N is the number of nodes and b typically has the value of 4. Moreover, each node requires locally storing only $O(\log_{2^b} N)$ entries of the hash table. Proximity-based message routing is performed by sending each message to the numerically closest *nodeId* that the sender has in its local hash table. If the receiver is not the message addressee, it forwards the message to its closest *nodeId*, and the process is repeated until the message reaches its intended addressee.

3.3. Decentralized ride-matching protocol

To match their rides, drivers and passengers need to advertise their offers and requests, thereby incurring some information disclosure. As discussed above, privacy concerns about that disclosure may deter users from participating in the ride-matching protocol [17, 14]. Thus, to ensure that the protocol remains co-utile (and thus self-enforcing for rational agents), we need to keep the disclosure of the peers' spatiotemporal features as low as possible.

Within a decentralized network, a straightforward way to look for ride matches is to broadcast to all active peers a message with the passengers' requests and drivers' offers. However, this incurs a large information disclosure, and it may be even worse for privacy than using a centralized matching agency. Moreover, broadcast is very inefficient because it unnecessarily floods the network with messages that are useless to most peers. Instead, we propose a mechanism predicated on a topic-based decentralized subscription system implemented by the network by which i) users only communicate with those peers with *similar* travel needs/offers and ii) their concrete spatiotemporal data are only disclosed if a potential match is found. The idea is that drivers who offer rides *subscribe* to *topics* according to the routes they offer (topics are locally stored by the peers), and passengers *publish* their ride needs to the appropriate topic in the subscription system. Being decentralized, an event notification system for topic-based subscribe-publish applications is in charge of communicating via multicast only those peers whose rides match (i.e., drivers subscribed to a topic describing a route for which a passenger has published a message). Since the subscription information is local for each peer, topic subscriptions by drivers can only be known by publishing a message in a matching topic. Thanks to the proximity-based message routing and the topic-oriented multicast, the notification system is capable of scaling to a potentially large number of subscribers per topic and to a large number of different topics. Technically, the subscription infrastructure is built upon the decentralized Pastry P2P network to manage topic creation, topic subscription and dissemination of messages published in each topic.

Topics can be initially created (and subscribed to) by any peer (drivers, in our case) by defining a *topicId*. In our system, *topicId*s are defined according to the detail of the routes being offered by drivers, that is, all the possible rides consisting of a pick-up location/time and a drop-off location that can be defined from each route (that is, all subroutes of each route). Drivers offering rides equal to an existing *topicId* can register to the same topic to become subscribers of it. On the other hand, only peers knowing a *topicId* can publish messages in a certain topic. In our case, passengers publish messages in *topicIds* that match their ride needs (desired pick-up location/time and drop-off location). Thus, a passenger's message is disseminated only to the drivers subscribed to the matching topic, that is, to a ride offer that matches the passenger's needs; only in this case, peers may exchange their personal details (e.g., identities).

Although the subscribe-publish mechanism routes the messages only to the peers whose offers/requests match and the anonymity of peers is preserved unless they identify themselves to engage in a shared ride, topic subscription and publishing via detailed routes still discloses the spatiotemporal features of the users in the *topicIds*. Moreover, the number of different *topicIds* that can be defined according to *exact* locations and times may be unmanageably large. Also, too much detail may be counterproductive for the flexible matching described in Section 3.1: passengers ought to publish as many topics as locations and times included in their space and time slacks.

We tackle these issues by defining topics as *generalizations* of spatiotemporal features, rather than exact values. Specifically, exact locations l_i (both of passengers and riders) are generalized to zones within the area A (e.g. a city) where the ridesharing system will be deployed. Zones are fixed by a predefined partition $Z(A)$ of the area (e.g., districts or neighborhoods in a city, uniform partitions in a map, etc.). Figure 1 shows an example of such partition for the city of San Francisco, where each $Z_i \in Z(A)$ corresponds to the name of a neigh-

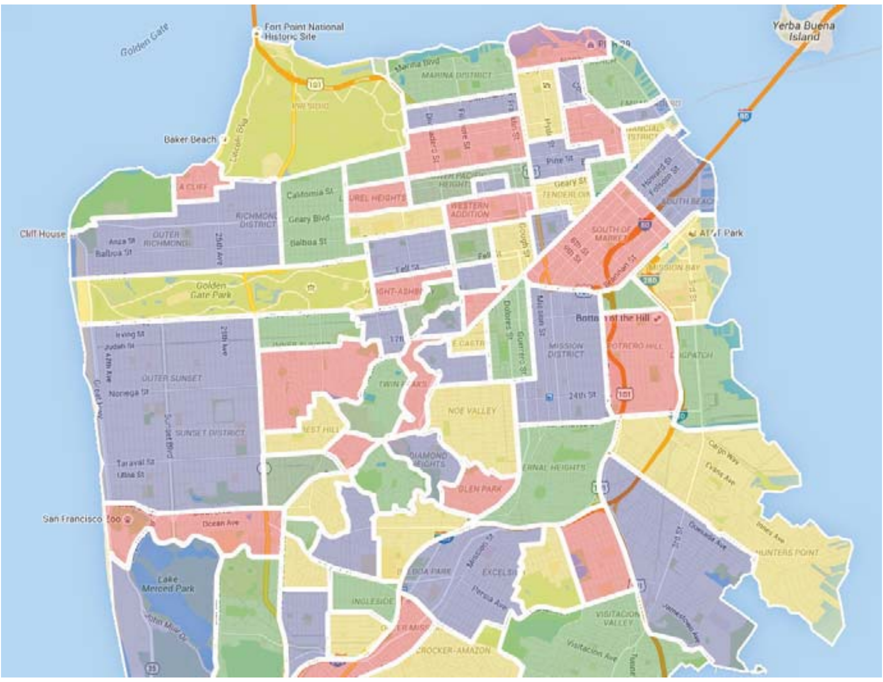

Figure 1: Example of zone partitioning for the city of San Francisco: zones correspond to neighborhoods

borhood. Likewise, concrete times t_i associated to positions l_i are generalized to fixed time intervals I (e.g., 10 minutes, a quarter of an hour, etc.). Specifically, since we measure t_i as the Epoch time (i.e., seconds since January 1st, 1970), the corresponding interval I is calculated as the integer part of the quotient between t_i and the size of the time interval (e.g., $\lfloor t_i/900 \rfloor$ for 15 minutes). Once generalized, each *topicId* is defined by the string resulting from concatenating the pick-up zone Z_o and time interval I_o with the drop-off zone Z_e. Since the generalization granularity in space and time is common to all peers in the network, the system designer should configure it beforehand in order to control the level of disclosure incurred. In fact, to optimize privacy, zones and time intervals should be large enough to encompass *several* peers, so that these become indistinguishable within a zone or an interval; in this way, we prevent unequivocal re-identification of individuals due to unique spatiotemporal features [29, 31].

Formally, the ride-matching protocol (depicted in Figure 2) is as follows:

1. Each driver D advertises the route $R(D)$ she would like to share by subscribing to as many topics as possible rides can be defined in her route. To do so, first D generalizes positions and times $(l,t)_i^D \in R(D)$ to zones and time intervals $(Z,I)_i^D$, respectively; we refer to the *generalized route* as $\bar{R}(D) = \{(Z,I)_0^D, \ldots, (Z,I)_n^D\}$. Then, D subscribes to all topics $S(D)$

