Smart Parking Systems Design and Integration Into IoT

Charles M. Menne
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
menne019@morris.umn.edu

ABSTRACT

This paper looks at two smart parking reservation algorithms, and examines the ongoing efforts to connect smart systems of different domains in a city's infrastructure. The reservation algorithms are designed to improve the performance of smart parking systems. The first algorithm considers the distance between parking areas and the number of free parking spaces in determining a parking space. The second algorithm uses distance between parking areas and driver destination, parking price, and the number of unoccupied spaces for each parking area. Neither of these smart parking systems cover how they could fit into a larger scale smart system. As a result, we also look at how interconnection of smart systems is a big priority in the European Union. The researchers there aim to avoid vertical silos of information. Sharing smart system information between different domains can be used to create new or improved services.

Keywords

Smart Parking, Internet of Things, Algorithms, Smart City

1. INTRODUCTION

The process of finding an open parking space can be difficult and time consuming, especially in a busy area. Unless you know the area well, finding a space is typically the result of luck. Imagine you have tickets to a Minnesota Vikings game at U.S. Bank Stadium. You're in the downtown Minneapolis area before the game and realize that you need find an open parking space. Where should you begin looking for a space? There's going to be many other drivers in the area competing for the same spaces that you are. This can make finding an open space very difficult and time consuming, potentially even causing you to miss part of the game due to parking. Smart parking can help alleviate some of these issues.

Smart parking systems aim to remove guessing from finding a space. A smart parking system assists drivers in locating and navigating to an open parking space. This reduces time spent looking for a parking space and as a result, the amount of fuel used. In our Vikings game example, smart parking can help you reserve a parking space through the use of an app on your smartphone. You'll then be provided with directions to the parking area that you've reserved a space in. This not only improves your satisfaction as a driver, but also reduces the impact of vehicle pollution on the environment.

This paper looks at two smart parking systems that are a combination of Internet of Things (IoT) based devices and a parking space reservation algorithm. The reservation algorithm is responsible for finding an optimal parking area for the driver. IoT is the connection of everyday devices, such as a toaster or coffee maker, to the Internet. This allows devices to transmit data, such as sensor information, over the internet. With smart parking, this information could be the status of parking space availability in a parking area.

The first system was created by Pham et al. [2], and is described in Section 3. Their goal is to reduce the time it takes to provide a driver with a parking space. This system has been successfully implemented at Feng Chia University in Taiwan. Along with being implemented in a real world setting, simulations have been performed to demonstrate the performance of the systems' reservation algorithm.

The second system was designed by Rehena et al. [3], and is described in Section 4. This system focuses on improving the reservation algorithm. Simulations using the reservation algorithm were used to analyze its performance. This systems' architecture closely relates to the framework used by Pham et al. [2].

After looking at these two systems, we examine the EU's ongoing efforts to connect different types of smart systems to provide new or improved services [1] in Section 5. A large sporting event at a stadium equipped with a smart parking system is used as a proof-of-concept. The example shows how an ambulance is able to respond to an accident in the smart parking system more efficiently. The hospital can access information about the congestion of a parking area and route the ambulance through the least busy areas. To achieve this, researchers have used new communication and data format standards to gather and manage different types of information. These standards are described in Section 5.2.

In addition to describing the three above mentioned systems in Sections 3, 4, and 5, we provide the background in Section 2. We conclude our findings in Section 6.

2. BACKGROUND

In order to understand how the smart parking systems work, a basic understanding of their underlying technologies is required. In this section, background is first given on radio-frequency identification (RFID). This background

UMM CSci Senior Seminar Conference, November 2018 Morris, MN.

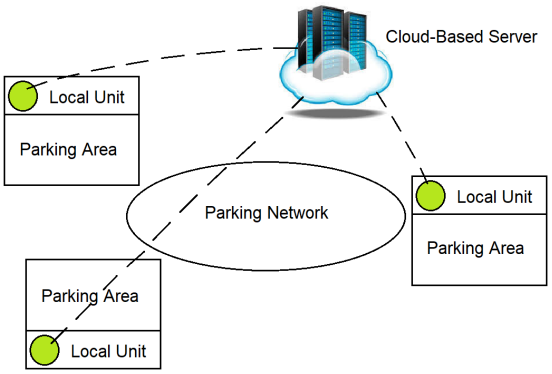


Figure 1: Overview of system architecture. Modified from [2].

