A city-scale IoT-enabled ridesharing platform

Claudio Gambella(B), Julien Monteil(B), Anton Dekusar, Sergio Cabrero Barros(B), Andrea Simonetto(B) and Yassine Lassoued

Ireland Lab, IBM Research, Ireland

ABSTRACT
The advent of on-demand mobility systems is expected to have a tremendous potential on the wellness of transportation users in cities. Yet such positive effects are reached when the systems under consideration enable seamless integration between data sources that involve a high number of transportation actors. In this paper we report on the effort of designing and deploying an integrated system, including algorithms and platforms, that can operate in cities, in an Internet of Things (IoT)-aware fashion. The system was evaluated by enabling/disabling the IoT components of the system, highlighting the necessity of real-time data integration for efficient mobility services.

Introduction

Ridesharing consists of optimizing the pickup and drop-off locations and routes of potentially-shared rides, using a fleet of vehicles (Furuhata et al. 2013; Mourad, Puchinger, and Chu 2019). Ridesharing is typically solved in a centralized fashion across the whole vehicle fleet and across customers, as to minimize the total travel time of the operating vehicles, while taking into account the customer preferences (pickup and drop-off times, whether sharing is desired or not, etc.) (Simonetto, Monteil, and Gambella 2019; Pandey et al. 2019; Alonso-Mora et al. 2017). Ridesharing emerged in the past few decades amongst the shared-mobility services as a means to limit traffic congestion and achieve environmental benefits (Yin et al. 2018; Lokhandwala and Cai 2018). In such a context, external information about the traffic situation and events affecting travel times, collected by IoT devices, may be taken into account in the scheduling of ridesharing. IoT is used to optimize the allocation, routes, pickup and drop-off locations based on real-time information received from the IoT ecosystem (e.g. traffic jams, incidents, road closures, traffic lights, etc.) (Coppola and Morisio 2016; Verma et al. 2017). In the current literature on decision support systems for ridesharing, on the one hand, it is known that an unregulated adoption of a fleet of ridesharing vehicles can have a negative impact on the traffic conditions (Li, Hong, and Zhang 2002; Stiglic et al. 2015), but, on the other hand, the external modifications of traffic conditions are typically neglected in the optimization process. While the effects of traffic conditions have been studied in small-sized ridesharing systems (Wang, Dessouky, and Ordonez 2016; Xu et al. 2015), the inherent complexity of managing city-scale systems hindered their analysis on traffic-aware scenarios (David Fiedler and Ćertický 2017). The complex relationship between ridesharing and traffic congestion is also studied in Xu, Ordoñez, and Dessouky (2015); however, the assignment model is static and tested on the Sioux–Falls instance ((Sioux Falls network)) with 24 geographical nodes. In this paper, we propose a city-scale IoT implemented architecture and system to augment the optimization-based ridesharing service proposed in (Simonetto, Monteil, and Gambella 2019). The contribution of the present paper is to leverage the increasing availability of IoT devices, and propose a dynamic real-time decision support system for ridesharing at city-scale, which integrates an optimization module with information received by the IoT infrastructure. In particular, the impact of the traffic congestion detected by IoT devices into ridesharing optimization is quantitatively analyzed on city-scale simulations. Such IoT-enabled ridesharing service has been developed as part of the Brainport pilot site of the European Horizon 2020 AUTOPILOT project.

An IoT-based approach for ridesharing

Architecture

Figure 1 shows the architecture of the IoT-based ridesharing service that we implemented. As can be seen in this architectural diagram, a customer interacts with the service through a mobile app. Customer requests are communicated to the ride-sharing service through a message broker and are processed in batches collected over regular time intervals. Shared cars communicate with the ridesharing service through a dedicated vehicle IoT platform. They act as ‘things’. Cars submit their events (locations, customer pickup and drop-off confirmations, detected events) to the vehicle IoT platform. They receive customer assignment and route instructions from the ridesharing scheduler through the same IoT platform. The ridesharing service queries a routing engine for the travel times between pairs of nodes of the area of interest. The routing engine receives notifications about changes in traffic speeds and situations through the oneM2M IoT platform, and updates its maps accordingly. In this way, traffic information is taken into account in the determination of travel times, and the assignment provided by the ridesharing scheduler takes this into account.