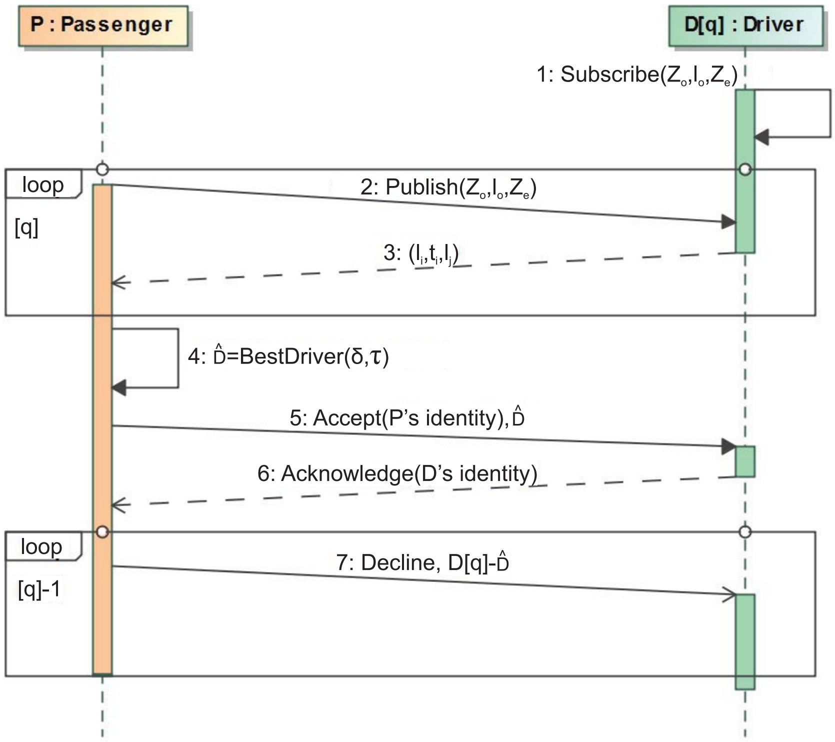

Figure 2: Sequence diagram of the ride-matching protocol based on a subscribe-publish mechanism

whose *topicIds* are triples (Z_o, I_o, Z_e), where $(Z_o, I_o) \in \bar{R}(D)$ is a generalized pick-up location-time pair and $Z_e \neq Z_o$ is a generalized drop-off location such that $(Z_e, I_e) \in \bar{R}(D)$ for some generalized time interval I_e. Note that, by considering all the possible rides within the driver's route, we support the *inclusive ridesharing* described in Section 3.1.

2. Each passenger P looking for a ride (with desired route $R(P)$) publishes his request in a list of topics $S(P)$ that includes *topicIDs* obtained in a way analogous to the *topicIds* subscribed by a driver (see above). Specifically, since $R(P) = \{(l,t)_0^P, (l)_1^P\}$, then $S(P) \supseteq \{(Z_o, I_o, Z_e)\}$ where Z_o and Z_e are the zones corresponding to the generalization of l_0^P and l_1^P, respectively, and I_o is the generalized time interval of t_0^P. If the passenger has flexible spatiotemporal requirements, $S(P)$ contains as many topics as zones and time intervals can be obtained by generalizing the locations and times resulting from adding/subtracting the passenger's space slack δ^P to locations in $R(P)$ and the passenger's time slack τ^P to the time in $R(P)$. As a result of the publication on these topics, only those drivers that have subscribed to the same topics will receive the passenger's request.

3. Each driver D who receives a passenger's request responds in case she is available with a direct message that discloses the precise pick-up and drop-off locations and pick-up time she generalized into the topic, that is, the specific $(l,t)_i^D$ and l_j^D. Let q be the number of drivers who respond.

4. Passenger P collects all the answers of the q matching drivers and selects the ride (and the respective driver) that actually matches his request considering also his spatiotemporal flexibility δ^P and τ^P, as detailed in Section 3.1. If several rides (drivers) match his request, P selects the best driver, $\hat{D}$, as the closest one to his actual origin and destination (in normalized average space and time).

5. Passenger P sends the selected driver $\hat{D}$ an "accept" message that reveals P's identity.

6. The driver $\hat{D}$ receiving the "accept" sends an "acknowledgment" to passenger P also revealing $\hat{D}$'s identity.

7. Finally, passenger P sends to the remaining $q-1$ drivers collected at Step 4 a "decline" message. Note that P sends "decline" messages only after receiving the "acknowledgment" message from the selected driver $\hat{D}$. However, if $\hat{D}$ does not reply to passenger P for some reason (e.g., $\hat{D}$ already received an "accept" from another passenger, sent him an "acknowledgment" during the protocol execution and reached her vehicle's maximum occupancy $c^{\hat{D}}$), P returns to Step 4 to select the next best driver.

To illustrate the above-described protocol, assume a driver D wants to share the route $R(D)$ across the north of San Francisco shown in Figure 3. $R(D)$

Figure 3: Example driver D's route, $R(D)$, with 7 points, and (flexible) origin and destination for a passenger P

Table 1: Points (locations coordinates and Epoch times) of the example route $R(D)$

$R(D)$	Location coordinates (l)	Epoch time (t)
$(l,t)_0^D$	37.783094, -122.462326	1455534795
$(l,t)_1^D$	37.784009, -122.457863	1455534882
$(l,t)_2^D$	37.784891, -122.448550	1455535206
$(l,t)_3^D$	37.786892, -122.434302	1455535487
$(l,t)_4^D$	37.788419, -122.422887	1455535851
$(l,t)_5^D$	37.790759, -122.419025	1455536448
$(l,t)_6^D$	37.791166, -122.417394	1455536552

starts in Richmond, ends in Nob Hill, and consists of seven points listed in Table 1.

Route $R(D)$ is first generalized to $\bar{R}(D)$ (as shown in Table 2) according to the zone partition $Z(A)$ of San Francisco shown in Figure 1 and time intervals of 10 minutes (i.e., $\lfloor t_i/600 \rfloor$).

To advertise the route, driver D subscribes to as many topics $S(D)$ as unique ordered combinations of pick-up/drop-off points can be derived from his generalized route $\bar{R}(D)$. Table 3 lists the triples (Z_o, I_o, Z_e) that constitute the *topicIds* D subscribes to.

Finally, let us consider that passenger P is looking for a ride with the following features

$$R(P) = \{(37.785146; -122.450288), 1455535117, (37.787215; -122.423938)\},$$

each one respectively corresponding to the points $(l,t)_0^P$ and $(l)_1^P$ shown in Figure 3. Then, the passenger publishes a message in the *topicId* corresponding to the generalized points of his request, that is,

$$S(P) = (\text{Lauren Heights}, 2425891, \text{Pacific Heights}).$$

There is a match between $S(D)$ and the 5th *topicId* subscribed to by D (Table 3). Therefore, the driver answers with the concrete details of that part of

Table 2: Generalized route $\bar{R}(D)$ using San Francisco neighborhoods as zones and time intervals of 10 minutes

$\bar{R}(D)$	Location zones (Z)	Time intervals (I)
$(Z,I)_0^D$	Richmond	2425891
$(Z,I)_1^D$	Lauren Heights	2425891
$(Z,I)_2^D$	Lauren Heights	2425892
$(Z,I)_3^D$	Pacific Heights	2425892
$(Z,I)_4^D$	Pacific Heights	2425893
$(Z,I)_5^D$	Nob Hill	2425894
$(Z,I)_6^D$	Nob Hill	2425894

Table 3: Set $S(D)$ of *topicId*s derived from the generalized route $\bar{R}(D)$ that driver D subscribes to

Z_o	I_o	Z_e
Richmond	2425891	Lauren Heights
Richmond	2425891	Pacific Heights
Richmond	2425891	Nob Hill
Lauren Heights	2425891	Lauren Heights
Lauren Heights	2425891	Pacific Heights
Lauren Heights	2425891	Nob Hill
Lauren Heights	2425892	Pacific Heights
Lauren Heights	2425892	Nob Hill
Pacific Heights	2425892	Pacific Heights
Pacific Heights	2425892	Nob Hill
Pacific Heights	2425893	Nob Hill
Nob Hill	2425894	Nob Hill

her ride, that is,

$$(l_2^D, t_2^D, l_4^D) = ((37.784891, -122448550), 1455535206, (37.788419, -122.422887)).$$

Since the driver's locations and time are within the passenger's spatiotemporal flexibility range given by δ^P and τ^P, the match is successful.

Note that all the steps of the protocol can be automatically managed without the intervention of the users. In Step 1, once drivers have defined their routes, subscription to the appropriate topics is deterministic and straightforward. Moreover, even if drivers just define the starting location and time and the destination of their rides, the system may also estimate the intermediate points and times of the route using a standard route planning algorithm. In Step 2, multicast messages are also automatically managed by the underlying network. In Step 3, response messages can be automatically created by de-generalizing zones and times and in Steps 4 to 6, the best match can be straightforwardly agreed upon according to the passenger restrictions.