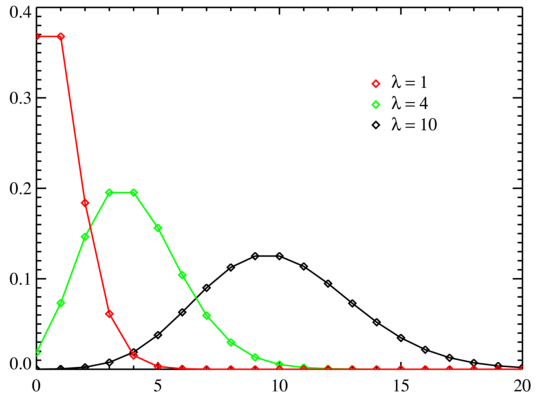


Figure 2: Poisson distribution with varying λ values. X-axis is minutes, y-axis probability. Taken from https://bit.ly/2EotVP6.

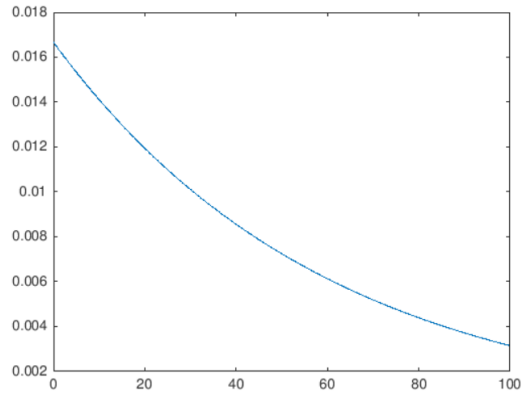


Figure 3: Exponential distribution with $\mu = 60$. X-axis is minutes, y-axis probability. Taken from https://bit.ly/2QoV7np.

information is necessary to understand how a smart parking system monitors its parking areas. A parking area can be thought of as a parking ramp or lot. We also discuss a general parking system architecture. Both of the smart parking systems we look at follow this general architecture. Finally, information about Poisson and exponential distributions is given, which is used in the simulation setup by Pham et al. [2]

2.1 Radio-Frequency Identification

Radio-frequency identification (RFID) provides an object a unique identity by using an RFID tag [5]. This is because each tag has a unique serial number. The tag can be thought of as a bar code on an item you find in stores.

RFID tags by themselves do nothing, much like a bar code on a store item. An RFID reader is needed to identify the tags. The reader sends out a signal that the tags are able to receive. Then, the tags return their identity and other information to the RFID reader.

In a smart parking system, RFID tags can be attached to vehicles for easy authentication when the driver arrives at a parking area equipped with a RFID reader. This is useful in monitoring the number of vehicles currently in that parking area.

2.2 Parking System Architecture

Figure 1 gives an overview of how a general smart parking system architecture could look [2, 3]. There is a parking network that contains all of the parking areas registered to the system. Each of these parking areas is equipped with a Local Unit. These Local Units are responsible for collecting information about their parking area. Typically, this information is the number of vehicles currently parked in that area. However, Local Units in different smart parking systems can vary in the information they collect and operations they perform. After collecting the parking area information, the Local Units send it to a cloud-based server where each parking areas information is stored. Aggregation of parking area information is beneficial for both the users and the system reservation algorithm. Users aren't required to look up information for each individual parking area, as it is all stored in a central location. The reservation algorithm requires the information from each parking area in order to guide users to an appropriate parking area.

2.3 Distributions

Poisson and exponential distributions can model arrival and service times [4]. The arrival time is the time between two drivers entering the system. This follows a Poisson distribution, which uses an average arrival rate denoted λ. Figure 2 shows this distribution with varying λ values. In the smart parking simulation, the x axis represents minutes and the y axis the probability of that x value being chosen. The service time is how much time a driver spends parked. This follows a exponential distribution, which uses an average service rate denoted μ. Figure 3 shows this distribution with a μ of 60. The x and y axis are the same as that in Figure 2.

