Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Slot-Based Co-operative Driver Guidance System for Public Events

Dylan Walsh

April 11, 2019

A Masters Dissertation submitted in partial fulfilment
of the requirements for the degree of
Master in Computer Science

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: ____________________ Date: ____________________

Slot-Based Co-operative Driver Guidance System for Public Events

Dylan Walsh

Master in Computer Science
Trinity College Dublin

Supervisor: Prof. Vinny Cahill

Abstract

Public events in urban environments often suffer from large amounts of congestion and unpredictable journey times due to the high volume of drivers in the surrounding road network e.g. concerts. To address this problem, the co-operative slot-based driver guidance system is proposed. This system applies traffic shaping measures in an attempt to organise the flow of traffic, leading to more accurate journey time predictions and alleviating congestion.

This project explores whether this slot-based system is a more beneficial form of driving, rather than the current unco-operative approach. The system that was developed focused on analysing the feasibility and potential benefits gained from incorporating this driver guidance into event management plans. The potential benefits include more accurate journey time predictions and an overall better-organised flow of traffic. Existing research has been completed for slot-based driving scenarios for highway on-ramp merging and intersections. However, this project will specifically target public events in an urban environment, particularly around the entry and exit routes to parking lots.

The slot-based solution is similar to time division multiple access (TDMA) in computer networking. In this instance, the road is the channel to be divided up and drivers are allocated time slots on demand to travel within towards their destination. Each driver is provided with real-time guidance information in order to maintain their position within the slot.

An implementation of this system has been developed and tested using PTV Vissim, a traffic simulation software. Initial results indicate that the overall throughput of vehicles in the vicinity of the event is increased and journey times are significantly more consistent during varying traffic volume levels. These slot-based results are in comparison to the results generated from the simulations run using the Wiedemann 74 human driver model within Vissim.

Summary

Research on traffic shaping and more efficient traffic management is dramatically increasing in popularity, due to the increasing number of vehicles on the road. Many institutions and organisations are funding that research and development in this domain. One concept which has been around since the early stages of these areas is the idea of co-operative driving. Co-operative driving is the concept of drivers on the road making informed decisions which benefit each other rather than driving in a selfish nature.

The specific area of interest for this project is using a slot-based co-operative approach to driving, to attempt to alleviate congestion by organising the flow of traffic. This project explores whether this slot-based system is a more beneficial form of driving, rather than the current unco-operative approach. The system that was developed focused on analysing the feasibility and potential benefits gained from incorporating this driver guidance into event management plans. The potential benefits include more accurate journey time predictions and an overall better-organised flow of traffic. Existing research has been completed for slot-based driving scenarios on highway on-ramp merging and intersections. However, this project will specifically target public events in an urban environment, particularly around the entry and exit routes to parking lots.

This project implements a dynamic slot provisioning and control system for all vehicles entering the specified road network surrounding an event. The system simulates the desired driver guidance system with one hundred per cent driver compliance. Each road is allocated a number of slots based on the length of the road. Vehicles that enter any of the roads within the boundary are assigned a slot. The driver guidance system, in turn, provides guidance to the driver to maintain a central position within the assigned slot, as it moves. The guidance provided directs the driver to the specified destination. The slot is propagated along the road at each time step in the simulation until the destination is reached. Vehicles that are attending the event are assigned a parking space in the parking lot and vehicles that are not attending the event are directed away from the event towards their destination. To reduce the complexity of the system, roads are restricted to one lane per direction. There are no turning restrictions at intersections except u-turns and routes that are predefined within PTV Vissim. Results are obtained by observing the simulation.

The slot-based guidance system has been developed in Java, an object-oriented computer programming language. The testing and visualisation of this system were carried out using PTV Vissim, a microscopic multi-modal traffic flow simulation software. This software allows for the creation of road networks, parking lots, vehicle movement and control. Vissim has built-in functionality with the COM API for seamless communication between Vissim and Java. The results of the simulations indicate that average journey times can be more accurately predicted using the slot-based system. In comparison to human drivers which have large differences in average travel times at varying degrees of traffic volume, the slot-based driver travel times remain relatively consistent. The overall throughput of vehicles is also significantly increased at high traffic volume levels by using the slot-based system.

Acknowledgements

I would like to thank Prof. Vinny Cahill for his initial idea for the topic and the weekly guidance given to me throughout the duration of this project. His advice was always helpful and pointed me in the right direction.

Secondly, I would also like to thank my friends and peers who have provided advice and proof reading including Rosee, Darragh, Michael and Sarah.

Finally, I would like to thank my family for their unwavering support and patience during not only this final year, but all five years which have led up to this point.

Contents

List of Figures

List of Tables

1 Introduction

This project seeks to address the problem of congestion in the road network of a public event located in an urban environment. This congestion causes journey times and the overall flow of traffic to be inconsistent at different times of the day or night. The proposed solution is to develop a slot-based driver guidance system which will track vehicles on the road and provide each of them with the required information to maintain their position within a virtual slot for the entire duration of their journey. This will implement traffic shaping measures in an attempt to alleviate the congestion and organise the flow of vehicles in the vicinity of an event.

1.1 Research Question

Does complete compliance with a slot-based driver guidance system positively affect the flow of traffic around events, allowing for vehicle journey times to be more accurately predicted?

1.2 Motivation

Humans are unpredictable by nature and this also applies to their driving behaviour [6]. Each driver has various instincts, opinions, reflexes and skill levels, leading to certain decisions being made while driving. These decisions include turning, stopping, slowing down and speeding up. If two drivers were faced with a similar scenario, both may choose to react differently and complete different manoeuvres. This unpredictable behaviour is amplified in high pressure situations, alongside the input of the other occupants of a vehicle [7].

The majority of events attempt to mitigate the impact of traffic and non-reoccurring congestion during an event through traffic management plans. Non-reoccurring congestion is defined as unusual congestion caused by unpredictable incidents including accidents and planned special events [8]. However, the unpredictable and reckless driving nature of road users ultimately can cause disruption to these events and all the surrounding roads [8]. The cause of congestion experienced by drivers at events can be narrowed down to variations of these four causes: poorly planned traffic management, unexpected events, poorly designed road networks and the drivers themselves [9].

This project proposes a co-operative slot-based driver guidance system as a solution to the problem of congestion in the vicinity of public events located in an urban environment. By removing the need for drivers to make their own decisions whilst within the event and the surrounding road network, one of the four variations of the causes of congestion is removed [9]. This project will focus on whether or not this has a significant impact on the flow of traffic in the vicinity of the event at a driver compliance rate of 100%.

The hypothesis is that a more organised approach will alleviate congestion and improve the predictability of travel times for drivers who are attending the event, and also those who are driving in the area and not attending the event.

1.3 Project Overview

1.3.1 Co-operative Driving

Co-operative driving is a specific approach to driving, whereby there is some form of communication between drivers on a road network. Every co-operative driver is in pursuit of a common goal. This communication allows for the vehicles to make decisions which will benefit other vehicles in the network [10]. This communication is either vehicle to vehicle (V2V), vehicle to infrastructure (V2I) or vehicle to central control to vehicle (VCV) communication.

1.3.2 Time-Slot Drving Approach

The time-slot approach follows the same principles as Time Division Multiple Access (TDMA) in networking. TDMA is a channel access method (CAM) which is used to allow channels to be shared among senders without any interference between the packets being sent [11]. When this is applied to a road network, the channel to be divided up is the road, and each vehicle is the equivalent of a packet which is assigned a slot on demand.