A driver D can also unsubscribe from topics when, for some reason, her route changes or when the maximum capacity c^D of her vehicle is reached. Likewise, subscriptions are automatically removed from the system when the ride's time interval I is reached.

From a co-utility perspective, the protocol minimizes by design the privacy loss p_i mentioned in Section 2 (Expression (2)) because: i) we do not rely on any central agency that systematically collects private data; ii) users remain anonymous behind a randomly assigned *nodeId* and do not disclose their identities until a successful match is found; iii) exact travel data are only exchanged between those peers whose ride offers/requests match (via subscription-driven multicast); and iv) drivers' subscriptions in the P2P network only provide generalized information about their routes that is learned by publishing a request in a matching topic. One may certainly conceive systematic attacks whereby malicious users, behaving as passengers (resp. drivers), publish in (resp. subscribe to) topics with fake offers (resp. requests). However, since this attack should be systematic in order to obtain a significant amount of detailed data (i.e., complete drivers' routes, lists of passengers), peers can easily detect the attack pattern and decline communication with suspect peers.

4. Decentralized reputation management

As discussed in Section 2, the reluctance of drivers and passengers to share a trip with strangers, r_i, is another great hurdle to ridesharing. A natural way to mitigate the mistrust of users when they lack direct experiences with their peers is to use a *reputation system*. Reputation, which captures the opinion of the community on each agent, has at least two positive effects [14]:

- Reputation allows agents to build *trust*, which can neutralize the negative utility r_i related to mistrust. The higher an agent's reputation, the more trusted she is by other agents.

- Reputation makes agents accountable for their behavior: if an agent misbehaves (e.g., during the shared ride or by abusing/attacking the ride-matching system as described above), his reputation worsens and the other agents mistrust him more and more and are less and less interested to interact with him. In this manner, malicious agents (who may try to subvert the system, even irrationally) may be identified (via a low reputation) and penalized (e.g., through limitation or denial of service).

In the following, we describe our proposal to incorporate a decentralized reputation management protocol into our P2P network. The protocol we use is designed so that it is itself co-utile, and hence even purely selfish agents are interested in following it. In this way, this protocol can be seamlessly used as a mechanism to enforce co-utility in protocols in which negative utilities would otherwise rule it out [14].

In ridesharing, reputation is understood as the mutual feedback by drivers and passengers on each other. Such feedback may consist of the aggregation of several objective and subjective outcomes (e.g., how pleasant the trip was or whether there were any unnecessary delays or overcosts). This reputation model is inherited from commercial platforms (like eBay) that successfully support buyers and sellers in building trust to unknown peers and eliciting honest behaviors [28].

Even though some ridesharing agencies (e.g., Carma, Carpool World, Golco) have incorporated reputation systems, they always manage reputations in a centralized way. Centralized reputation is straightforward to implement, but it adds to the privacy loss caused by centralized ridesharing: agencies learn not only where users go, but also what their reputations are.

To overcome this problem in a way that is coherent with the decentralized nature of our ridesharing management system, we propose a fully decentralized reputation management protocol. Moreover, our protocol should ensure that it is in the best interest of rational agents to follow it and report truthful reputation information (this is in fact the most critical issue in reputation systems [21]). In other words, *the reputation management protocol should be itself co-utile.* This aspect is crucial if we want to use reputation management as a catalyst to make ridesharing co-utile by neutralizing negative utilities (i.e., concerns about strangers).

For reputation management to be co-utile, the agents involved in the reputation calculation protocol should not benefit by deviating from it. Otherwise, since agents are assumed to be selfish and rational, they will deviate. Deviations in the calculation of reputations that provide benefits to their orchestrators are usually referred to as “rational attacks” [19]:

- *Self-promotion*, whereby agents are able to illegitimately increase their own reputations at a small or zero cost.

- *Whitewashing*, in which agents circumvent the consequences of abusing the system to obtain an unfair benefit, for example by creating new “clean” identities or performing Sybil attacks.

- *Slandering*, by which agents may falsely lower the reputation of other agents if, by doing so, their own reputations become comparatively higher.
- *Denial of service*, in which agents block the calculation and dissemination of reputation values. This may happen, for example, if the reputation calculation has a cost that agents deem higher than the benefits the calculation of their own reputation (by other agents) would bring to them.

Desirable features of the reputation calculation protocol that help withstand the above rational attacks are:

- *Anonymity*. Reputation management should not rely on personal identifiers (e.g. IP addresses) that reveal the real identity of agents who contribute to computing the reputation of other agents. Otherwise, the privacy loss may negatively affect the positive payoffs of collaborating in the reputation management. Moreover, the possibility of creating coalitions between agents that may know each other would facilitate collusion attacks to the reputation system (like self-promotion or slandering).
- *Low overhead*. Reputation management should not require a large expenditure of resources (e.g., bandwidth, storage, calculation); otherwise, these negative payoffs may dominate the benefits brought by reputation, thereby leading to denial of service.
- *Proper management of new agents*. Newcomers should not enjoy any reputation advantage; otherwise, malicious peers may be motivated to create new anonymous identifiers after abusing the system in order to regain the advantages of a good reputation (whitewashing).

4.1. Measuring reputations

After examining a number of decentralized reputation mechanisms available in the literature [19], we implemented a version of the well-known EigenTrust protocol [22], because it offers most of the desirable features identified above; the version we use here is the extension we propose in [14], which ensures that the reputation calculation protocol is itself *co-utile*. The plain EigenTrust protocol is designed to filter out inauthentic content in peer-to-peer file sharing networks and offers many of the desirable features identified above: decentralized (i.e., distributed) reputation calculation, low overhead, anonymity and robustness to some attacks. Our extended version [14] generalizes EigenTrust by accomodating non-binary opinions on transaction outcomes as input, and it provides a more secure distributed calculation that cancels the benefits of deviating from the protocol. In this way, it achieves co-utility.

The basic idea of both EigenTrust and our extension is to calculate a global reputation for each agent based on aggregating the local opinions of the peers that have interacted with the agent. If we represent the local opinions by a matrix whose component (i, j) contains the opinion of agent i on agent j, the distributed calculation mechanism computes global reputation values that approximate the left principal eigenvector of this matrix.

For the distributed calculation protocol to work properly, it is important for local reputation values computed by the different peers to rate similar transaction outcomes in a similar way. For objective outcomes, this is not problematic; however, for subjective opinions the reputation system may provide some rules to control the ratings. Also, since in ridesharing both drivers and passengers may be reluctant to share the ride with strangers, ratings should be reciprocal. Moreover, for the reputation calculation to work properly, neutral rides should be rated as 0, satisfactory ones should receive positive scores and unsatisfactory ones should get negative scores. For example, passengers may rate a pleasant ride with fair cost sharing and no delays as +1 for the driver, whereas an unpleasant ride, unfair costs and/or unnecessary delays can be rated as -1; likewise, drivers may rate passengers according to the pleasantness of their company and/or their punctuality.

According to these ratings, the opinion of agent $\mathcal{P}_i$ (either a passenger or a driver) on another agent $\mathcal{P}_j$ (a driver or a passenger, respectively) with whom $\mathcal{P}_i$ has shared a ride is the reputation s_{ij} of $\mathcal{P}_j$ local to $\mathcal{P}_i$. This value is defined as the aggregation of ratings (positive or negative) that $\mathcal{P}_i$ has issued as a result of the set Y_{ij} of rides shared with $\mathcal{P}_j$:

$$s_{ij} = \sum_{y_{ij} \in Y_{ij}} rating_i(y_{ij}).$$

In order to properly aggregate the local reputation values computed by each peer, a normalized version c_{ij} is first calculated as follows:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}.$$

In this manner, the normalized local reputation values lie between 0 and 1 and their sum is 1. In other words, each agent has a reputation budget of only 1 that he has to split among his peers proportionally to his positive experiences (negative experiences are truncated to 0). This makes all agents equal contributors to the global reputation and avoids dominance by agents with a larger number of experiences. Moreover, this normalized calculation deters peers from colluding by assigning arbitrarily high values to each other. Finally, the fact that negative reputation values are truncated to 0 prevents selfish agents from assigning arbitrarily low values to good peers. This deters the *self-promotion* and *slandering* attacks during the rating step.