3. SINGLE-CRITERIA SYSTEM

Before examining the reservation algorithm created by Pham et al. [2], we present the technologies that the system is comprised of in Sections 3.1 and 3.2. The reservation algorithm is presented in Section 3.3. The simulation results for this algorithm are shown in Section 3.4.

3.1 System Architecture

The Local Unit for this system consists of a Control Unit and screen. The Control Unit is an Internet connected Ar-

duino module that is responsible for authenticating drivers, opening the parking area gate after successful authentication, and updating the cloud-based server with the status of parking spaces. The Arduino module has an RFID reader connected for authentication of drivers. Drivers can use an RFID tag or ID card to verify with the system. The screen displays useful information about the parking area for the driver and the status of their authentication. A small map showing the parking area is also shown on the screen.

In our Vikings example, you reserved a parking space with your mobile phone. You were provided with a parking area and directions on how to get there. On arrival at the parking area, you must have an RFID tag or ID card to be verified. The control unit verifies that you have reserved a space. If you have, the screen will indicate that you've been authenticated and the gate will open. After this, the control unit sends the updated parking area information to the cloud-based server.

3.2 Network Architecture

The connection of parking areas in [2] is referred to as a "car park network (CPN)". The CPN is made up of routers with a gateway/bridge. The Local units can send their parking area information to the routers, which then communicate through the gateway/bridge to update the cloud-based server via the internet. A parking area in this architecture is considered to be a node in the CPN.

3.3 Reservation Algorithm

In this section we look at how the smart parking system determines the appropriate parking area for a driver. This is based on a cost value between two nodes. This is calculated by the cost function described in Section 3.3.1. We also look at a mathematical model for the total cost of all vehicles in the system, which is discussed in Section 3.3.2. With this, the researchers can minimize the total cost of the system.

3.3.1 Cost Function

An important part of this algorithm is the function for determining the cost between two nodes. Cost is used in forwarding a driver to a new node if the one they arrived at is full [2]. If the node is full, the driver will be forwarded to the next node with the least cost. The function is called $F_{ij}(\alpha, \beta)$. α and β are both coefficients in the range from 0 to 1 inclusive. Both α and β sum up to exactly 1. The distance between nodes is weighted by α and the percentage of full spaces by β. The cost between current node i and target node j, can be calculated by:

$$F_{ij} = F_{ij}(\alpha, \beta) = \alpha \cdot \frac{d_{ij}}{D_{up}} + \beta \cdot \frac{t_j}{T_{up}} \quad (1)$$

Here d_{ij} is the distance between nodes i and j. D_{up} is the greatest maximum distance between any two nodes. t_j is the number of full spaces in node j. T_{up} is the greatest maximum capacity of any node. Both D_{up} and T_{up} are constants in the system.

The initial request for a parking space calculates $F_{ij}(\alpha, \beta)$ using the distance between the driver's current location i, and parking area node j. The researchers have used simulations to determine good values for the α and β coefficients. Different values produce better or worse average wait and total times for drivers. The simulations are discussed in Section 3.4.

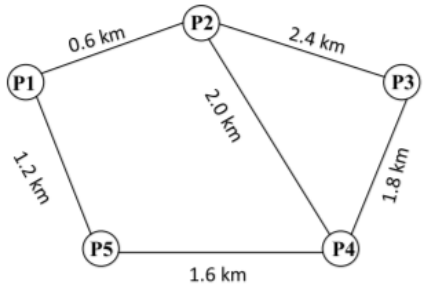


Figure 4: Simulation parking areas setup. Taken from [2].

3.3.2 Total Cost

Now that the cost of each node has been calculated, a model is used for handling parking requests in a parking system. They let P and S be the set of vehicles in the system and the number of parking areas, respectively. C is the total cost of all drivers in P. C can be calculated by:

$$C = \sum_{i=1}^{M} \sum_{j=1}^{N} F_{ij}(\alpha, \beta) \cdot x_{ij} \quad (2)$$

where M and N correspond to the size of P and S. The cost function $F_{ij}(\alpha, \beta)$ is the same equation shown in equation 1. The value of x_{ij} will be 1 if the driver P_i will park at the parking area S_j, otherwise it will be 0. The researchers aim to minimize C while ensuring that a driver only gets one parking space and a parking space is only assigned to a single driver. Minimizing C will help reduce other costs to drivers and the environment.