A standard oneM2M-based context IoT platform is used to access traffic events communicated by external devices, road sensors, and vehicles. The vehicle IoT platform exchanges data with the context IoT platform through a oneM2M interworking gateway. Data are exchanged as follows.

- In one direction, relevant traffic events affecting the operations of the shared car fleet are communicated from the

CONTACT Claudio Gambella claudio.gambella1@ie.ibm.com Ireland Lab, IBM Research, Ireland

© 2019 Informa UK Limited, trading as Taylor & Francis Group
oneM2M IoT platform to the vehicle IoT platform through the oneM2M interworking gateway. These may then be received by the shared cars to adjust their routes.

- In the other direction, any events detected by the shared cars (e.g., accidents, hazards, etc.) are communicated from the vehicle IoT platform to the oneM2M IoT platform through the oneM2M interworking gateway.

### Ridesharing optimization

The aim of the ridesharing service is to find an assignment of available vehicles to customers so as to optimize a performance indicator, such as vehicle travel times (Agatz et al. 2012; Ho et al. 2018). A vehicle \( v \in V \) is considered a candidate to serve a customer \( c \in C \) at a given time \( t \) if it has a residual seat capacity, and it is located at an acceptable distance from \( c \). Each customer specifies his trip request as an origin-destination coordinate pair and, optionally, time preferences for pickup and drop-off (e.g., maximum waiting time). As observed by (Simonetto, Monteil, and Gambella 2019), ridesharing optimization can be modeled as a linear assignment problem (Bertsekas 1990). Let \( x_{ij} \in \{0, 1\}, i \in V, j \in C \) be the set of binary decision variables: \( x_{ij} = 1 \) only if the request of customer \( j \in C \) is assigned to vehicle route \( i \in V \) at cost \( c_{ij} \). The optimization module of the ridesharing service seeks to assign each vehicle to exactly one new customer by solving the following linear assignment problem

\[
\min \sum_{i \in V, j \in C} c_{ij}x_{ij} \tag{1a}
\]

subject to:

\[
\sum_{i \in V} x_{ij} = 1, \quad j \in C \tag{1b}
\]

\[
\sum_{j \in C} x_{ij} = 1, \quad i \in V \tag{1c}
\]

\[
x_{ij} \in \{0, 1\}, \quad i \in V, j \in M \tag{1d}
\]

At time \( t_k \), the ridesharing scheduler works as follows:

- It obtains from a message broker the customer requests submitted during the time interval \([t_{k-1}, t_k]\).
- It asks insertion costs \( c_{ij} \) to the vehicles indicated by the context mapping. This involves to solve a single-vehicle Dial-a-Ride problem (see, e.g., (Cordeau and Laporte 2003)), via a greedy insertion heuristic.
- It queries an optimization module for an optimal assignment of vehicles to requests, obtained by solving Problem (1)-(4).
- If some customers cannot be served, it calls an internal rebalancing module, which runs the logic again (steps 2 to 3) with looser customer time constraints and for idle vehicles only.
- It sends the assignments and their corresponding routes to the customers and vehicles.

For a detailed discussion on the ridesharing scheduler, the reader is referred to (Simonetto, Monteil, and Gambella 2019).

### Evaluation

We validated our framework in two ways. Firstly, we ran large-scale (city-wide) simulations with virtual vehicles and customers. This allowed us to evaluate the benefits of the IoT-based approach performance in terms of quality of service. Secondly, we demonstrated the ridesharing service using two real-automated vehicles. This allowed us to verify the feasibility of the approach.

### Simulations

In this validation approach, we ran simulations in six regions in the Brainport area, Netherlands, with virtual fleets of cars and virtual
customers. Here, we report on the Brainport simulations. As illustrated in the screenshot of Figure 2, the simulation area covers Eindhoven and Helmond and the highway that links both towns. The area contains 82629 Open Street Map nodes. From these simulations, we draw conclusions relative to the scalability of our proposal and the advantages of integrating it with traffic data generated by IoT devices.