A side effect of the truncation of negative values is that reputation values do not distinguish between agents with whom $\mathcal{P}_i$ had a bad experience (negative local reputation) and those with whom $\mathcal{P}_i$ has not interacted so far. In this manner, newcomers do not have any reputation advantage, because their reputation is indistinguishable from the one of misbehaving agents. As a result, a selfish agent has no incentive to take a new virtual identity in order to "clean" his reputation after misbehaving with other peers (i.e., the *whitewashing* attack is prevented). Likewise, newcomers become instantly motivated to positively contribute to the system in order to earn the minimum reputation that other agents would require from them.

We can thus see that normalizing local reputation values biases the system towards positive reputation; that is, agents need a minimum *positive* reputation value in order to be trusted by peers.

Local reputation values are then disseminated and aggregated through the network peers by following the transitive reputation calculation algorithm we describe in [14], in order to obtain the global reputation value g_i of each agent $\mathcal{P}_i$. To make it robust against self-interested attacks (i.e., *self-promotion* or *slandering*), the reputation value of an agent $\mathcal{P}_i$ is computed by *several* other agents, named *score managers*.

Once a score manager computes the reputation of another agent, he keeps the reputation value for that agent until the next protocol execution. The reputation calculation protocol is meant to be run periodically, in order for reputations to stay up-to-date. The reputation update period can be set depending on the activity of the agents, in order to obtain faster updates when the frequency of agent interactions increases. Ideally, the protocol should be run in parallel and asynchronously with respect to the ride-matching protocol.

Under the co-utility framework, rational (selfish) agents want their reputations to be correctly computed (by the score managers responsible for that); moreover, they are interested in maximizing their reputation by any means (either by correctly following the protocol or by deviating from it). From the discussion above, we can see that agents cannot increase their own reputations by deviating from the reputation calculation protocol (i.e., attacks are either ineffective or easily identifiable and punishable). Thus, since collaborating in the reputation calculation protocol is mutually beneficial for all agents (i.e., they get as high a reputation as their actions deserve) and they cannot increase their own reputation by deviating, following the protocol is the only rational choice and, thus, it is co-utile. Hence, if we consider that most (if not all) agents in the network are rational, the co-utility of the protocol deters malicious behaviors. A more formal step-by-step discussion of the co-utile nature of the reputation calculation protocol and of its robustness against attacks is given in [14].

4.2. Incorporating reputations into the ride-matching protocol

In ridesharing interactions, the lack of trust between driver and passenger is mutual and may hamper collaboration (drivers may be reluctant to share their cars with strangers, and passengers may be concerned about sharing a ride with a driver they do not know). With our reputation calculation mechanism, once global reputation values have been computed, it is immediate for any agent $\mathcal{P}_i$ (either driver or passenger) to learn the reputation value g_j of $\mathcal{P}_j$ (passenger or driver, respectively) in order to decide whether to ride with $\mathcal{P}_j$ or not. Agent $\mathcal{P}_i$ can query the score managers of $\mathcal{P}_j$ for the latter's reputation; the resulting values obtained from the score managers should be the same, because the inputs of the score managers are the same. However, if some values differ (e.g., if some score managers or agents involved in the calculation have altered the computation for some -non-rational/random- reason), $\mathcal{P}_i$ can take as g_j the most common value among the ones sent by the score managers.

In addition to the global reputation of $\mathcal{P}_j$, agent $\mathcal{P}_i$ may also rely on his direct experiences with $\mathcal{P}_j$, if any, which are reflected in his local normalized value c_{ij}. In some cases, local and global reputation values may not be coherent because the latter are the aggregated version of the former. Thus, for a more robust trust enforcement, it is better for $\mathcal{P}_i$ to consider both local reputations and global reputations and take the lowest value in order to make decisions about collaboration. In this manner, agents are discouraged from selectively behaving well with some agents while misbehaving with others.

Additionally, individual agents may have different levels of reluctance vs. strangers and, thus, may require different reputation levels to build trust and agree to collaborate with other agents. To model this notion, we may allow each agent $\mathcal{P}_i$ to specify a *minimum reputation* threshold ρ_i that any $\mathcal{P}_j$ should have in order for $\mathcal{P}_i$ to ride with $\mathcal{P}_j$. Since ridesharing is predicated on reciprocal trust, it will only happen if both the driver's reputation g_D and the passenger's reputation g_P are above the other party's reputation threshold (i.e., $(g_D \geq \rho_P) \wedge (g_P \geq \rho_D)$).

In summary, the ride-matching protocol detailed in Section 3.3 can be easily extended to enforce trust among the agents by considering the peers' reputations in Steps 3 and 4. In Step 3, the drivers receiving a passenger P's request may evaluate P's reputation and, only if it is above their desired respective reputation thresholds, the drivers will answer with their respective ride details. Likewise, in Step 4, passenger P will select the best ride (driver $\hat{D}$) only from those drivers with a reputation above his desired reputation threshold ρ^P. In terms of information disclosure, we can also see that with this reputation-aware protocol a passenger and a driver reveal their identities to each other (in Steps 5 and 6, respectively) *only if they both trust each other*. Finally, reputations can also be used to punish agents that are suspected of abusing/attacking the system. Specifically, in the disclosure attack we detailed in Section 3.3, malicious passengers (resp. drivers), may publish in (resp. subscribe to) topics with fake offers (resp. requests) in order to gather personal data from their peers. However, according to our ride-matching protocol, the actual disclosure (detailed locations and identities of peers) only happens when the ride is accepted (Step 5), both for drivers (who disclose their identity in Step 6) and for passengers (who implicitly disclose their approximate location by accepting the ride in Step 5). If the request/offer was fake (that is, it was orchestrated just to gain some information about random drivers/passengers matching the offer/request), the malicious agent will accept as many rides as matches he gets, because this is the only way to obtain the drivers' identities together with their locations or to confirm passengers' data. However, all affected peers will be immediately aware of the attack, because the ridesharing will not happen. These peers will then punish the malicious agent with a low reputation rating and, since this low rating comes from many peers, the attackers' reputation will significantly and rapidly decrease. As a result of his low reputation, further fake offers/requests by the attacker are likely to be rejected/ignored by the other peers, thereby preventing the attacker from gathering more personal data (and also from benefiting from the system in general). By using reputation as punishment, we make peers aware of the negative consequences of attacks and prevent rational agents from attempting them.

All these enhancements brought by the reputation mechanism make following the ride-matching protocol the only rational choice and, thus, they enforce co-utility even if peers do not know each other.

5. Empirical study

This section reports the results of an empirical study that simulates realistic rides in our P2P network. Data about rides have been extracted from the cab mobility traces provided by the Exploratorium museum within the cabspotting project[2]. This data set contains the traces of the trips of approximately 500 cabs during May 2008 in San Francisco Bay Area [27], which match the usual features of dynamic ridesharing (i.e., short rides in urban areas planned on short notice). Each trace is defined as the GPS coordinates and the absolute times of its origin, destination and intermediate positions measured every 10 seconds (on average).

5.1. Configuration of the experiments

In our experiments we used the mobility traces of a whole week to simulate ride offers (by drivers) and requests (by passengers). We considered only those traces in which the cab was occupied by a customer, because these are the ones corresponding to realistic routes with meaningful and precise origins and destinations. In contrast, we omitted cabs wandering in search of customers, because this may result in seemingly random routes. We also omitted very short routes of less than 500 meters (for which sharing a ride is not worth while) and/or those with less than 4 position measurements. The resulting set contains 94,070 traces. Figure 4 (left) shows that the traces are uniformly distributed through the week (with a small dominance of Saturdays), whereas Figure 4 (middle) shows that they are concentrated during working hours ([8h-18h]) and the leisure time ([19h-1h]). Likewise, Figure 4 (right) shows that the length of most rides is 3 km or less.