3.4 Results

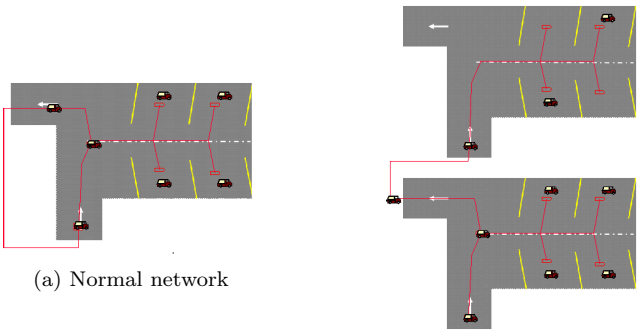
Evaluation of the reservation algorithms performance is done with a simulation tool called Arena [2]. Arena allows researchers to model the processes of a system and export statistical data about the system. The two statistics exported for this smart parking system are the average driver wait time for their parking request and the average total time a vehicle spends in the system. Time in the system refers to the sum of average vehicle waiting, travel, and parking times.

3.4.1 Simulation setup

In the simulation, drivers are randomly created to join the system. This process follows a Poisson distribution, denoted POIS(X). The total time drivers spent parked follows an exponential distribution, denoted EXPO(Y). In their simulations two different X values, 15 and 20 minutes, and a Y value of 60 minutes were used.

The researchers decided to use five parking areas with four spaces each in their simulations. Figure 4 shows how the parking areas were arranged. They simulated with the same number of vehicles arriving at each parking area; the different number of vehicles being 50, 60, 70, 80, 90, and 100.

As mentioned in Section 3.3.1, the researchers are trying to find good values for α and β in the cost function. Certain values are better at reducing the average wait and total time. They simulated all values from 0 to 1 for α. The resulting β is $1 - \alpha$. However, they've provided the results for values



(a) Normal network

(b) Proposed network

Figure 5: Simulation network models. Taken from [2].

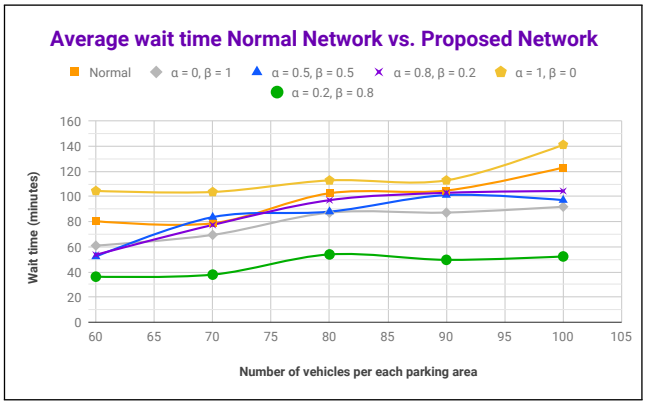


Figure 6: Average wait time results with POIS(15). Based on [2].

0, 0.2, 0.5, 0.8, and 1.

The researchers tested their model for forwarding drivers in the simulations. This is the proposed network model shown in Figure 5b. This shows a vehicle arriving at a full parking area and being forwarded to a new parking area with open spaces. However, they wanted to test their model against a control. This control is the normal network model shown in Figure 5a. Here a vehicle arrives at a full parking area and waits until an open space is available. They believe this model shows how traditional parking is done with no solution for arriving at a full parking area.

3.4.2 Simulation Results

In the simulations, the researchers examined the average time a driver waited to receive a space and the total time that a driver spent parked in the system. The less time drivers are required to spend on these two tasks, the better. Values of α and β that produce the lowest average times are considered to be the most optimal.

Figure 6 shows the average wait time between the normal network model and the researchers' proposed network model. Using an α value of 1 and a β value 0, shown as the yellow line in Fig. 6, produces the greatest wait times of their proposed model. They suggest this is a result of only considering the distance between the two parking areas, and not the amount of full spaces in the target parking area. As a result, there is a high likelihood of being forwarded to a full parking area. The normal network model, shown as the orange line in Fig. 6, has a slightly better average wait time than the worst α, β values of the proposed model. However, it is still much worse than the other α, β values. An α value of 0.2 and a β value of 0.8, shown as the green line in Fig. 6, produces a significantly better average wait time than the other values simulated and the normal network model. $\alpha = 0.2$ and $\beta = 0.8$ are what they found to be optimal for reducing both the average wait and total times.