Combining the Time-Slot Approach and Co-operative Driving

The combination of co-operative driving and the slot-based approach allows for a more advanced driving system to be created. The co-operative driving portion of the system retrieves real-time information from all of the vehicles in the network which will then be used by the centralised guidance system, to provision slots for the vehicles and manoeuvre all vehicles accordingly. Real-time information gathered from the vehicles includes origin, destination, current speed and position in the road network. The central guidance system will, in turn, assign slots to the vehicle as they become available. The slot will act as a guide for where the driver should position themselves. The system will identify the required speed and direction for the car to drive at to prevent collisions with other vehicles, while allowing the vehicle to make progress towards the final destination in a more co-operative fashion. In regards to simulating this guidance, the slot-based system will assume full control of the vehicles in the simulator, replicating a 100% driver compliance rate. This system will allow for cars to be routed towards available parking lot spaces when the driver is attending the event.

1.3.3 Research Aims

The aim of the study is to prospectively evaluate how the flow of traffic in the vicinity of an event is affected by the introduction of a slot-based driver guidance system, and in turn, whether or not journey times can be more accurately predicted [2]. The specific objectives of this study are:

- Create a slot-based driver model to provide drivers with guidance on roads and parking locations of varying sizes in the vicinity of an event.
- Examine the resultant effects on the flow of traffic at multiple driver volume levels.
- Design test scenarios, using various road networks within the PTV Vissim simulation software.
- Analyse how the degree of compliance with the driver guidance system affects the flow of traffic.

1.3.4 Potential Benefits of this Research

The potential benefits of this research include improvements to the congestion on roads in the vicinity of events. Another benefit is a reduction in congestion levels and the amount of accumulative time spent stationary without making any progress on the roads. This is in comparison to the current unconstrained approach to driving. There is also potential positive impacts on the intelligent organisation of parking lots [12], whether these are permanent or temporary parking locations.

1.3.5 Description of Research Area

The direct area of research for this project is traffic management systems, specifically the area of co-operative vehicle systems. Co-operative driving targets the safety and efficiency of road traffic [13]. Inter-vehicle communication means that guidance can be directly provided to the driver of the vehicle in order to make more informed decisions and reduce dangerous driver behaviour.

1.3.6 Project Scope

This project will solely focus on whether any benefits can be gained from a basic implementation of the system at a 100% compliance rate. The project will not take into account various safety aspects which must be considered before a production release. This is due to the fact that rigorous testing cannot be completed outside of the simulation environment and the simulation software also has testing limitations as to what can and cannot be controlled within the simulation. The discussion of how information could be transferred between drivers and the system is also out of scope for this project. Potential security and privacy concerns will be briefly discussed to highlight the future research and work required for this project to be applied to physical real world scenarios, however, it is not a primary concern for this research project.

1.3.7 Road Map

The following sections will begin with several pieces of related research about slot-based driving, parking lot design and parking management systems. Following on from the research that currently exists, the design and implementation of the solution will be examined and the main challenges encountered during the project will be discussed at length. This chapter will also include information on the intended implementation of the system and what was actually completed within the time frame. Finally, the results of all simulations run with the slot-based system will be compared to the control cases within the simulator, accompanied by analysis and evaluation of the overall project.

1.3.8 Key Words

Search strategy keywords: Co-operative driving, Car following model, Driver model, Vissim, TDMA, time-slot driving, slot-based driver, driver guidance system, driver guidance mobile application, traffic management, traffic flow, traffic stream, traffic density, calculating vehicle journey times, traffic shaping, parking lot management, parking lot.

2 Background

This chapter provides an insight into the multiple areas related to this project which were examined at the beginning of the project. It takes a deeper look at existing research which was completed in the domains of slot-based driving, traffic management systems, co-operative driving and parking lot management systems. The papers which were examined are discussed in relation to events and with the design of the new driver guidance system in mind. Third-party software and tools are also discussed, to provide insight into why they were chosen to be used in this project.

2.1 State of the art

The current state of research on slot-based driving highlights specific circumstances where benefits can be gained. These circumstances are focused on particular use-cases and driving scenarios, which are more specialised than an entire control system for driving in the urban road network surrounding a public event e.g a concert. To date, there is very little publicly available research on slot-based driving, which would suggest benefits could be gained from using this approach within the traffic management plan of an event. There are two key elements to this project: the road network surrounding an event and the parking lots for attendees.

2.2 Related research

As previously mentioned in this report, there are several research papers which have attempted to implement and analyse the slot-based driving approach in different driving scenarios. These papers include street intersections [14], on-ramp merging on highways [15] and guaranteed arrival times with slot-based traffic shaping [2]. Work has been carried out on analysing parking lot layouts, smart parking and optimising parking spaces. This will also be discussed in this section.

2.2.1 The Managed Motorway

The Managed Motorway paper details a very early novel approach to vehicle scheduling on a motorway type road network [1]. This approach proposes a system which divides the available road space into slots. Vehicles are assigned slots for which to travel in for the duration of their journey. This paper relates this system to the channel access method of TDMA, which is used to allocate slots to messages as they are transported through a network. However, this TDMA-approach is modified to allow for unexpected events to occur, which is common in vehicular traffic. The types of unexpected events listed in the paper vary from vehicles suffering from a malfunction to pedestrians appearing on the road. To cater for these unexpected events, a local real-time vehicle coordination system is suggested. Vehicles can communicate in an on-the-fly type situation to adapt to these unexpected events in a way which does not drastically disrupt the overall global scheduling of slots. The system shifts to a more local coordination style approach where the inter-coordination of affected vehicles supplements the global slot-based control system.

The paper assumes either the driver or an on-board cruise control system is responsible for maintaining the position of the participating vehicle within the assigned slot. Where the driver is the responsible party, a 'heads-up' display is proposed which will provide feedback. Also in the case of the driver, the researchers suggest the use of enforcement mechanisms to ensure driver compliance with the system.

The architecture for the managed motorway is proposed as being a series of entry and exit ramps with a selection of lanes in both directions. Emphasis is placed on one direction of the motorway in particular. Any motorway structures or irregularities are ignored (bridges, tunnels, medians). Hard shoulders are also excluded. Before entering a motorway, vehicles are assumed to have all the requirements necessary to comply with the system. Vehicles remain in the queuing lane until a slot becomes available on the motorway.

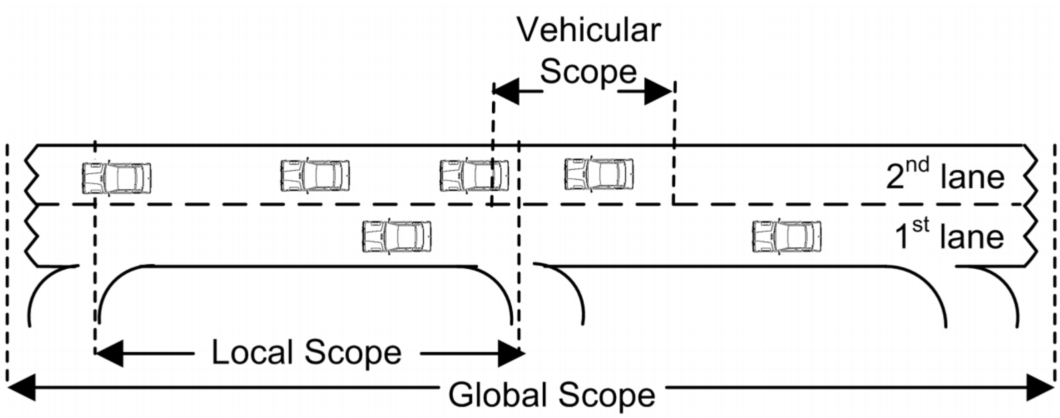

Figure 2.1: The Managed Motorway [1]

The researchers identify the concept of scopes as a method for managing the motorway. The global scope represents the entire motorway and manages the capacity and coordination of all local scopes. The local scope covers a specific segment of the motorway e.g. between entry/exit ramps. This scope includes a local control system and is responsible for managing all vehicles within the motorway segment of which the scope has been assigned. Each local scope collaborates with other neighbouring local scopes to coordinate the hand-off of vehicles. Finally vehicular scope is only responsible for the vehicle and the vehicle's direct surroundings through sensors. Both the local scope and vehicular scope make decisions which comply with the global scope.