We simulated three well-differentiated scenarios by randomly assigning a percentage of the traces to drivers (ride offers) and passengers (ride requests), as follows:

- *Balanced scenario*: 50% of the traces (47,035) are assigned to drivers' offers and 50% (47,035) to passengers' requests.
- *Driver-dominated scenario*: 70% (65,849) of the traces are assigned to drivers' offers and 30% (28,221) to passengers' requests. This is a favorable setting for passengers, since they can choose from a wide range of offers.

[2] `http://www.exploratorium.edu/id/cab.html`

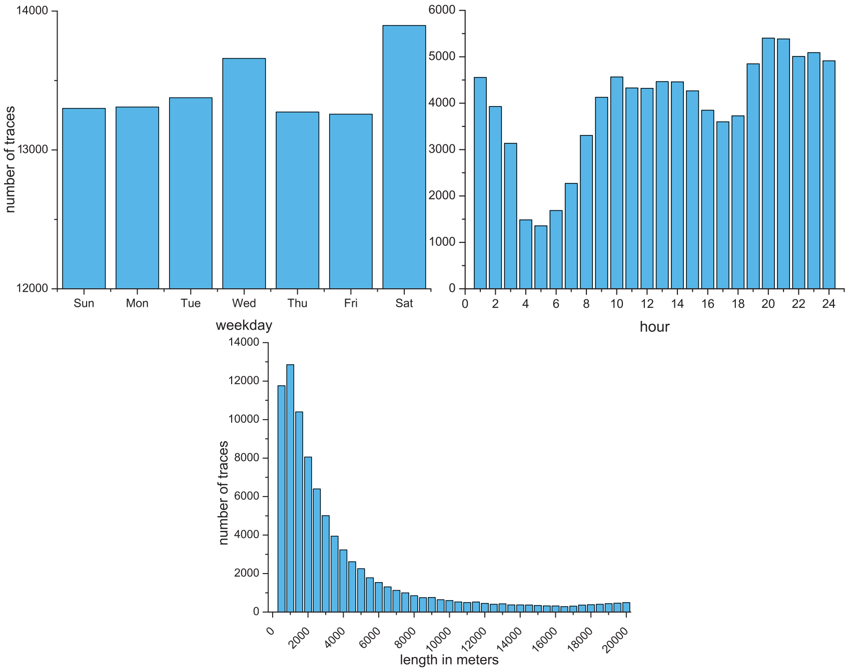

Figure 4: Distribution of the mobility traces used in our experiments: by days of the week (top left), by hours of the day (top right) and by length (bottom)

- *Passenger-dominated scenario*: 30% (28,221) of the traces correspond to drivers' offers, whereas 70% (65,849) correspond to passengers' requests. This favors drivers, since they can choose from more passengers. Since there are less drivers than passengers, ridesharing may involve sharing the same trip with several passengers, which also contributes to reducing travel expenses for each ridesharer.

For each driver D, we used all the measurements of her mobility trace (positions and times) in the data set to define her route $R(D)$. For each passenger P, we used only the initial position and time of his trace to define l_0^P and t_0^P, respectively, and the final position to define l_1^P. We also set different levels of passenger flexibility regarding the pick-up/drop-off locations and times: the maximum distance δ^P that a passenger P is willing to walk (with respect to the pick-up and drop-off locations) has been set to $\delta^P = \{250, 500, 750, 1000\}$ meters, whereas the maximum waiting time τ^P for pick-up has been set to $\tau^P = \{5, 10, 15, 20\}$ minutes. The maximum values of both dimensions are coherent with recent surveys on carpooling [6], which show that ride partners are usually found within one km radius (roughly 20 minutes walk time) of their current locations.

The management and matching of rides have been implemented as detailed in Section 3.2: drivers define their ride offers in a privacy-preserving way by subscribing to the set of topics defined by the city zones and generalized time intervals corresponding to their routes, and receive requests from potential passengers whose (generalized) ride requirements match. City zones for San Francisco Bay have been taken to be neighborhoods and generalized time intervals have been taken as 10-minute intervals. Since the sub-routes defined by the intermediate positions of the rides are also considered (during the subscription and ride matching), both the *identical* and the *inclusive* ride-matching types detailed in Section 3.1 have been implemented. Finally, we set a maximum occupancy per vehicle of $c^D = 4$ passengers (plus the driver).

The results of the simulations for the different scenarios have been evaluated according to the following metrics:

- Percentage of passengers that found a matching driver's offer. This metric reflects the functional success of ridesharing for passengers because, if passengers cannot find drivers, they will be forced to look for an alternative conveyance (e.g., public transportation, cab or their own private vehicle).

- Average occupancy (in number of passengers other than the driver) of the vehicles. This represents the economic success of ridesharing, both for the driver (the higher the occupancy, the more people she can share costs with) and also for the passengers (who will each bear a lower cost). For simplicity, we measured the occupancy as the number of different passengers that share (a part of) the ride through the entire driver's route.

- Percentage of kms saved by sharing rides compared to all agents (both passengers and drivers) traveling separately (for example, with their own private vehicles). This metric measures the boost of social welfare resulting from ridesharing, since saved travel kms imply a reduction of CO_2 emissions.

Our system is a decentralized P2P network implementing fully distributed algorithms (both for ride matching and for reputation calculation). Hence, the CPU and bandwidth load are balanced among peers, thereby achieving a good scalability. In fact, we use the Pastry P2P network [7] that is intended for file sharing and is designed to accommodate large communities of users exchanging data pieces (file fragments) much bigger than the (small) messages we employ to match rides. Thus, despite the large figures involved in the simulations (number of peers, topic subscriptions, published messages, etc.), the performance of our system remains very acceptable (see results below).

5.2. Results

Figure 5 shows the results for the three scenarios introduced above according to the three evaluation metrics. Regarding the percentage of passengers that found a matching ride, results are proportional to i) the ratio of passengers/drivers (i.e., the more drivers are available, the higher the chance of finding a match) and ii) the passengers' flexibility regarding time and locations. Flexibility turns out to have the greatest influence on the results, as the percentage of matches increases by 6-9 times between the least flexible and the most flexible requirements (i.e., 14.8% vs. 82.74%, 11.5% vs. 78.6% and 7.82% vs. 69.56% for the driver-dominated, balanced and passenger-dominated scenarios, respectively). Since most ridesharing solutions available in the literature [17] assume that passengers and drivers are matched by proximity rather than by exact position/times, we can see that (flexible) ridesharing has a high potential.

Regarding vehicle occupancy, results are opposite: the more passengers available, the more occupied are vehicles, which results in more saving for everyone. Differences are more significant when comparing the different scenarios: an average 1.623 passengers/vehicle is achieved in the passenger-dominated scenario vs. an average 0.355 in the driver-dominated one. However, even the higher average occupancy is still far from the maximum occupancy of 4 passengers/vehicle. Like matchings, occupancy increases very much with the passengers' flexibility, specifically by a factor 6-9 between the least flexible and the most flexible requirements.

The percentage of kms saved as a result of sharing rides is perfectly proportional to the number of passenger matches (the more matches, the more passengers avoid resorting to their own private cars). For the driver-dominated scenario with the maximum passenger flexibility, as many as 65% kms are saved (130,588 kms in absolute figures). In terms of social welfare and extrapolating the weekly data we considered in our experiments to a one-year period, saving 65% kms in this scenario represents an annual saving of around 739 thousand gas liters (assuming a consumption of 0.1089 l/km [25]) and a reduction of 1.73 tons of CO_2 emissions (assuming that 255.38 grams of CO_2 are emitted per kilometer [26]).