4. MULTIPLE-CRITERIA SYSTEM

Rehena et al. [3] have decided to mainly focus on improving the reservation algorithm to incorporate driver preferences into the process of finding a parking space. In order for their algorithm to work, they've made assumptions regarding the architecture of the smart parking system. These assumptions closely follow the general parking system architecture provided in Section 2.2. One unique assumption is the use of ultrasonic sensors to track the occupancy of parking spaces. This provides similar information as RFID used in [2]. The reservation algorithm is presented in Section 4.1. The simulation results for this algorithm are shown in Section 4.2.

4.1 Reservation Algorithm

In this section we look at how the Multiple-Criteria Parking Reservation algorithm (MCPR) in [3] determines an appropriate parking area for a driver. This is based on a utility value. This utility value is calculated for each parking area by the MCPR algorithm, which takes into account the preferences of a driver. This algorithm is similar to the cost function shown in Section 3.3.1. However, a higher utility value represents a better parking area for the driver. The utility value for a parking area can be calculated by:

$$U(O_i) = \sum_{k=1}^{m} Z_k(O_i) \cdot W_k, i = 1, 2, ..., n \quad (3)$$

Here $U(O_i)$ is the utility value that's being calculated for parking area O_i. There are n number of parking areas. k is the current criterion. m is the total number of criteria. $Z_k(O_i)$ is the normalized score for criterion k in parking area O_i. This normalized score is in the range from 0 to 1 inclusive. $Z_k(O_i)$ is calculated by:

$$Z_k(O_i) = \left| \frac{O_{ik} - O_{kmin}}{O_{kmax} - O_{kmin}} \right| \quad (4)$$

Here O_{kmin} and O_{kmax} are the minimun and maximum values that criterion k can be, respectively. The normalized score is then multiplied by W_k, which is the weight for criterion k. All of the weights for the criteria sum up to 1.

4.2 Results

Evaluation of the MCPR algorithms performance is done with a simulation tool called MATLAB [3]. This is similar to the Arena simulation tool used in [2]. In this simulation the researchers want to use three different preference sets. These preference sets use different weights for each of the criteria. Using these preference sets, they look at the average walking distance, parking area utilization rate, and the average utilization rate of each parking area.

	P1	P2	P3	P4
C1 (meters)	500	1900	700	1000
C2 (rupees)	50	30	50	40
C3 (open spaces)	50	90	20	80

Table 1: Simulation parking area setup with criteria values. Based on [3].

4.2.1 Simulation Setup

In the simulation the researchers decided to use three different criteria. In an actual system the criteria could be different. They denoted the criteria as Ck, where k is the criterion. C1 is the distance between parking area and the driver's destination. The destination is the same for all drivers. C2 is the price per hour of reserving a space in that parking area. C3 is the number of open spaces in that parking area. Table 1 shows how each parking area, P1-P4, is set up with different values for criteria C1-C3. They ran the simulation for a total of 60 minutes.

Three different preference sets were created using the these three criteria:

- Preference set 1 is equal priority, where W_k for each criterion is 0.33.
- Preference set 2 prioritizes distance, where W_k for $k = 1$ is 0.6, with the other criteria weights being 0.2.
- Preference set 3 prioritizes price, where W_k for $k = 2$ is 0.6, with the other criteria having weights of 0.2.

4.2.2 Simulation Results

Figure 7 shows the average walking distance in kilometers for each of the three preference sets simulated. Preference set 1 has an average walking distance of a little less than 0.8 km. Set 2 around 0.7 km. Preference set 3 has about a 1.2 km average walking distance. As preference set 2 puts more priority on the distance to the driver's destination, it makes sense that this would produce the shortest average walking distance. Surprisingly, the equal priority preference sets average walking distance is only 0.1 km greater than the distance priority. A significant increase in average walking distance is seen in the price priority preference set. It has a 0.5 km greater average distance than the preference set 2. This is the trade off of prioritizing towards a lower price for parking, rather than the distance to your destination.