Admission control is stated as being required to ensure safety measures are met and the number of available slots on a motorway cannot exceed a peak number. This will prevent safety distances from being shortened and allow for local coordination to be less restricted by the smaller available area. This is enforced by tollgates situated on each on-ramp to the motorway. The tollgates check the availability of slots through communicating with the global scope. This paper concludes by proposing various types of solutions for allowing vehicles to communicate with each other, which is beyond the scope for this project. Marinescu et al.[2] run tests over a highway distance of 2472 metres, the tests included four different traffic volume levels: 2000, 3000, 4000 and 5000 vehicles per hour. The human driver simulations were controlled using Vissim's Wiedemann 99 driver model. Results showed that at lower traffic volume levels, travel times were consistent for both human drivers and slot drivers. However, at higher traffic volume levels (5000), the human drivers had a travel time increase of up to 400%. On the other hand, the slot-based drivers had similar times to the slot-based drivers at lower traffic volume levels. Hence, this data proved the hypothesis that the slot based approach is a definite improvement for guaranteeing travel times in the presence of bottlenecks.

2.2.2 Street Intersections

This paper discusses the potential benefits of implementing what the researchers call Slot-based Intersections (SIs) [14]. The motivation behind their research was the apparent lack of a comprehensive analytical framework to compare the SIs to existing traffic light intersection access systems. Their goal for this project was to create this framework, where their implementation of SIs could be examined against current traffic light systems [16].

The research identifies intersections as the common shared resource in urban road networks which are a bottleneck, and therefore, needs an organised system in place to coordinate vehicles travelling on conflicting paths. The coordination of vehicles is achieved through a complex switching process, allowing intersection access to vehicles travelling on different paths which do not conflict. Traffic lights currently carry out this coordination through a periodic series of phase switching operations. A phase is identified as a time period whereby only a certain number of vehicles travelling on non-conflicting paths are allowed to enter and pass through the intersection.

This paper examines the delay required when switching between phases (the amber light) which generally lasts up to 8 seconds [17]. This delay directly impacts the volume of traffic entering and exiting the intersections every hour. More frequent phase changes directly lead to an increase in the number of delays occurring at the intersection every hour. The delays are required to ensure smooth transitions between phases, and hence, this increase in delays leads to a decrease in the overall capacity of an intersection per hour.

The advancements in technology and opportunities it provides in vehicle to vehicle and vehicle to infrastructure communication enables co-operative driving and slot-based approaches to become a reality [18]. Past SIs provide information to vehicles, identifying the speed at which they need to travel, in order for them to reach the intersection at the beginning of their assigned access slot. The purpose of the SI is to maximise the capacity of the intersection while minimising the total number of delays in switching between phases. If consecutive vehicles are grouped, intersection capacity is increased. However, the delay is increased for vehicles who need to slow down to allow the group of vehicles to pass before they can obtain access to the intersection. The researchers have developed two main strategies in finding a balance between capacity and delays at intersections, FAIR and BATCH. FAIR seeks to establish fairness between vehicles and grants access slots to vehicles on a first come, first served basis. BATCH is a strategy used to increase capacity by granting slots to groups of vehicles and attempting to form platoons of vehicles that can access the intersection over a short period of time.

The research carried out concluded that the evidence gathered during simulations of their system, gave support to the theory that slot-based intersections offered improvements versus traffic lights. The results identified an increase of up to twice the intersection capacity of traffic light systems for the BATCH strategy and even better reductions in the delays occurring at these intersections for both the BATCH and FAIR strategies. This higher performance of SIs is determined to be caused by the increase in flexibility, finer granularity in merging traffic flows and better usage of road space [14]. Limitations of their research are stated as, further work is required in order to scale the analysis to a network of multiple road intersections. A bonus benefit of the SIs is stated to be a reduction in car emissions caused by the 'stop-and-go' effect from the different driving styles of humans [14].

2.2.3 On-ramp Merging

Another paper in which slot-based driving systems were applied, examined the on-ramp merging of vehicles on a highway. Under heavy traffic, congestion develops due to the inefficient ways in which merging manoeuvres are completed by human drivers. The research conducted by Marinescu et al. proposed an optimised merging algorithm based on their previous work mentioned in 2.2.1 of this paper, which uses a slot-based system to coordinate vehicles on both the main highway and vehicles merging between the highway and on-ramp [15].

Research conducted by Shladover [19] showcases how a maximum of 5% of the total road surface of a highway is utilised at any one point, due to safety measures between vehicles. Another research paper by Chen et al. [20] analyses how congestion is caused by poor operation of motorways under heavy traffic conditions. The motivation behind the paper by Marinescu et al. is in reducing the amount of congestion on highways using this co-operative slot-based system, which in turn allows for journey times to be more accurately predicted. The work by Marinescu et al. defines a slot S as S = z, p, t, b, o, z is the size of the slot, p is the position of the slot at the time t and b represents the behaviour of the slot in terms of acceleration, deceleration and lane changing manoeuvre. Finally, o represents the occupancy status of the slot (free or vacant) [15]. Vehicles are assumed to be equipped with radar, DGPS, wireless communication and are (semi)-autonomous. The proposed TMS system assumes vehicles drive under their own guidance until a threshold is reached and the TMS decides that more efficient management of the highway is required to prevent congestion.

There are two approaches examined for determining a slot's availability. The first approach discussed is a hierarchical approach. This approach instructs the TMS to maintain occupancy information for all slots on the highway. If a car wishes to move into another slot, the TMS accepts or rejects the change, rejections occur if multiple vehicles attempt to access slots at the same time. This approach relies only on vehicle to infrastructure (V2I) communication. The second approach is the distributed approach. This approach relies on vehicle to vehicle (V2V) communication. When a vehicle wishes to change slot, this manoeuvre is then coordinated with all surrounding vehicles using V2V. The use of V2V communication and coordination makes this approach scalable in comparison to the hierarchical approach. However, it requires a complex protocol and group communication which was beyond the scope of the paper.

The final proposed approach combines the best aspects of both the hierarchical and distributed approaches. Vehicles on the road use the distributed approach to coordinate. However, a roadside unit (RSU) acts as a proxy between vehicles on the main road and vehicles on the on-ramp, enabling vehicles to communicate with the RSU (Vehicle to Infrastructure communication) to request a slot. The RSU uses V2V to determine suitable slots with existing vehicles on the highway. An optimised version of the merging algorithm discussed in [2] is used to allow merging between vehicles on the on-ramp and main highway. This optimisation allows for the merging algorithm to utilise all available slots across all lanes, not just the first lane.