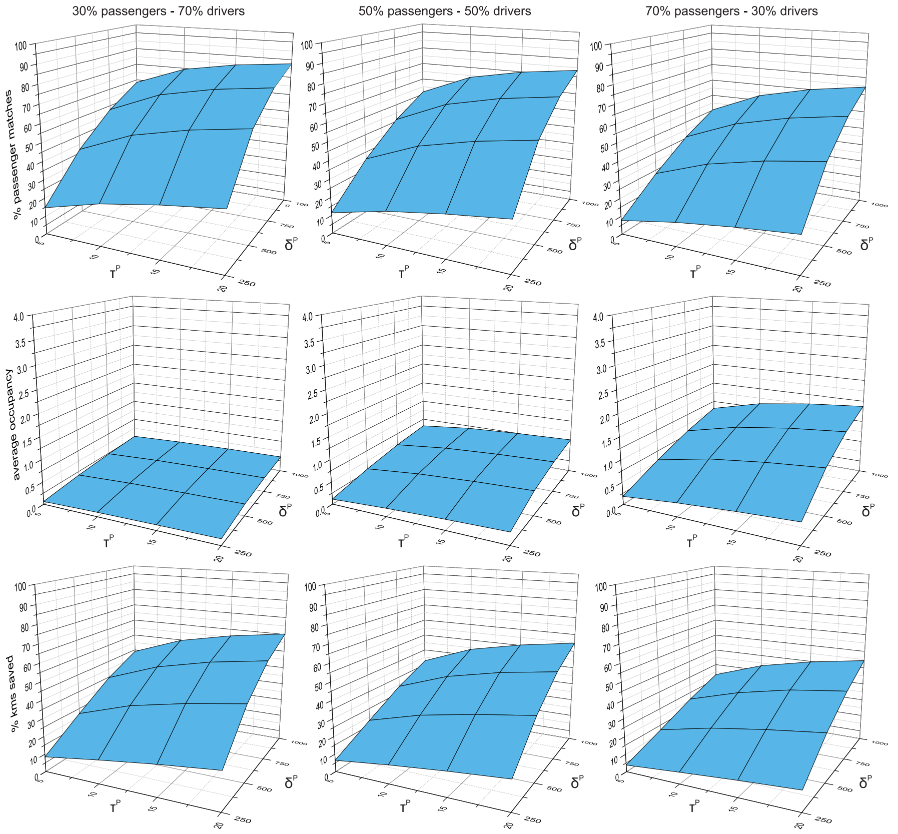

Figure 5: Percentage of passengers that found a matching ride (top), average occupancy of vehicles (middle) and percentage of kms saved (bottom) as a result of ridesharing for the three scenarios (left: driver-dominated; center: balanced; right: passenger-dominated)

Table 4: Percentage of passengers that found and accepted a matching ride, and vehicle occupancy for different scenarios of trust and reputation requirements

Scenario	P's matches	D's vehicle occupancy
Trusted	82.74%	0.355
Untrusted (no reputations)	5.79%	0.025
Untrusted ($\rho \geq 0.8$)	11.88%	0.051
Untrusted ($\rho \geq 0.6$)	28.28%	0.121
Untrusted ($\rho \geq 0.4$)	46.14%	0.198
Untrusted ($\rho \geq 0.2$)	64.54%	0.277
Untrusted ($\rho \geq 0.1$)	73.25%	0.314

5.3. Lack of trust and reputation management

In the previous simulations, we have assumed that peers do not mind sharing rides with strangers. In this section, we drop this assumption and evaluate the effectiveness of the reputation management mechanism proposed in Section 4 to cope with mistrustful peers.

As a baseline, we focus on the most favorable scenario for passengers, the driver-dominated scenario with maximum passenger flexibility. In the second row of Table 4 we report both the percentage of passengers that found a matching ride (very high: 82.74%) and the average vehicle occupancy in the ideal *Trusted scenario* in which all agents trust each other (or at least do not mind sharing rides with strangers). However, as discussed in [10] only 7% of the respondents seem to be willing to accept rides from strangers. The third row of Table 4 simulates a completely *Untrusted scenario* in which only that 7% of passengers will accept rides with drivers and in which the effectiveness of ride matching is severely hampered (only 5.79% of matches). Finally, the remaining rows simulate *Untrusted scenarios* in which agents rely only on reputations to decide whether to share or not a matching ride. In all cases, each agent $\mathcal{P}_i$'s global reputation g_i has been randomly chosen in the range $[0, 1]$, which corresponds to the normalized reputation range of the calculation mechanism depicted in Section 4. Then, different scenarios have been defined by setting a minimum and equal reputation requirement ρ for all agents in the system, $\rho = \{0.1, 0.2, 0.4, 0.6, 0.8\}$; that is, if we set $\rho \geq 0.4$, by following the reputation-based protocol depicted in Section 4.2, a matching ride would only be accepted if both the driver and the passenger have a global reputation value $g \geq 0.4$. Due to the reputations being randomly assigned, for consistency all the reported results are the average of 5 runs. We can see that in this *Untrusted scenario*, the use of reputations builds trust among the agents and increases the effectiveness of ridesharing in a way that is inversely proportional to the reputation requirements of the agents.

Finally, we simulated a more realistic and heterogeneous scenario in which each agent $\mathcal{P}_i$ is assigned a random global reputation g_i and also a random reputation requirement ρ_i with respect to the other peers, both uniformly distributed within the reputation value range [0..1]. Then, we measured how the global reputation values of peers affected the success of passengers in finding a

Table 5: Passenger's average waiting time and walked distance, and driver's average vehicle occupancy for matching peers according to their global reputation values

Peers' global reputation	P's wait. time	P's walk. distance	P's matches	vehicle occup. for matched D	D's matches
[0.0 . . . 0.25)	478.99 s	297.95 m	13.83%	1.09	5.97%
[0.25 . . . 0.50)	409.25 s	293.60 m	24.84%	1.24	17.61%
[0.50 . . . 0.75)	368.17 s	280.20 m	29.15%	1.41	30.30%
[0.75 . . . 1.0]	342.36 s	268.70 m	32.18%	1.65	46.12%

ride and the success of drivers in finding passengers. Moreover, for passengers we also measured the average distance they needed to walk from their origin to their pick-up location (and also from the drop-off location to their destination) and the time they had to wait for pick-up. For drivers, we measured the average vehicle occupancy for those drivers that were able to find a match. For both passengers and drivers, this was measured according to their global reputation values. Results are shown in Table 5, both for passengers and drivers, for (global) reputation quartiles; again, reported figures are the average of 5 simulation runs.

It is clear from Table 5 that, in a scenario with uniformly distributed reputations and reputation requirements, the higher the reputation of the peers, the higher their chance of finding other peers with matching rides that *trust* them. This is also reflected in the higher vehicle occupancy achieved by drivers with high reputations: they are able to find even more than one matching passenger. On the other hand, passengers with high reputations also fare better: they need to walk shorter distances to/from the pick-up/drop-off location and they wait shorter times for pick-up. The explanation is that passengers with higher reputations can select the best matching offer in spatiotemporal terms from a wider range of offers (at Step 4 of the ride-matching protocol detailed in Section 3.3).

This last aspect strengthens reputation as the mechanism to enforce co-utility in ridesharing: if both passengers and drivers are aware that a higher reputation will allow them to find more (and better) matches, it is in their own interest to behave well with their peers in order to increase their reputations. In other words, *reputation becomes a new, secondary utility that agents wish to optimize*, along with the primary time and cost utilities. In turn, by promoting the good ridesharing behaviors that peers rate positively (like avoiding unnecessary delays or costs, or providing a pleasant company), we make peers' collaboration more harmonious and sustainable.

5.4. Influence of ridesharing variables

In this section, we evaluate the effectiveness of the ride-matching algorithm with respect to the following variables:

- *Community size.* Dynamic ridesharing requires matching drivers and passengers on very short notice. This constraint requires a critical mass of

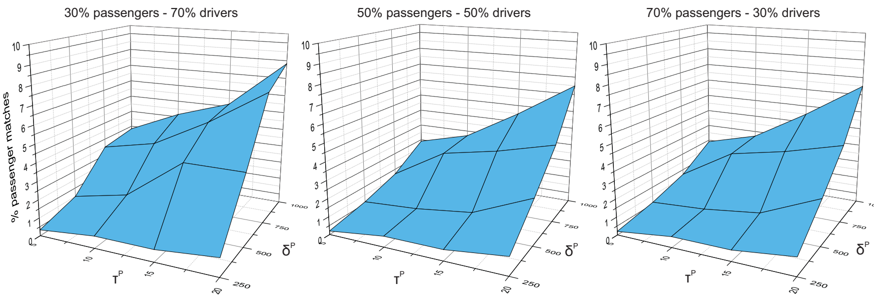

Figure 6: Percentage of passengers that found a matching ride as a result of ridesharing for the three scenarios (left: driver-dominated; center: balanced; right: passenger-dominated) with a small community (1,000 traces)

drivers, so that new requests by passengers can be successfully matched almost in real time [3]. In our previous experiments, we assumed that a large community of drivers and passengers was available (94,070 traces). In the first experiment of this section, we test the effectiveness of ride matching when dealing with a much smaller community consisting of just 1,000 traces randomly sampled from the dataset.