Figure 8 shows the parking area occupancy rate for preference set 2. The occupancy rates for each parking area don't start at the same rate due to some of the areas being more full than others initially. This is the distance priority preference set. Parking area P1 is the first area to be utilized by drivers. This makes sense as P1 has a distance of 500 meters from the destination. P3 starts to become utilized when P1 is full. P4 is only utilized when P3 is nearly full. P4 being utilized before P3 is actually full could be a result of there being a lot more open spaces in P4, thus the likelihood of finding an open space is greater. P2 isn't utilized at all during the simulations. This is due to it being the furthest away from the destination at 1900 meters. However, if they ran the simulations for longer than 60 minutes, then we would expect P2 to eventually become utilized once the other parking areas become full.

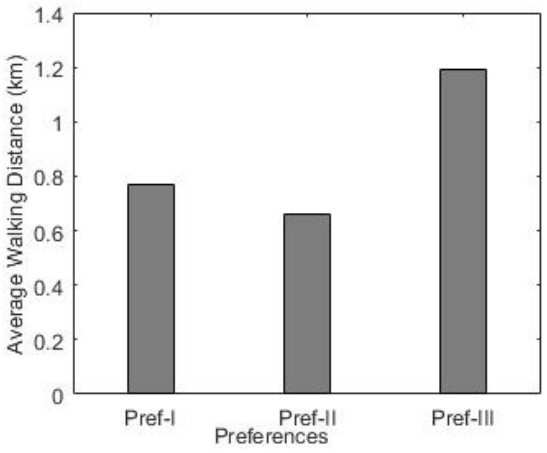


Figure 7: Average walking distance for the three preference sets from Table 1. Taken from [3].

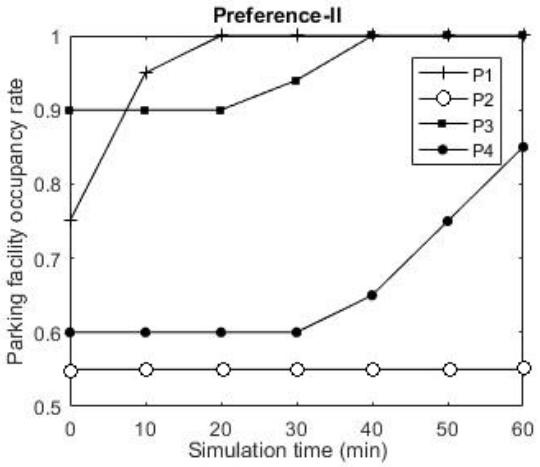


Figure 8: Parking area occupancy rate for preference set two. Modified from [3].

5. CONNECTING SMART SYSTEMS

We've looked at two smart parking systems up to this point. However, both of these systems are closed. This means that no other system can access the information these systems collect, even if doing so could provide new or improved services. As a result, information silos are created.

Kubler et al. [1] present the European Union's vision for IoT systems. Also, they present two IoT standards that have been used to overcome the issue of information silos. Along with this, these standards are used in a sporting event management situation as a proof-of-concept.

5.1 European Union Vision

The current IoT vision of the European Union (EU) is to make their initial IoT systems economically viable [1]. One big reason for this vision is due to the EU's IoT systems being created by many smaller companies. In contrast, the US has a few larger companies starting to create these sys-

tems. The EU has been conducting projects focused on IoT from 2007 to 2015. These projects looked into the creation of IoT systems and architectures. This was the first phase of their IoT programs. The second, and current, phase is to create an IoT ecosystem to allow many different groups to help the overall ecosystem last. To achieve this, the EU has created an alliance called the Alliance for Internet of Things Innovation (AIOTI), with the aim of innovating IoT systems across different industry sectors such as farming or manufacturing. However, to achieve this requires that information can be safely and efficiently exchanged between these different systems. As a result, they've created several research and innovation projects to find solutions for this issue.

5.2 IoT Message Standards

Kubler et al. [1] focus on a project called bIoTope, which uses IoT messaging standards developed in previous IoT projects. Theses standards are the Open Messaging Interface (O-MI), and the Open Data Format (O-DF).

