Credit-based Peer-to-Peer Ride Sharing using Smart Contracts

Somay Chopra
Dept. of Computer Sc. & Engineering
Indian Institute of Technology Kharagpur
thisissomaychopra@iitkgp.ac.in

Balaji Palanisamy
School of Computing and Information
University of Pittsburgh
bpalan@pitt.edu

Shamik Sural
Dept. of Computer Sc. & Engineering
Indian Institute of Technology Kharagpur
shamik@cse.iitkgp.ac.in

***Abstract*—Existing ride sharing services are monetary-based and are managed by centralized service providers. In this paper, we propose a decentralized non-monetary ride-sharing platform in which users interact directly with each other. Fairness is ensured through the use of credits so that a user not only enjoys rides but also offers to drive from time to time. The application is developed on the Ethereum blockchain and is designed to provide transparency and verifiability in its operations.**

I. Introduction

App-cab services like Uber and Lyft allow customers to share rides with other people and benefits the individuals by saving cost and reducing the number of cars on the road. Existing services are monetary-based and are centralized and the pricing and operations are managed by the service provider. A potential solution to enhancing the transparency and decentralization of operations is to use a blockchain so that users can interact and participate directly in a peer-to-peer manner. Indeed, a number of models for blockchain based car sharing systems have been proposed in the recent past [10] [11] [8] [3] [9] [6] [7]. However, these solutions involve hard currency or cryptocurrency payments and hence, they are simply decentralized versions of the existing commercial services. In contrast, we propose a non-monetary peer-to-peer ride sharing application on Ethereum where users cooperate with each other for sharing a ride. The incentive for a user is not to earn money but to leverage and appreciate the societal benefits of ride sharing [5].

II. System Design

In this section, we first describe the proposed approach for credit based ride sharing and present an illustrative example for elucidation. Figure 1 gives a high level overview of the various components of the service.

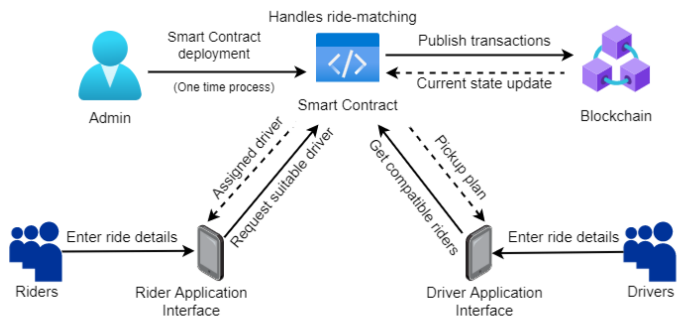

Fig. 1: Ride Sharing System Overview

The system accepts requests from users willing to ride and those offering to drive. As and when these are available, our blockchain based ride sharing application attempts to match the driver with the riders having compatible routes. A ride matching algorithm deployed as a smart contract on Ethereum takes the drivers and riders at a given point of time as input along with other information including the pick up and drop-off locations and number of seats in the vehicle. For every driver, it outputs a plan from the starting point to the destination and the locations of the riders to be picked up on the way. This is done in such a way that the primary route of the driver is not affected significantly, while ensuring to pick as many riders as possible. Instead of being currency driven, the application makes use of ride credits for enforcing fairness. Each user starts with an initial number of credits. As the user offers to drive, additional credits are earned, which are deducted when the user enjoys a ride. Use of blockchain guarantees transparency of the entire process.

978-1-6654-9538-7/22/$31.00 ©2022 IEEE

The ride matching algorithm runs for all the drivers who are currently active in the system and not assigned any plan yet. The first step for any given driver is to determine the shortest path from the starting point to the destination, treating it as the user's original path plan. The extra distance that a driver is required to travel while picking up the riders should not exceed the original path beyond a certain percentage. The algorithm next looks up for a potential rider to be picked up within a neighborhood of the current location (initialized to the originating point of the driver). This lookup distance is equal to the difference between the distance of the new path plan and the shortest distance with no riders to pick. The lookup distance is updated every time a rider is added to the pick up plan. A rider can be added to the existing plan only if the updated path after picking and dropping the rider satisfies the aforementioned constraints. If such a rider is found, then the path is updated according to the new plan and the current node is moved to the pick up node of the newly added rider. Otherwise, the current node is updated to the next node in the existing pick up plan. This process is repeated until the current