In the Brainport simulations, we generate a fleet of 100 virtual vehicles serving between 360 and 1440 user requests during 1 h of service. Vehicles are assigned to requests every 10 s; (Simonetto, Monteil, and Gambella 2019) showed that such frequency for batch assignment provides an optimum trade-off between level of service and computation time. We have several speed cameras connected to the oneM2M IoT platform which provide traffic events every 5 min. These cameras are placed on a stretch of the highway between Eindhoven and Helmond. We recorded traffic events from the actual speed cameras for an hour and then we played them back during simulations. To make simulations more realistic we also simulated more speed cameras on the circular road around Eindhoven. Having gathered this IoT data we aim to evaluate its effect on the ridesharing system.

Each vehicle has the capacity to carry up to four passengers at the same time, and its location is initialized randomly in one of the six areas of interest of Eindhoven, Larger Eindhoven, Helmond, Neunen, Geldrop, Whole area (gray box in Figure 2), as reported in Table 1. Users are simulated by generating ride requests, each consisting of the origin and destination coordinates for a single passenger. Passengers expect their requests to be attended within 7 min. In case a request is not met, a rebalancing module attempts to serve it by assigning an idle vehicle as mentioned in the logic description in Section 2.2. Similar to the initial vehicle locations, the origin of each request is calculated using six predefined areas with the probability indicated in Table 1. Once the origin is selected, the destination is selected following the conditional probability documented in Table 1. For example, if the origin is a random point in Larger Eindhoven, there is a 0.05 probability of selecting a random point in Helmond as destination. Ride requests are grouped into batches of 10 s, and the batches are processed by the ridesharing service. Every batch has between 1 and 4 requests. For reproducibility purposes, ride requests are generated only once, offline, and then injected into the simulations at predefined time instants.

Respectively, the columns indicate: the North East and South West coordinates of the bounding box of each area, the probability for the vehicles to have initial location in an area, the probability for the trips to have given origin and destination area.

The experiment setup consists of a custom traffic simulator that updates vehicle positions on a map considering the road network and traffic speeds per road link. The simulator is connected to the ridesharing service through the IoT platforms. The simulator is connected to the vehicle IoT platform to update vehicle routes and positions, and makes use of prerecorded IoT event data sitting on the oneM2M IoT platform. So, from the service point of view, there is no difference between the simulator and the real world. In addition, the simulator can be configured to simulate real-time traffic on selected roads, by updating the OSRM routing contract. We inserted IoT traffic data using this mechanism so as to simulate traffic jams in different roads. We are then able to use the routing

**Figure 2.** Screenshot of the simulation user interface showing the study area in Brainport, the Netherlands. Markers indicate the position of vehicles and the number of passengers they are carrying.

| Destination | Vehicle | Origin | E | L | H | N | G | W |
|-------------|---------|--------|---|---|---|---|---|---|
| E           | 51.4584, 5.5157 | 51.4154, 5.4453 | 0.30 | 0.30 | 0.30 | 0.30 | 0.05 | 0.15 | 0.15 | 0.05 |
| L           | 51.4927, 5.5323 | 51.4106, 5.4310 | 0.20 | 0.20 | 0.20 | 0.20 | 0.10 | 0.10 | 0.10 | 0.10 |
| H           | 51.5016, 5.7155 | 51.4511, 5.6008 | 0.20 | 0.20 | 0.20 | 0.20 | 0.10 | 0.10 | 0.10 | 0.10 |
| N           | 51.4814, 5.5756 | 51.4566, 5.5302 | 0.10 | 0.10 | 0.10 | 0.10 | 0.30 | 0.30 | 0.05 | 0.20 | 0.05 | 0.10 |
| G           | 51.4366, 5.5831 | 51.4068, 5.3342 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| W           | 51.5021, 5.7249 | 51.4018, 5.3950 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
engine in both the 'IoT-enabled' and 'IoT-disabled' modes. In both modes, vehicles move at the appropriate speeds as measured by the IoT devices, and update their locations accordingly. The ridesharing scheduler is only aware of the IoT measurements in the 'IoT-enabled' condition. Thus, in the 'IoT-disabled' condition, assignments are optimized assuming free-flow traffic, i.e. traffic in all roads is fluid. The expectation is that this will lead the ridesharing scheduler to suboptimal recommendations that are detrimental to the service quality.