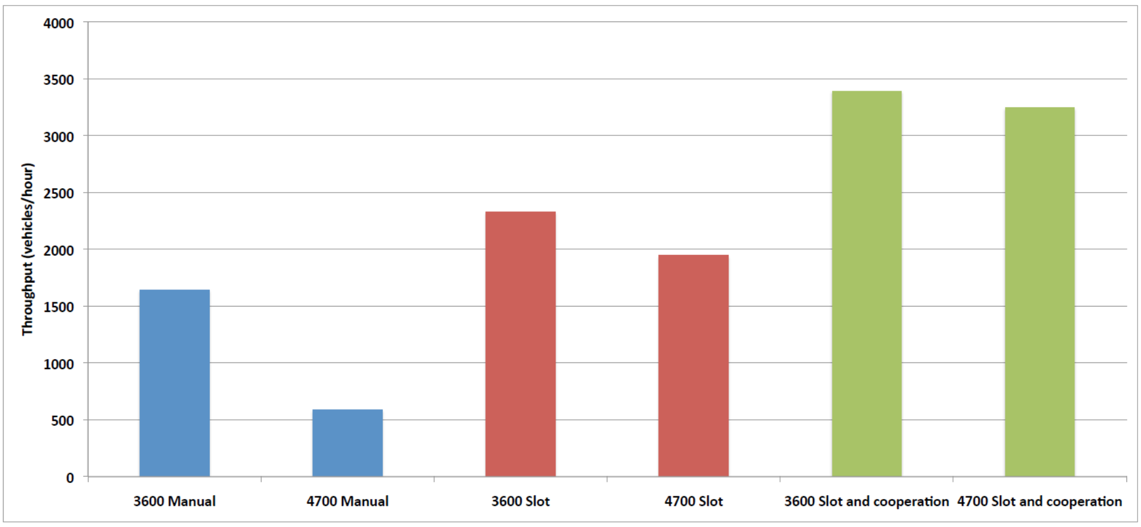

Figure 2.2: Maximum on-ramp throughput [2]

Results from the research concluded that under medium traffic conditions the slot-based driving without cooperation achieved a 41% increase and the slot-based driving with cooperation achieved a 106% increase for heavy traffic conditions in comparison with PTV Vissim's human driver model. Under heavy traffic conditions, both approaches achieved a 230% and 452% increase in throughput compared to the human driver model. The results conclude that slot-based approaches can create a more efficient merging manoeuvre between on-ramp vehicles and vehicles on the highway.

2.2.4 Optimisation of Parking Spaces

The increase in car usage in recent years has not been sufficiently matched by the advancements in the organisation and planning of vehicular infrastructure. A case study by researchers from the University of Teknologi examined different layouts of parking spaces. The three different parking designs that were examined were: parallel parking, perpendicular parking and diagonal (angled) parking. The parallel parking design positions cars in a line with the front bumper of one car facing the back bumper of an adjacent car. This design layout is more common for on-street parking layouts, however, this design may also be used in parking lots and other parking structures. The perpendicular design organises cars in a format with which each car is parked side by side. Each car is parked so that either the front or back bumper is perpendicular to the curb, wall or line which separates the aisles of parking spaces. Finally, diagonal parking is designed so that each parking space is positioned at an acute angle to the aisle from which cars approach the parking spaces.

Three mathematical algorithms were proposed for calculating the maximum number of parking spaces that can be created from a given available area. The first of the three being parallel parking. The first step in the algorithm for parallel parking is to generate the set of all the required variables. $P = \{w, k, y\}$ where parking lot P is represented by w the width of the lane, k the length of the lane and y the length of available parking space. The exact area available for parking spaces needs to be measured and finally, the maximum number of spaces is calculated using:

$$\text{Number of Parking Spaces} = \frac{y}{k} \tag{1}$$

The algorithm for calculating the maximum number of diagonal spaces in a region is calculated by $P = \{x, z, \theta, k, w\}$ where parking lot P is represented by the curb length x, the depth z, the angle θ, the lane length k and the lane width w. The values of w, k and z are calculated according to the value of θ by using:

$$\tan\theta = \frac{w}{z} \text{ and z} = \frac{w}{\tan\theta} \tag{2}$$

The minimum space for a lane is then calculated using:

$$\text{Minimum area of parking space} = (w \times k) + \frac{1}{2} \times (z \times w) \tag{3}$$

Finally, the number of lanes to optimise the area of available parking spaces is determined by:

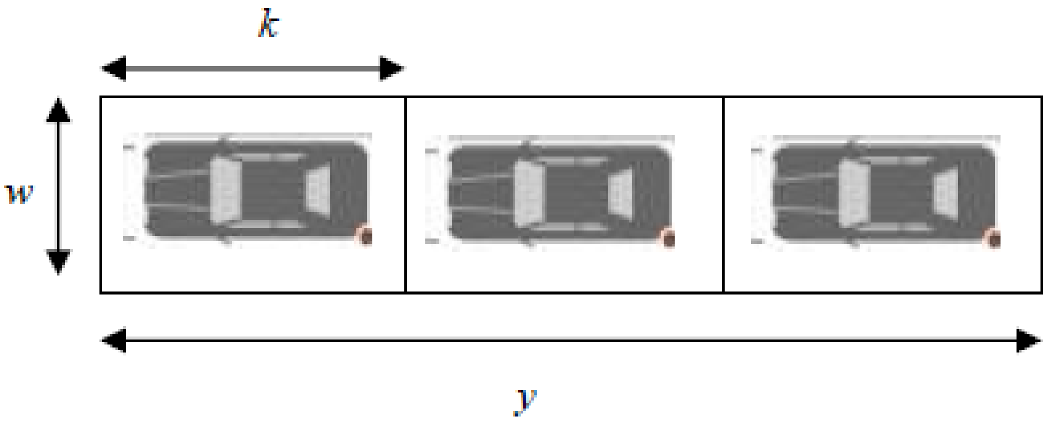

Figure 2.3: Parallel parking layout [3]

$$\text{Number of spaces} = \frac{\text{Area Available}}{\text{Minimum area of parking space}} \tag{4}$$

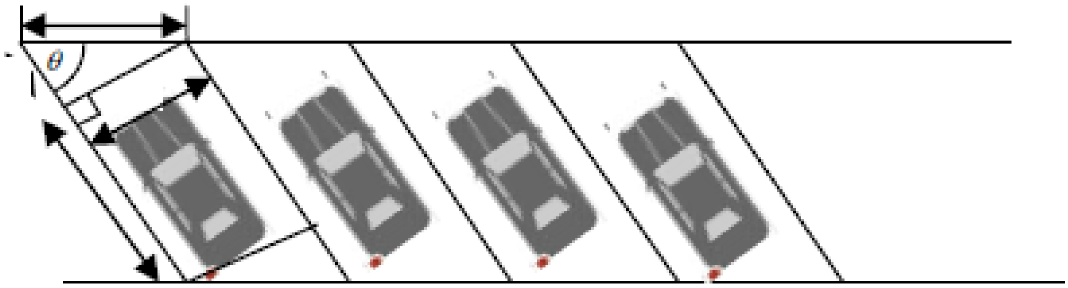

Figure 2.4: Diagonal parking layout [3]

The third design is of the perpendicular parking lot spaces. $P = \{w, k\}$ where w is the width of the lane and k is the length of the lane. The minimum space for a parking space is determined by $k \times w$. Finally, the maximum number of parking spaces that can be created is calculated by:

$$\text{Number of spaces} = \frac{\text{Area Available}}{\text{Minimum area of parking space}} \tag{5}$$

The researchers applied these three designs to the total parking area they had available, the results showed a noticeable increase in the number of parking spaces in comparison to the parking lot design that was currently in use for the available parking area. This current design of their parking lot had 127 parking spaces, however, each of the new optimised layouts was calculated to create 166 perpendicular spaces, 129 diagonal spaces or 100 spaces for the parallel design.