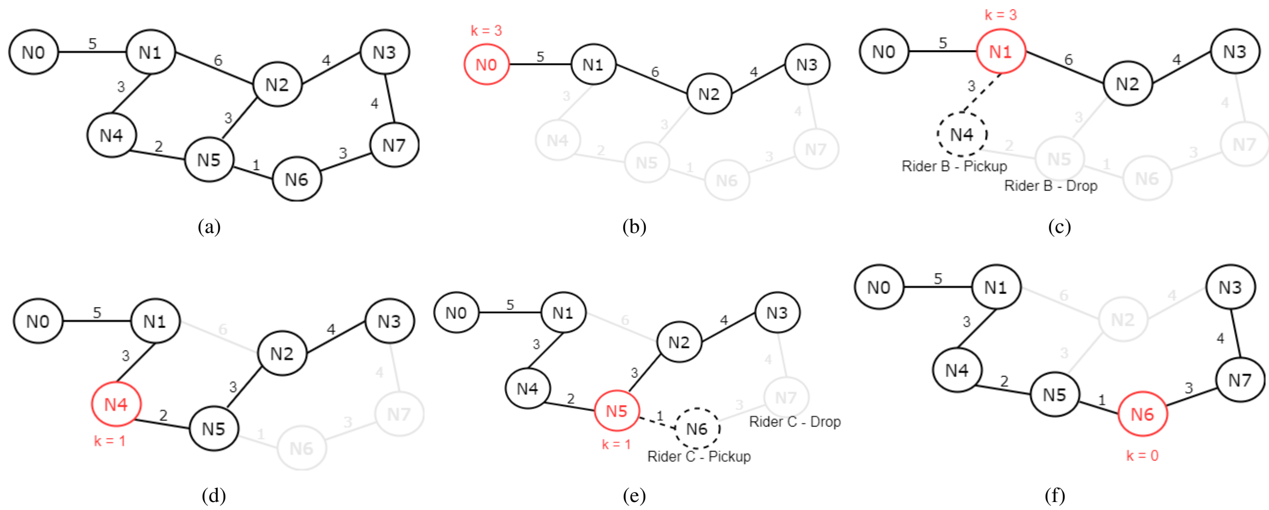

Fig. 2: (a) Sample Graph (b) Initial Path with no Eligible Rider (c) At Node 1, Eligible Rider at Node 4 (d) Path updated to pick Rider at Node 4 (e) At Node 5, Eligible Rider at Node 6 (f) Final Path picking up Riders B, C

User	Role	Start	End	Credits	Seats
A	Driver	Node 0	Node 3	1 (3)	2
B	Rider	Node 4	Node 5	4 (3)	-
C	Rider	Node 6	Node 7	5 (4)	-

TABLE I: Sample Users for the Illustrative Example

node reaches the destination of the driver, or if there are no more vacant seats available in the driver's vehicle.

We use the illustrative example in Fig. 2 to demonstrate the working of the above steps and the output obtained from the smart contract. Fig. 2(a) shows a sample graph with the nodes (representing road intersections) marked, and their pairwise distance labeled on the edges. Table I shows three users A, B and C for this example. We consider three scenarios: **Case 1:** Only User A is in the system at the time of executing the algorithm. **Case 2:** Users A and B are in the system. **Case 3:** All the three users are in the system.

In Case 1, as there are no riders to pick (Fig. 1(b), with current node marked in red and current path plan highlighted), the algorithm determines the shortest path possible for the driver, although the driver is free to choose any path in this case since there are no commitments to any rider.

In Case 2, the driver starts from Node 0 with original path as in Case 1 with a path length of 15 units. Considering the additional distance threshold of 20%, the maximum distance that the driver can travel after a possible re-route is 18 units. The value of k (the radius for potential rider lookup) is 3. At Node 0 (Fig. 2(b)), there is no rider within k distance and the algorithm follows the initial path and proceeds to Node 1. At this node (Fig. 2(c)), Rider B is at a distance of 3, which is within k. If this rider is picked, the new path will be through Nodes 4, 5 and 2, with a distance of 17 units, which is less than the maximum allowed distance, making Rider B eligible to be picked up. The plan for the driver is hence updated to this new path (Fig. 2(d)).