- *Trip length.* Dynamic ridesharing usually involves short trips within urban areas, which are the dominant ones in the dataset we use. However, other ridesharing models consider longer trips (i.e., across different urban areas). In the second experiment of this section, we test the effectiveness of the ridesharing types we consider (which are the usual ones in dynamic ridesharing [3]) when using only the longest trips of the dataset (i.e., those longer than 10 kms).

- *Granularity of generalizations.* The privacy of peers is protected in our system by generalizing their spatiotemporal features before agreeing on a ride. The granularity of the generalizations affects the level of disclosure and, as explained in Section 3.3, the designer should configure it (for example, according to the density of peers in the zones) in order to prevent unequivocal disclosures. However, very large (and, thus, very privacy-preserving) generalizations are not desirable, since they will increase the number of unmatchable offers received by the passengers (that is, offers such that the generalizations for drivers and passengers match but the actual locations and times do not). In this third experiment, we test the behavior of the ride-matching algorithm for different generalization granularities.

In the first two experiments, we measured the percentage of passengers that found a matching driver (which is the most representative figure of ridesharing effectiveness) for the three scenarios described in Section 5.1.

Figure 6 shows that, with a community around 100 times smaller than in the former experiments, the chance of finding a match is significantly (but not

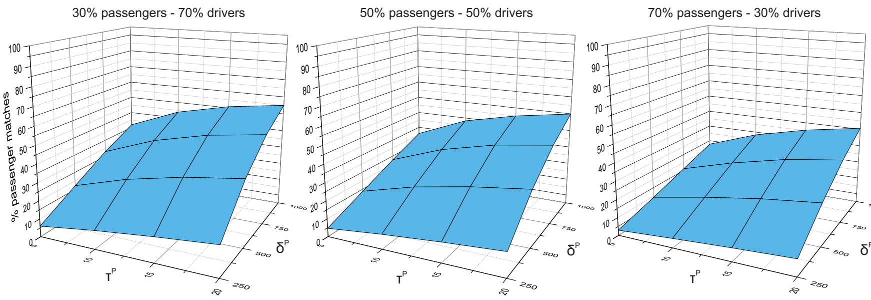

Figure 7: Percentage of passengers that found a matching ride as a result of ridesharing for the three scenarios (left: driver-dominated; center: balanced; right: passenger-dominated) with trips longer than 10 kms

linearly) lower. For example, in the driver-dominated scenario, which is the most favorable one to find matches, the percentage of matches decreases from 14.8% to 0.33% (around 45 times lower) for passengers with the least flexible requirements, and from 82.74% to 8.33% (around 10 times lower) for the most flexible passengers; we can thus see that in small communities the flexibility of passengers is more important than in large communities (experiments in previous sections), because a higher flexibility yields a much sharper increase of matches.

At a more theoretical level, a probabilistic analysis is presented in [18] whose conclusion is that dynamic ridesharing is viable in dense enough geographical areas (i.e., cities, congested freeway corridors). In contrast, it is severely limited in sparse areas or very small communities.

On the other hand, other authors have studied the issues related to the start-up phase of a dynamic ride-sharing system, in which enough participants (from those available in the geographical area) should be attracted by the system in order to gather the critical mass that enables good matches on short notice. According to the Bass diffusion model of new products [5], the probability that a peer participates into a system is a linear function of the number of previous participants. In turn, the probability of initial adoption closely depends on the innovation of the system with respect to other similar products available in the market [2]. Even though this can only be tested in a real market, we believe that our system incorporates enough differential features (decentralization, zero running costs, privacy protection, scalability, seamless trust/reputation management) to make it innovative w.r.t. available systems and, thus, potentially capable of gathering a large community of users in dense enough areas.

Figure 6 shows the effectiveness of ride matching when considering only the trips in the dataset that are longer than 10 kms. It should be noted that, due to the length constraint, the number of traces we consider is this experiment for passengers is 5,394 (rather than 28,221), 8,961 (rather than 47,035) and 12,496 (rather than 65,849) for the three respective scenarios. Due to the stronger spatial constraints imposed by the longer trips needed by the passen-

Table 6: *Passenger-dominated scenario*: percentage of offers received by passengers that were unmatchable against their spatiotemporal requirements, for varying generalization sizes of locations and times

Passengers' flexibility (time/location)	No generalization	time=300s space=100m	time=600s space=200m	time=1,800s space=1,000m
$300m/250s$	0%	52.11%	73.29%	95.02%
$600m/500s$	0%	36.19%	51.01%	86.91%
$900m/750s$	0%	22.31%	37.86%	78.22%
$1200m/1000s$	0%	10.99%	31.50%	70.62%

gers (which render many offers from drivers useless), the percentage of matches moderately decreases from 14.8% to 6.04% (for the least flexible passengers) and from 82.86% to 60.09% (for the most flexible passengers). These results suggest that, even though our system focuses on dynamic ridesharing, nothing prevents using it to manage longer trips. At most, the lack of some ridesharing types that would be more adequate for longer trips (such as *detour ridesharing*, in which the driver may detour to accommodate the needs of the passengers, or *partial ridesharing*, in which the passenger concatenates several rides with different drivers) may limit the effectiveness of ride matching.

The third experiment evaluates how the granularity of the generalizations influences the effectiveness of ride matching. The size of the generalizations does not affect the actual number of matches because, if a match between a driver and a passenger is possible, their generalized zones and time intervals will overlap and, thus, the match will be found. However, if generalizations are coarse, the chance that the passenger receives driver offers that do not match his spatiotemporal requirements will increase. In Tables 6, 7 and 8, we report the percentage of driver offers received by passengers that were unmatchable against the passengers' spatiotemporal requirements, with varying generalization sizes for locations and times. As expected, the percentage of unmatchable offers increases as we make generalizations coarser, and decreases when the spatiotemporal requirements of the passengers become less strict. Specifically, *when generalizations are coarser than the flexibility of passengers*, the percentage of unmatchable offers surpasses the one of matchable offers (roughly between 70% and 95% unmatchable offers in the three scenarios of the tables); this obviously creates a significant overhead in the system. To prevent this, the system designer should configure the granularity of the generalizations so that the disclosure protection and the efficiency of the matching process are properly balanced.

6. Conclusions and future work

Coming up with mechanisms that contribute to the adoption of ridesharing is of great interest, both for end users and for the society at large. Indeed, ridesharing has the potential to bring significant benefits to the involved agents and a more sustainable management of transportation to society.

Table 7: *Balanced scenario*: percentage of offers received by passengers that were unmatchable against their spatiotemporal requirements, for varying generalization sizes of locations and times

Passengers' flexibility (time/location)	No generalization	time=300s space=100m	time=600s space=200m	time=1,800s space=1,000m
$300m/250s$	0%	55.21%	71.27%	94.27%
$600m/500s$	0%	32.70%	47.55%	85.60%
$900m/750s$	0%	14.51%	33.58%	76.43%
$1,200m/1,000s$	0%	3.87%	26.35%	68.40%

Table 8: *Driver-dominated scenario*: percentage of offers received by passengers that were unmatchable against their spatiotemporal requirements, for varying generalization sizes of locations and times

Passengers' flexibility (time/location)	No generalization	time=300s space=100m	time=600s space=200m	time=1,800s space=1,000m
$300m/250s$	0%	56.06%	69.86%	93.89%
$600m/500s$	0%	31.14%	45.93%	85.04%
$900m/750s$	0%	12.17%	32.00%	75.78%
$1,200m/1,000s$	0%	1.47%	24.48%	67.72%

In this paper, we have tackled two of the main obstacles discouraging ridesharing (namely, the lack of trust and privacy concerns with respect to matching agencies) by means of a reputation-enabled privacy-preserving decentralized P2P ridesharing network. Even though decentralization entails additional challenges in comparison with the usual solutions based on central matching agencies, our ride-matching and reputation management protocols have been carefully designed so that peers are rationally motivated to adhere to them. In addition to being self-enforcing, our protocols are also *co-utile*, because they bring a number of mutual benefits to the involved peers (reduced travel costs, fair reputation calculation, better ride matches, etc.).