However, while the perpendicular design was the best approach for creating the most parking spaces in the given area, in [21] evidence is provided for specific designs where angled parking

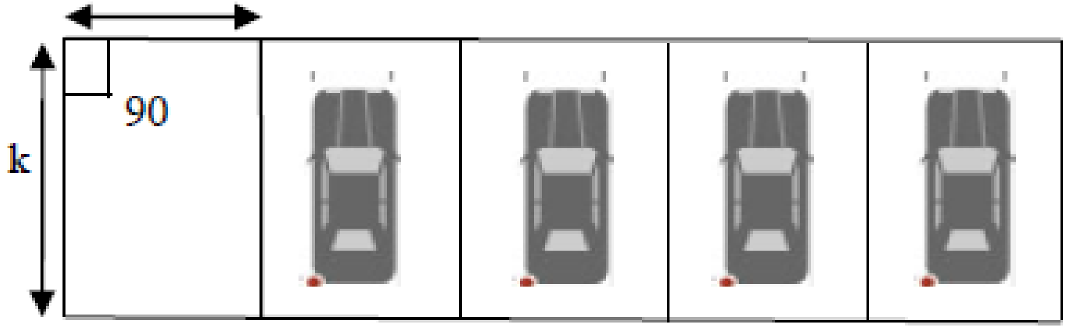

Figure 2.5: Perpendicular parking layout [3]

is the more optimised design. In a parking lot of eighty spaces, the diagonal space layout uses less floor area: 22,532 Sq ft. in comparison to 24,200 Sq. ft used by the perpendicular layout when the use of turning bays are required. This is caused by the need for perpendicular spaces to have wider entrance lanes to allow more room for vehicles to manoeuvre in and out of the 90 degree spaces. Furthermore, the diagonal space design requires only one, one-way entrance lane of regular width, due to the vehicles being restricted to only entering the parking spaces from one side. Diagonal parking spaces also require significantly less lane width to manoeuvre in and out. However, in [22], concerns are raised regarding the approach to the diagonal parking lot spaces. Drivers may approach the spaces the wrong way and attempt u-turns or reverse backwards as an attempt to drive into the free space, even if there is not enough room for most vehicles to make these manoeuvres in tight parking lot environments, which can further cause congestion and unpredictability.

2.2.5 Smart Parking

Dsouza and Hussain [23] propose a system for an entire intelligent parking lot management system. This will allow for drivers to book a parking space and be directed to the space, once it has been booked. The implementation utilises several modern technologies to track the availability of the spaces by leveraging existing CCTV infrastructure and computer vision algorithms to process the video feed and determine whether a parking space is occupied or not. This occupancy information is sent to a centralised control, where the booking of spaces is managed and payments are handled for the use of each parking space. The proposed system will include marker based navigation through an augmented reality interface which will guide the user to the booked parking space.

The concept of pre booking parking spaces is not a new concept, however, utilising modern technologies will improve the ease and efficiency of the booking procedures. In [24], a parking lot reservation system is proposed which is accomplished through the use of the Internet of Vehicles and intelligent transport systems. IoV is described as being an integrated network system which connects different people in different vehicles and environments within an urban setting. The proposed parking lot reservation model has three main components: the co-operative network of drivers, parking lots and a control system [24]. This model contains parking intention parameters, which includes a walking distance threshold from parking lot to destination, the parking duration, the type of parking space required (disabled, motorbike, road level) and also the threshold parking fee. These values are consumed by the control system when evaluating the best parking space to recommend to the driver. Weighting is used to determine the order of preference for these parameters.

Two main algorithms are used by the system. The first is the candidate parking lot selection algorithm, the walking distance threshold is screened and all parking lots within that distance from the destination are returned in a set. The parking lot types are then screened against the set of parking lots, which were found in the previous step. All parking lots which do not match the user's requirements are removed. The parking lot fee is then screened against the set of parking lots and any parking lots above the fee threshold are removed from the candidate set. Finally, any unavailable parking lots are also removed from the set, all remaining parking lots are considered candidate parking lots. If no suitable parking lots have been found, the driver is notified to lower their demands, and the algorithm repeats to find a suitable parking lot.

The other fundamental algorithm used by the reservation system is the optimal parking lot selection algorithm. The algorithm uses the newly created set of candidate parking lots and re-screens the users preferences against this set. One parking lot is chosen which is both available and the best option, given the weights provided for each parameter by the user. The researcher's evaluated their system by studying the occupancy rate for three parking lots in the same vicinity: A, B, C. Observations concluded that drivers spent significantly more time than was necessary queuing to enter parking lot A during peak times, even though parking lot B and parking lot C had a large number of unoccupied spaces. When the algorithms were applied to this data, it was found that the utilisation rate of parking lot A remained high, as usual, however, the utilisation rates of B and C improved immensely during peak times. Hence, the overall utilisation rates for the area containing the three parking lots improved by using this optimal reservation algorithm in comparison to blindly choosing a parking lot.

2.2.6 Parking Lot Induction Method

This paper analyses the main considerations of a driver on how to choose a parking space, in turn proposing a solution for path optimisation within parking lots by using a modified version of Dijkstra's algorithm [4]. To eliminate the congestion caused by drivers spending too much time driving around parking lots, blindly attempting to find an available space, the modified Dijkstra's algorithm is applied to enable the driver to find an available parking space quickly and accurately.

The parking system model uses Dijkstra's algorithm to calculate the distance between two given points. Dijkstra's algorithm is described as a graph search algorithm that solves the single source shortest path problem [4]. The traditional algorithm is used to find the shortest path between nodes on a graph, which can be applied to a road network. Each road is constructed of a series of nodes which are all connected. Connecting roads share a node and parking lot spaces are each connected to the graph by a single node.