In Case 3, the algorithm works as in Case 2 till Node 5. At Node 5 (Fig. 2(e)), Rider C is eligible to be picked up and allows a possible re-route, hence it changes the path of the driver through Nodes 6 and 7 (Fig. 2(f)). The updated credits are shown in the table within parenthesis assuming an addition of one credit for each rider picked up and deduction of one credit for each ride enjoyed.

III. Prototype Implementation

The ride-matching algorithm has been implemented as an Ethereum smart contract written in Solidity [2]. The Remix IDE was used for compiling and testing. Deployment and execution of transactions were done on the Göerli test network [1], which provides the complete functionality of the main Ethereum network and also issues some free Ethers to work with. The smart contract deployed on the Göerli test network stores the current state of the system and contains the functions to be called for ride-matching every time a driver enters the system. Note that deployment of the smart contract in Ethereum and the transactions for ride matching would cost some gas to be subsequently paid in Ethers. The initial smart contract deployment is an one time process and can be done by the application service admin. Subsequently, users can be billed nominally in hard currency for the usage gas costs depending on the level of their individual interactions with the platform.

IV. Future Directions

We plan to further optimize the ride-matching algorithm by reducing the time and memory complexity to reduce the gas cost [4]. We plan to evaluate our platform using a real world map and we aim to implement the proposed approach as a full-fledged DApp which will include a front-end that will use a location API to get the physical locations of the users and dynamically execute the ride-matching algorithm.

REFERENCES

[1] Göerli: Ethereum test network. https://goerli.etherscan.io/.

[2] Solidity - solidity 0.8.0 documentation. https://docs.soliditylang.org/en/v0.8.10/.

[3] Sid Chau, Shuning Shen, and Yue Zhou. Decentralized ride-sharing and vehicle-pooling based on fair cost-sharing mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–11, 11 2020.

[4] Muhammad Milhan Afzal Khan, Hafiz Muhammad Azeem Sarwar, and Muhammad Awais. Gas consumption analysis of Ethereum blockchain transactions. *Concurrency and Computation: Practice and Experience*, page e6679, 2021.

[5] Marco Migliore, Gabriele D'Orso, and Domenico Caminiti. The environmental benefits of carsharing: The case study of Palermo. *Transportation Research Procedia*, 48:2127–2139, 01 2020.

[6] Panchalika Pal and Sushmita Ruj. BlockV: A blockchain enabled peer-peer ride sharing service. In *2019 IEEE International Conference on Blockchain*, pages 463–468, 2019.

[7] Sathya A. Renu and Barnali Gupta Banik. Implementation of a secure ride-sharing DApp using smart contracts on ethereum blockchain. *International Journal of Safety and Security Engineering*, 11:167–173, 04 2021.

[8] Nishant Saurabh, Carlos Rubia, Anandakumar Palanisamy, Spiros Koulouzis, Mirsat Sefidanoski, Antorweep Chakravorty, Zhiming Zhao, Aleksandar Karadimce, and Radu Prodan. The articonf approach to decentralized car-sharing. *Blockchain: Research and Applications*, page 100013, 05 2021.

[9] David Sánchez, Sergio Martínez, and Josep Domingo-Ferrer. Co-utile P2P ridesharing via decentralization and reputation management. *Transportation Research Part C: Emerging Technologies*, 73:147–166, 2016.

[10] Somanath Tripathy, Mayank Aggarwal, and Sandip Chakraborty. Beyond Uber and Lyft: A decentralized cab consortium over blockchains. In *Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, BSCI '21, page 97–102, New York, NY, USA, 2021. Association for Computing Machinery.

[11] Edgar Vazquez. and Dario Landa-Silva. Towards blockchain-based ride-sharing systems. In *Proceedings of the 10th International Conference on Operations Research and Enterprise Systems - ICORES,*, pages 446–452. INSTICC, SciTePress, 2021.