In the following, we show some results obtained for the area of interest and we compare the performance of the ridesharing service in both 'IoT-enabled' and 'IoT-disabled' scenarios. Figure 3 shows the evolution of the number of customers served over time, which is a common key performance indicator for ridesharing services (see, e.g. (Agatz et al. 2012)). The growth of the served customers is larger for the 'IoT-enabled' service. Making ridesharing aware of IoT data has a positive impact not only on the number of users served, but also on vehicle utilization, and on the number of customers in waiting condition, as shown in Figure 4. By analyzing the average number of customers per vehicle in Figure 5 and the number of customers waiting for a ride in Figure 6, it is possible to observe that the advantage of the 'IoT-enabled' service over the 'IoT-disabled' one is noticeable toward the end of the simulation. This is because neglecting the variations of the travel times due to traffic appears to have a cumulative effect during the

---

**Figure 3.** Number of users served over the one-hour simulation, for the 'IoT-enabled' and 'IoT-disabled' ridesharing service.

**Figure 4.** Customer status at the end of the one-hour simulation for the 'IoT-enabled' and 'IoT-disabled' ridesharing service. The histogram reports the number of served customers, the number of waiting customers, and the number of customers traveling on the cars.
simulation. It is not possible however to establish a strict dominance of the 'IoT-enabled' service at each instant of the simulation for each of these indicators, as shown by the waiting customers' plot in Figure 6. An important indicator for the customer satisfaction reported in Figure 7 is the detour time, computed as the difference from the customer-preferred travel time at trip destination and the actual arrival time. In the 'IoT-disabled' service, fewer customers are affected by delays of more than 6 min.

**MaaS demonstration**

In addition to the above simulation results, we further validated the feasibility of our architecture and approach in a real environment (Brainport) using real connected autonomous vehicles. The demonstration was carried out at the ITS Europe Congress, in Eindhoven, June 2019 (Autopilot at the 13th ITS European Congress 2019). In the demonstration, two automated vehicles were used. A customer wanting to travel from the automotive campus of Helmond to Eindhoven booked a shared ride through our ride-sharing service, which assigned one of the autonomous vehicles to the customer request. The vehicle, which was initially parked, drove itself autonomously from its parking location to the customer. This was operated by an automated valet parking (AVP) system developed by the AUTOPILOT project partners. The vehicle then picked up the customer and headed to Eindhoven. The ride-sharing was also demonstrated in a broader context of mobility as a service, where platooning with the second automated vehicle was tested.
Conclusion

The paper proposed an IoT architecture for city-scale ridesharing services. The impact of traffic events affecting the vehicle travel times has been assessed via a simulation in a geographical area in the Netherlands. The results advocate for the need of IoT-aware ridesharing in urban environments, as an approach to improve the number of served customers as well as the utilization of the vehicles, and to limit the inconveniences experienced by customers, such as waiting time and detour time. The service has been successfully demonstrated during the ITS Europe Congress, 2019. Future work will focus on IoT simulations of larger scale.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731993 (Autopilot).

References

Agatz, N., A. Erera, M. Savelsbergh, and X. Wang. 2012. “Optimization for Dynamic Ride-sharing: A Review.” European Journal of Operational Research 223 (2): 295–303. doi:10.1016/j.ejor.2012.05.028.

Alonso-Mora, J., S. Samaranyake, A. Wallar, E. Frazzoli, and D. Rus. 2017. “On-demand High-capacity Ride-sharing via Dynamic Trip-vehicle Assignment.” Proceedings of the National Academy of Sciences 114 (3): 462–467. doi:10.1073/pnas.1611675114.

Autopilot at the 13th ITS European Congress. 2019. https://autopilot-project.eu/iot-for-cad-the-success-of-autopilot-at-the-13th-its-european-congress/

Bertsekas, D. P. 1990. “The Auction Algorithm for Assignment and Other Network Flow Problems: A Tutorial.” Interfaces 20 (4): 133–149. doi:10.1287/inte.20.4.133.