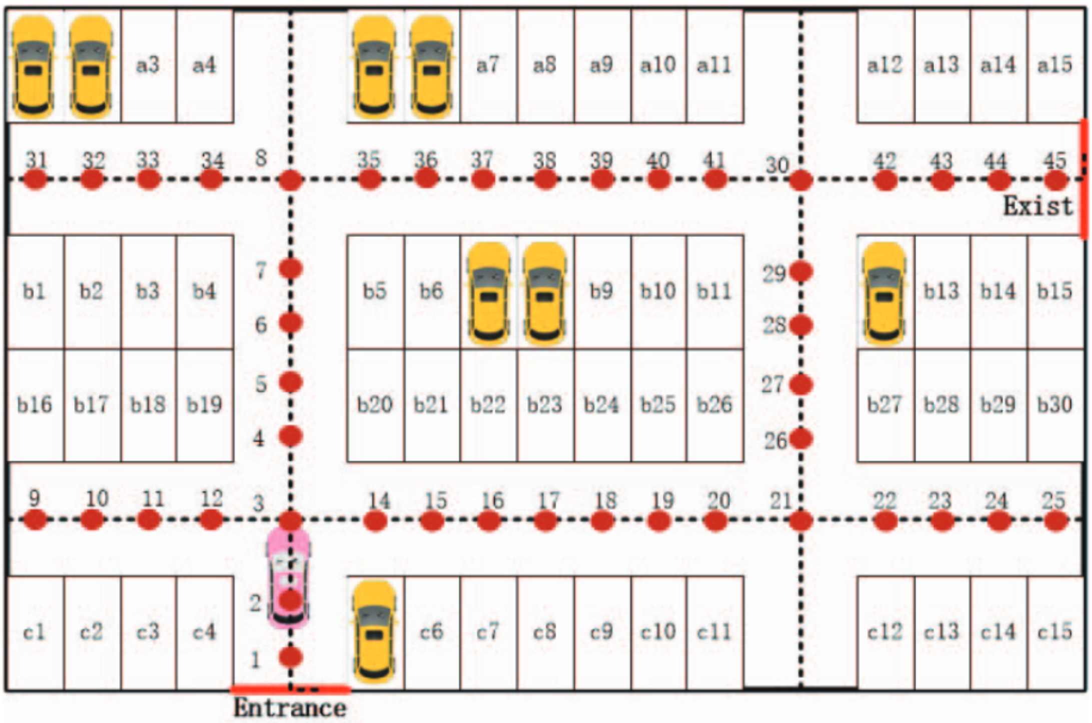

Figure 2.6: Parking lot layout as a graph [4]

The proposed solution is to have sensors monitoring the occupancy status of spaces; the best of the available parking spaces is found and the shortest path from the entrance to the parking lot is shown on the screen to the next incoming vehicle. However, the original Dijkstra algorithm returns the shortest path from one known source node to all other nodes, but for the researcher's proposal they only required the shortest path between two known nodes. The modified algorithm finds the shortest path from source to destination while excluding all other paths and accounting for factors including driving distance, the number of vehicles currently on the same route and the occupancy status of spaces (the weights between nodes are adjusted to account for these factors).

[4] solves the problems of:

- the drivers not being alerted to when and where a parking space becomes available
- the drivers not being allocated optimal parking spaces and many driver's driving to the same space, when a driver has been allocated a spot further away but finds an alternative along its route (which has been assigned to someone else)
- no parking lot induction, causing the drivers to spend a lot of time finding parking spaces by themselves, without guidance.

2.2.7 Vehicle Networking

Vehicle Ad Hoc Networking (VANET) is a specific adaptation of Mobile Ad hoc Networking (MANET). This solution allows for vehicle to vehicle (V2V) communication. Vehicles equipped with the appropriate VANET technology can communicate with all other equally equipped vehicles in their vicinity. The direct use cases of V2V are to allow for a more low level co-operative approach for vehicles to self organise as necessary. Vehicles can form platoons or prevent collisions, based on the information they receive from the neighbouring vehicles. Each vehicle and road side infrastructure forms a node in the network. The information gathered from each node is sent to every other node in the network. The accumulated data is analysed and informed decisions can then be made by the vehicles [5]. The nodes in the network are free to join and leave the network as per design, VANETs are an open and fault tolerant. While conventional MANETs are composed of nodes communicating without a central network, the freedom given to nodes by MANET to leave or stay at will, a variation on the routing protocol is required. Multi-hops are used to allow for the efficient transferral of information over a longer distance.

VANET assumes unlimited power and storage capabilities of each node, however, vehicle to broadband (V2B) communication enables vehicles (nodes) to send and store their information in the cloud [25]. This circumvents the challenges of having large physical on-board storage for each node. V2B also is a solution to allow a central control system to provide input to the VANET and provide external guidance to the nodes. In [25] the two main applications of VANETs are considered to be:

- Safety applications, increasing the overall safety of road and other users by implementing collision avoidance, co-operative driving and traffic optimisation.
- Infotainment application, providing drivers with information that hold entertainment value e.g nearest car repair station, parking lot, supermarket.

2.3 Technology Used

This section will briefly provide detail on the types of software and technology used in the development of the slot-based system. The four specific cases that are covered are OpenStreetMap, the publicly available OSM Java library, OSM-Commons and finally, PTV Vissim.

2.3.1 OpenStreetMap

The OpenStreetMap (OSM) collaborative project aims to deliver free geographic data to anyone. The collection of mapping data is created by local contributors and is supported by the OSM foundation. The data is open data and free to use for any purpose, on the condition that openstreetmap is given credit alongside its contributors. OSM data is exported in either XML or PBF formats. OSM utilitises a topological data structure consisting of four main components:

- Nodes - points with a pair of latitude and longitude coordinates. Nodes are used to represent ‘ways’ or other significant points of interest on a map.
- Ways - an ordered collection of nodes which create a line. A way denotes linear elements including roads and rivers.
- Relations - used to represent the relationships between existing nodes and ways.
- Tags - attributes which contain additional information about each map object (node, way, relation). Tags are used to store speed limits, names, road types etc.

OSM-Common

Java is the programming language used to develop this slot-based system, a connection was required between the Java system and OSM. OSM-Common is a Java library for accessing OpenStreetMap services [26]. This library allows for parsing and processing of OSM data. The supported OSM APIs include OverPass and Nominatim. OverPass is a read-only API that returns custom segments of OSM map data. The overpass API is optimised for accessing large amounts of data in a short period of time, in comparison to the main OSM API which is used for both reading and editing the map data. On the other hand, Nominatim is a search engine for OSM data. The search engine accepts queries constructed of a name and address, returning OSM map data relating to the query.

2.3.2 PTV Vissim

PTV Vissim is a microscopic multi modal traffic flow simulation software developed by PTV Planung Transport Verkehr AG. This software is used to model and simulate road networks of vehicles. The software can simulate many types of traffic including vehicles, public transport and pedestrians. The software package uses an optimised version of the human car following model developed by R. Wiedemann. Vissim provides a graphical user interface for the modelling of simulations. For more in depth control of simulations, Vissim is provided with the COM interface which is a hierarchical model in which functions and parameters of the simulator that are normally controlled by the GUI can be controlled by external programming [27]. The COM interface provides control to any programming language which is able to handle COM objects including C++, Java and Python. The interface allows for the dynamic manipulation of a selection of Vissim object attributes. The main elements of PTV Vissim are:

- Links - objects within vissim which denote a continuous strip of road.
- Connectors - objects which are used to connect two links.
- Routes - objects used to indicate a course to follow along a series of links and connectors.
- Vehicle Inputs - the starting point from where vehicles and other moving objects will be generated from.

2.3.3 Simulation Software Comparisons

To evaluate the system, the use of traffic simulation software is required. Modern traffic simulation software includes various car-following models, which enable the user to simulate human drivers in custom road networks and scenarios. The two simulation software packages that were examined for this project were Simulation of Urban Mobility (SUMO) and PTV Vissim. SUMO is an open source traffic microscopic and continuous road traffic simulation package which is capable of handling large road networks. Vehicles move under the control of the default car following model (others may be chosen) in this simulator, collisions and accidents are simulated. Vehicle behaviour is taken into consideration for manoeuvres like changing lanes. Roads are shown as a combination of lanes and each lane has a fixed width. Vehicle width is also fixed and SUMO has various APIs that allow simulations to be controlled remotely.