O-MI provides the necessary operations for reading and writing data to and from an O-MI node. An O-MI node represents a server using the O-MI communication standard. These nodes can function as both a server and client. This means that O-MI nodes can communicate with each other and also with traditional servers. O-MI can send information in many different data formats. While the O-DF has been designed alongside O-MI, it isn't a required format.

O-DF provides a generic data format for the IoT. The format can be thought of as a data structure. The server can navigate this data structure to locate specific values.

5.3 Sporting Event System

The proof-of-concept scenario takes a look at a stadium sporting event [1]. The stadium has a smart parking system with four different parking areas. O-MI node 1 is in charge of collecting and posting each parking area's information. O-MI node 2 is located in the stadium office. This node can subscribe to any information that O-MI node 1 has posted, as long as the node has proper access rights. In the example, O-MI node 2 subscribes to the license plate information of vehicles entering node 1. When a vehicle arrives at a parking area, node 2 receives a notification with that vehicle's license plate number. If the vehicle is approved to park there, then node 2 can send a request to node 1 to open its gate.

O-MI node 3 is located at the city's main government building. This node can access information about the status of the stadium by subscribing to the information posted by node 2. At this level, node 3 could also be getting information from other domains around the city. Using this information, the city could locate areas of the city that could use improvements to become more efficient, such as reducing traffic congestion. There are many different benefits that could be provided by having a greater understanding of how a city is currently functioning.

The researchers take the example of cross-domain communication further by introducing a emergency situation in one of the stadium's parking areas. In this situation there has been an accident in one of the parking areas. The hospital receives a notification about the accident. O-MI node 4 is located at the hospital. The hospital can search a list of the cities O-MI nodes to find the one corresponding to the stadium office, node 2. From here they can navigate the data structure to locate the stadium's parking area information, node 1. Now they have access to the status of each parking area. If parking areas 1-3 have a status of busy, but not parking area 4, then the hospital can direct their ambulance to enter through that parking area. This can help reduce the time it takes for an emergency vehicle to respond to the accident. Along with this, the hospital can even open the parking area gate for the ambulance.

6. CONCLUSIONS

Smart systems bring many benefits in terms of user satisfaction and overall efficiency. We've looked at two smart parking systems that have been able to reduce driver time spent looking for a parking space. However, both systems only provide the driver with a parking area, and not an individual space. This is a future research area that could make these systems more accurate. The use of driver preferences in [3] is beneficial as the driver is able to obtain a parking space that they prefer.

Connecting smart systems of different domains is an important next step for these IoT systems. This will allow for new and improved services. The EU is actively working towards this goal using new IoT standards that their previous projects have developed. They've created a proof-of-concept sport event management system which demonstrates the potential of these standards. However, the initial version of these standards aren't suitable for real-time systems, as the amount of data being transmitted isn't small enough to reduce the probability of errors occurring. Thus, research into real-time IoT standards could provide further value for such systems.

Acknowledgments

Thanks to Nic McPhee and Elena Machkasova for their invaluable advice and feedback. Also, thank you to my external reviewer Brian Mitchell for their helpful feedback.

7. REFERENCES

[1] S. Kubler, J. Rober, A. Hefnawy, C. Cherifi, A. Bouras, and K. Främling. IoT-based smart parking system for sporting event management. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 104–114, New York, NY, USA, 2016. ACM.

[2] T. N. Pham, M.-F. Tsai, D. B. Nguyen, C.-R. Dow, and D.-J. Deng. A cloud-based smart-parking system based on internet-of-things technologies. *IEEE Access*, 3:1581–1591, September 2015.

[3] Z. Rehena, M. A. Mondal, and M. Janssen. A multiple-criteria algorithm for smart parking: Making fair and preferred parking reservations in smart cities. In *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, pages 1–9, New York, NY, USA, 2018. ACM.

[4] Wikipedia. Queueing theory — Wikipedia, The Free Encyclopedia, 2018. [Online; accessed 28-October-2018].

[5] Wikipedia. Radio-frequency identification — Wikipedia, The Free Encyclopedia, 2018. [Online; accessed 11-September-2018].