Coppola, R., and M. Morisio. 2016. “Connected Car: Technologies, Issues, Future Trends.” ACM Computing Surveys (CSUR) 49 (3): 46. doi:10.1145/2988524.

Cordeau, J.-F., and G. Laporte. 2003. “The Dial-a-ride Problem (DARP): Variants, Modeling Issues and Algorithms.” Quarterly Journal of the Belgian, French and Italian Operations Research Societies 1 (2): 89–101.

David Fiedler, M. C., and M. Certicky.. 2017. “Impact of Mobility-On-demand on Traffic Congestion: Simulation-based Study.” In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, JAPAN, 1–6. IEEE.

Furuhata, M., M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig. 2013. “Ridesharing: The State-of-the-art and Future Directions.” Transportation Research Part B: Methodological 57: 28–46. doi:10.1016/j.trb.2013.08.012.

Ho, S. C., W. Y. Szeto, Y.-H. Kuo, J. M. Y. Leung, M. Petering, and T. W. H. Tou. 2018. “A Survey of Dial-A-ride Problems: Literature Review and Recent Developments.” Transportation Research Part B: Methodological 111: 395–421. doi:10.1016/j.trb.2018.02.001.

Li, Z., Y. Hong, and Z. Zhang. 2002. “Do Ride-sharing Services Affect Traffic Congestion? an Empirical Study of Uber Entry.” Social Science Research 31: 1–29. 2016.

Lokhandwala, M., and H. Cai. 2018. “Dynamic Ride Sharing Using Traditional Taxis and Shared Autonomous Taxis: A Case Study of NYC.” Transportation Research Part C: Emerging Technologies 97: 45–60. doi:10.1016/j.trc.2018.10.007.

Mourad, A., J. Puchinger, and C. Chu. 2019. “A Survey of Models and Algorithms for Optimizing Shared Mobility.” Transportation Research Part B: Methodological 123: 323–346. doi:10.1016/j.trb.2019.02.003.

Pandey, V., J. Monteil, C. Gambella, and A. Simonetto. 2019. “On the Needs for Maas Platforms to Handle Competition in Ridesharing Mobility.” Transportation Research Part C: Emerging Technologies 108: 269–288. doi:10.1016/j.trc.2019.09.021.

Simonetto, A., J. Monteil, and C. Gambella. 2019. “Real-time City-scale Ridesharing via Linear Assignment Problems.” Transportation Research Part C: Emerging Technologies 101: 208–232. doi:10.1016/j.trc.2019.01.019.
Sioux Falls network. https://github.com/bstabler/TransportationNetworks/tree/master/SiouxFalls
Stiglic, M., N. Agatz, M. Savelsbergh, and M. Gradisar. 2015. “The Benefits of Meeting Points in Ride-sharing Systems.” Transportation Research Part B: Methodological 82: 36–53. doi:10.1016/j.trb.2015.07.025.
Verma, S., Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato. 2017. “A Survey on Network Methodologies for Real-time Analytics of Massive IoT Data and Open Research Issues.” IEEE Communications Surveys & Tutorials 19 (3): 1457–1477. doi:10.1109/COMST.2017.2694469.
Wang, X., M. Dessouky, and F. Ordóñez. 2016. “A Pickup and Delivery Problem for Ridesharing considering Congestion.” Transportation Letters 8 (5): 259–269.
Xu, H., F. Ordóñez, and M. Dessouky. 2015. “A Traffic Assignment Model for A Ridesharing Transportation Market.” Journal of Advanced Transportation 49 (7): 793–816. doi:10.1002/atr.v49.7.
Xu, H., J.-S. Pang, F. Ordóñez, and M. Dessouky. 2015. “Complementarity Models for Traffic Equilibrium with Ridesharing.” Transportation Research Part B: Methodological 81: 161–182. doi:10.1016/j.trb.2015.08.013.
Yin, B., L. Liu, N. Coulombel, and V. Viguié. 2018. “Appraising the Environmental Benefits of Ride-sharing: The Paris Region Case Study.” Journal of Cleaner Production 177: 888–898. doi:10.1016/j.jclepro.2017.12.186.