On the other hand, PTV Vissim, as discussed in the previous section 2.3.2, is a licenced microscopic multi-modal traffic simulation software package. Vissim is one of the most used traffic simulation software on the market. Vissim allows for the user to define custom vehicles and other road users. The biggest advantage of Vissim is the component object model (COM) interface. The COM API allows a user to programmatically control the Vissim simulation using many programming languages. Currently the COM interface is unmatched in other traffic simulation software packages including SUMO, as it allows access to signal control, path flows, vehicle behaviour and the entire road network topology. This enables the user to model complex flows, transportation systems and components, including custom driver models which was the most important component regarding the development of the slot-based driver guidance system.

Simulators	Model			Category		System		Visualization		Infrastructure: Difficulty			Infrastructure: Flexibility			Vehicles and pedestrians					Scope Area			Detectors		GIS		
	I	*E*	*A*	*O*	*C*	*D*	*C*	*2D*	*3D*	*E*	*M*	*D*	*F*	*L*	*VL*	*T*	*D*	*R*	*E*	*O*	*I*	*R*	*O*	*WD*	*WL*	*Y*	*P*	*N*
AIMSUN	√	√	√		√		√	√	√			√	√					√	√	√		√	√	√	√	√		
ARCHISIM	√				√	√		√				√			√						√			√				√
CORSIM	√				√	√		√	√						√			√	√		√	√		√				√
MATSim	√			√			√	√						√							√	√		√		√		
MITSIMLab	√			√				√						√							√			√				√
Paramics	√				√	√		√	√				√					√	√	√	√	√		√	√			
SimTraffic	√				√			√	√	√				√				√	√		√	√		√				√
SUMO	√			√			√	√				√			√						√	√		√				
TRANSIMS	√	√		√		√		√							√							√	√	√				
TransModeler	√	√	√		√			√	√					√					√			√	√	√		√		
VISSIM	√				√		√	√	√	√			√					√	√	√	√	√		√	√	√		

Figure 2.7: A comparative table of traffic simulators [5].

As shown in Fig.2.8, in the comparative study of all traffic simulators, Vissim is the only simulator to achieve an easy difficulty rating and "Flexible" infrastructure flexibility rating in terms of its infrastructure. It is also centred around urban road networks, which is the appropriate network

Criteria		Abbreviation	Signification
Model		I E A	Microscopic Mesoscopic Macroscopic
Category		O C	Open-source Commercial
System		D C	Discrete Continuous
Visualization		2D 3D	Two-dimensional Three-dimensional
Infra-structure	*Difficulty*	E M D	Easy Medium Difficult
	Flexibility	F L VL	Flexible Limited Very Limited
Vehicles and pedestrians		T D R E O	Type Dimension Priority Pedestrian Other vehicles (Bus and Tram)
Scope Area		I R O	City Region Country
Detectors		WD WL	Wired sensor Wireless sensor
GIS		Y P N	Yes Partially No

Figure 2.8: Legend for the comparative table of traffic simulators Fig.2.8 [5].

type for testing the slot-based driver guidance system in an event scenario. Hence PTV Vissim was chosen as the most suitable simulator for this project.

2.3.4 Conclusion

The related work discussed in this section were in the areas of slot-based driving and parking. The previous work on slot-based driving showcased the benefits that can be gained from using the approach and also the considerations that must be made when doing so. However, the slot-based driving research targeted single instances of specific road segments including highways, on-ramps and intersections. The chaining of intersections and on-ramps has not been examined.

This project will continue on from the work completed on slot-based intersections by accounting for the chaining of intersections. The concept of scope will be used from [2], to have different entities controlling specific segments of road. The various levels of scope will be at the global scope of the entire system, parking lot scope of the parking lot, the local scope of each road and finally the vehicular scope of each vehicle. This project will apply similar considerations to the previous work on intelligent parking lot management system and parking space reservations, however, this system will implement these elements using the slot-based driving approach.

3 Design and Implementation

This chapter details how the slot-based cooperative driver guidance system was implemented and how it functions. Each component of the system is discussed, whether they were included in the final implementation or not. Explanations for design choices are also included, to allow the reader to understand how the project developed over time. The implementation is focused on creating a slot-based system with 100% compliance, that will eliminate the stops incurred by each driver [14]. This solution will attempt to alleviate congestion [2] by applying traffic shaping measures to maintain traffic speeds, consistent trip times and queuing for light and heavy vehicle traffic volumes. This solution attempts to test the hypothesis that journey times can be more accurately predicted, due to more consistent average journey times and overall traffic flow.

The design and implementation choices led to the successful build of the slot-based driver guidance system. The central guidance system was developed in Java and had working features allowing external communication with both OpenStreetMap's online database and also the simulation software PTV Vissim. This communication enabled real world roads to be replicated in the guidance system, however the limitations of the simulation software meant separate test road networks had to be designed and used when testing the simulations. Parking lots were designed in Vissim using diagonal spaces as previously discussed in 2.2.4, due to the diagonal spaces requiring less lane width and allowing quicker manoeuvres for moving in and out of the spaces. Aspects of the intelligent parking lot management system 2.2.5 were also used to incorporate parking space reservations and routing within the parking lots to all available parking spaces. The implemented system successfully provisions slots for each vehicle and manages their position on route to the destination, with no collisions between cars at intersections.

3.1 Requirements

- The system should allow road networks containing multiple roads and complex intersections with many incoming and outgoing roads.
- Each road may be constructed of multiple lanes.
- Information should be sent to the system from the vehicles.
- Information from the vehicles should be received from the system.
- Vehicles should maintain a safe distance from one another and no vehicles should collide.
- Parking lots should be capable of holding 100+ spaces.

- Parking spaces should be booked in advance.
- Traffic should utilise the entire road network, being re-routed as necessary.
- Statistics should be gathered for simulations (slot-based simulations and control cases).

There were many challenges during the planning and development of this system; the majority of these challenges came from connecting to, and controlling, the simulation software. These challenges forced compromises to be made on the original design, to allow for a basic guidance system to be developed within the designated time frame.

The slot-based system is complex. The build of this system had to be planned very carefully. It was important to identify the MVP components of the entire system. These components were either external standalone features of the system, functioning outside of the Java implementation, or these components were internal features within the Java implementation. Determining the requirements 3.1 for the system were crucial to then be able to identify the MVP components. The MVP components of the system included: a road management system for managing all roads in the external road network surrounding the event and a parking lot management system for managing all roads and parking spaces within each parking lot of the event.

3.2 Restrictions

The project had to be managed, so that the intended system could be developed within the short time frame. In order to accomplish the build and testing of this system certain restrictions had to be placed in regards to the road networks, vehicles, compliance levels:

- All vehicles have the same capabilities, in regards to speed, acceleration, technology etc.
- All vehicles are the same size and type.
- All road segments have a maximum of only one lane in each direction.
- All Vissim Connectors only have room for one slot (one vehicle).
- All Vissim links end before an intersection (there is no overlap between a connector and its ToLinks or Fromlinks)
- All intersection zones are composed of overlapping connectors
- All incoming links to an intersection zone are connected to all other outgoing links except for the outgoing link in parallel.
- There are no turning restrictions at intersections, except for u-turns.
- There is only one type of intersection zone. All intersections with incoming and out-going links are considered to be the same type of intersection zone, there is no type difference between "forks" and "T junctions".
- Intersections do not allow multiple vehicles access at once.

- All routes must end at a link with no other further connections.
- There are no security and privacy concerns in this project, however, they will be briefly discussed.
- Parking lots have diagonal spaces.
- Parking lots use a one-way system.
- All vehicles in the system are 100% compliant with the system and immediately follow all guidance given to them by the system.