The reputation management protocol is a clear example of how intrinsically co-utile incentive protocols can spark co-utility in other protocols that are not intrinsically co-utile: the reputation mechanism is self-enforcing and it enables co-utility in the ride-matching protocol, that would not be co-utile without reputation, due to the lack of trust. Moreover, as shown by our simulations, reputation can also enhance the benefits of collaboration (that is, a higher reputation brings more and better matches). This makes reputation an additional utility that agents wish to maximize, which motivates them to behave well and results in smoother collaboration and more social welfare.

Future work will include deploying a field trial of the proposed mechanism in practice in a metropolitan scenario with insufficient public transportation. We are currently developing apps to be used by passengers and drivers on their smartphones. Trying the system with real users will allow: adjusting the score grading for local reputations and the reputation thresholds users view as desirable; deciding whether more complex ridesharing types are needed (flexible drivers, partial ridesharing, detour ridesharing, etc.); refining the performance metrics, etc. Another non-technical but very relevant piece of future work is finding a suitable business model for a decentralized ridesharing system such as the one we have designed. Whereas in a centralized system the central matching agency can make profit out of the matching, in a decentralized system the business model should rely on less direct sources of revenue, such as advertising, app sale and/or maintenance, etc.

Beyond the ridesharing application, co-utility has the potential to improve other P2P interactions in the so-called "sharing-economy", by which peers share the access to goods and services through community-based online services. A promising research avenue is to apply co-utility to other scenarios framed both in the digital world (like file sharing) and in the physical world (like crowdsourcing).

Acknowledgments and disclaimer

The following funding sources are gratefully acknowledged: Templeton World Charity Foundation (grant TWCF0095/AB60 "CO-UTILITY"), Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2014 SGR 537), Spanish Government (projects TIN2014-57364-C2-1-R "SmartGlacis", TIN2015-70054-REDC and TIN2016-80250-R "Sec-MCloud"), European Commission (projects H2020-644024 "CLARUS" and H2020-700540 "CANVAS"). The authors are with the UNESCO Chair in Data Privacy, but the views in this paper are the authors' own and are not necessarily shared by UNESCO or any of the funding sources.

References

[1] AASHTO. Commuting in America 2013: The National Report on Commuting Patterns and Trends, 2013. Available at `http://traveltrends.transportation.org/Pages/default.aspx` Last accessed: Sep. 14, 2016.

[2] N.A.H. Agatz, A.L. Erera, M.W.P. Savelsbergh, X. Wang. Dynamic ridesharing: a simulation study in Metro Atlanta. In: Proceedings of the 19th International Symposium on Transportation and Traffic Theory, 2011, p. 532-550.

[3] N.A.H. Agatz, A.L. Erera, M.W.P. Savelsbergh, X. Wang. Optimization for dynamic ridesharing: a review. European Journal of Operational Research 223(2) (2012) 295-303.

[4] A.M. Amey. A proposed methodology for estimating ridesharing viability within an organization, application to the MIT community. In: Transportation Research Board Annual Meeting, 2011.

[5] F. Bass. A new product growth model for consumer durables. Management Science 15(5) (1969) 215-227.

[6] R. Buliung, K. Soltys, R. Bui, C. Habel, R. Lanyon. Catching a ride on the information super-highway: toward an understanding of internet-based carpool formation and use. Transportation 37(6) (2010) 849-873.

[7] M. Castro, P. Druschel, Y. C. Hu, A. Rowstron. Exploiting network proximity in peer-to-peer overlay networks. Technical report MSR-TR-2002-82, (2002).

[8] B. Caulfield. Estimating the environmental benefits of ridesharing: A case study of Dublin. Transportation Research Part D 14 (2009) 527-531.

[9] N.D. Chan, S.A. Shaheen. Ridesharing in North America: past, present and future. Transport Reviews 32(1) (2012) 93-112.

[10] V. Chaube, A.L. Kavanaugh, M.A. Pérez-Quiones. Leveraging social networks to embed trust in rideshare programs. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS), 2010, pp. 1-8.

[11] C. Cottrill. Approaches to privacy preservation in intelligent transportation systems and vehicle-infraestructure initiative. Transportation Research Record: Journal of the Transportation Research Board 2129 (2009) 9-15.

[12] D.J. Dailey, D. Loseff, D. Meyers. Seattle smart traveler: dynamic ridematching on the world wide web. Transportation Research Part C 7(1) (1999) 17-32.

[13] S.E. Deering, D.R. Cheriton. Multicast routing in datagram internetworks and extended LANs. ACM Trans. Computer Systems 8(2) (1990) 85-110.

[14] J. Domingo-Ferrer, O. Farràs, S. Martínez, D. Sánchez, J. Soria-Comas. Self-enforcing protocols via co-utile reputation management. Information Sciences 367-368 (2016) 159-175.

[15] J. Domingo-Ferrer, D. Sánchez, J. Soria-Comas. Co-utility: self-enforcing collaborative protocols with mutual help. Progress in Artificial Intelligence 5(2) (2016) 105-110.

[16] R.N. Fries, M.R. Gahrooei, M. Chowdhury, A.J. Conway. Meeting privacy challenges while advancing intelligent transportation systems. Transportation Research Part C 25 (2012) 34-45.

[17] F. Furuhata, M. Dessouky, F. Ordóez, M.E. Brunet, X. Wang, S. Koening. Ridesharing: the state-of-the-art and future directions. Transportation Research Part B 57 (2013) 28-46.

[18] R.W. Hall, A. Qureshi. Dynamic ride-sharing: theory and practice. Journal of Transportation Engineering 123(4) (1997) 308-315.

[19] K. Hoffman, D. Zage, C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. ACM Computing Surveys 42(1) (2009) art. no. 1.

[20] INRIX. Economic & Environmental Impact of Traffic Congestion in Europe & the US. INRIX Cost of Congestion Report, 2014. Available at *http://inrix.com/economic-environment-cost-congestion/* Last accessed: Sep. 14, 2016.

[21] R. Jurca, B. Faltings. An incentive compatible reputation mechanism. In: Proceedings of the IEEE International Conference on E-Commerce (CEC), 2003, pp. 285-292.

[22] S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web, ACM, 2003, pp. 640-651.

[23] K. Leyton-Brown, Y. Shoham. Essentials of Game Theory: A Concise, Multidisciplinary Introduction. Morgan & Claypool, 2008.

[24] C. Morency. The ambivalence of ridesharing. Transportation 34(2) (2007) 239-253.

[25] Office of Highway Policy Information. Annual Vehicle Distance Traveled in Miles and Related Data. U.S. Department of transportation, Federal Highway Administration, 2013. Available at *https://www.fhwa.dot.gov/policyinformation/statistics/2013/vm1.cfm* Last accessed: Feb. 26, 2016.

[26] Office of Transportation and Air Quality. Measuring Greenhouse Gas Emissions from Transportation. United States Environmental Protection Agency, 2016. Available at *http://www3.epa.gov/otaq/climate/measuring.htm* Last accessed: Feb. 26, 2016.

[27] M. Piorkowski, N. Sarafijanovoc-Djukic, M. Grossglauser. A parsimonious model of mobile partitioned networks with clustering. In: Proceedings of the International Conference on COMmunication Systems and NETworkS (COMSNETS), 2009.

[28] P. Resnick, R. Zeckhauser. Trust among strangers in internet transactions: empirical analysis of eBay's reputation system. In: Baye, M.R. (Ed.), The Economics of the Internet and E-Commerce, vol. 11, pp. 127-157.

[29] P. Samarati. Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering 13(6) (2001) 1010-1027.

[30] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '01), ACM, 2001, 149-160.

[31] L. Sweeney. k-Anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5) (2002) 557-570.

[32] U.S. Federal Trade Commission. Data Brokers, A Call for Transparency and Accountability. May, 2014.