A selection of restrictions were added to this list as the project developed. This was mainly due to the limitations of the simulation software that could only be discovered by attempting to implement certain features (e.g OSM maps).

3.3 High-Level Overview

The MVP components of the system are the shown in Fig.3.1. The Driver guidance system, developed in Java communicates with the simulation software through the COM interface. The driver guidance system requests network and vehicle specific information from the simulation every tenth of a second. The parking lot routes towards the parking spaces and parking lot exit routes from all parking spaces are exported from Vissim to CSV files. These CSV files are then imported by the driver guidance system to be assigned to the vehicles attending the event. The parking routes had to be exported from Vissim as CSV files, as there were no functions for requesting the parking routes from Vissim through the COM API.

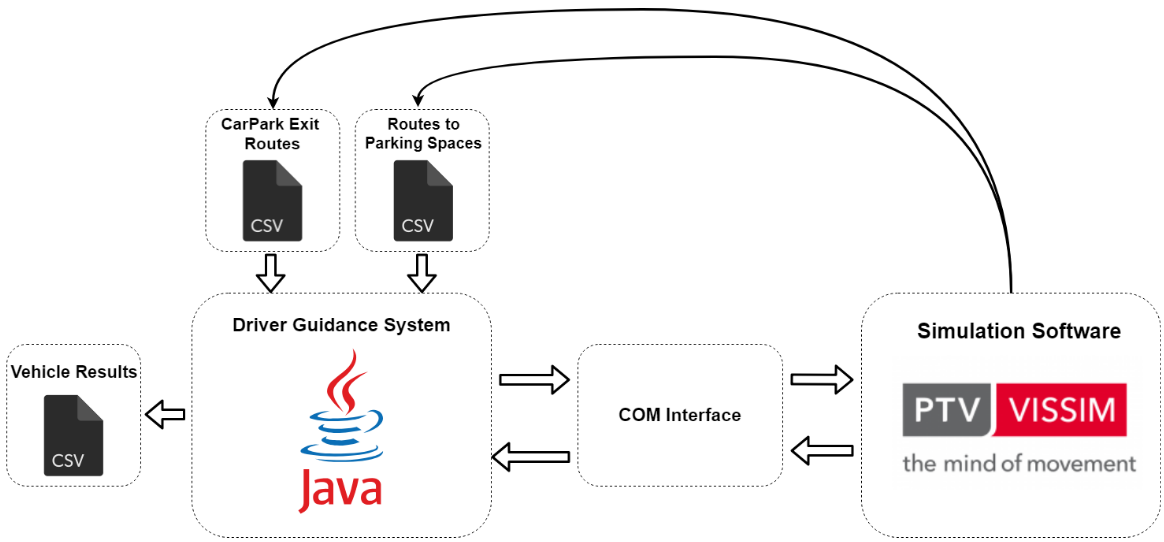

Figure 3.1: High-level system overview

System Architecture Development

The requirements gathering process, provided a good foundation for composing an architecture diagram of the ideal system. After many iterations, the proposed architecture is shown in Fig.3.2.

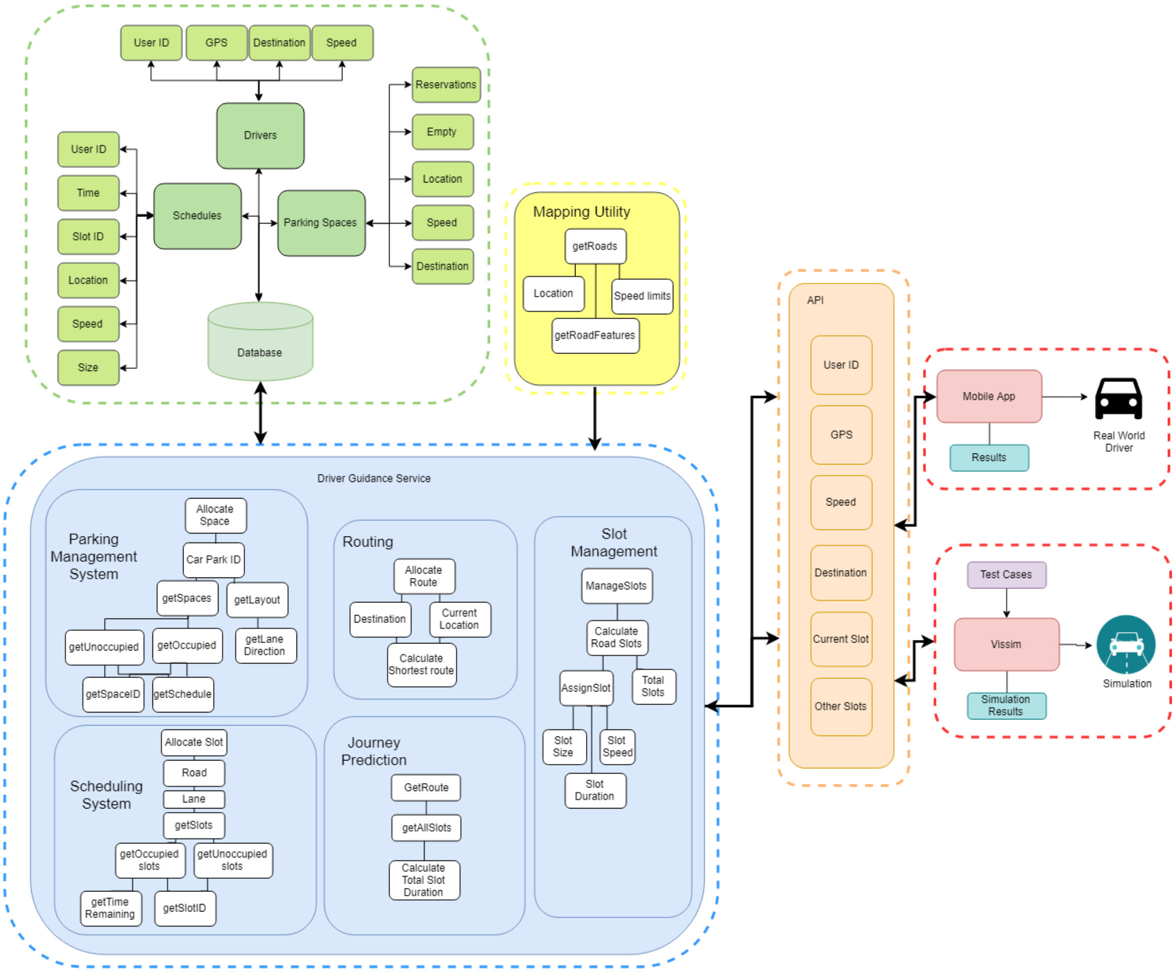

Figure 3.2: Proposed system architecture.

The proposed ideal architecture Fig.3.2 included a mapping utility, which is used to extract mapping information from OpenStreetMap (OSM) and this data is then used by the driver guidance system to recreate real world road networks. A database entity was proposed to store all the schedules, driver information and parking information of the system in a well organised and efficient structure. The proposed slot-based driver guidance system has five key components: the parking management system, the scheduling system, the slot management system, the routing component and a system for predicting the average journey times. To connect the guidance system to the mobile application and the simulation software, an API is required. The API allows for the sending and receiving of vehicle information in real time.

Due to the challenges encountered over the course of the project and time constraints, re-adjustments to the previously planned deliverables had to be made. This meant reducing the system down to a more basic architecture which can be seen in Fig.3.3. The mobile application

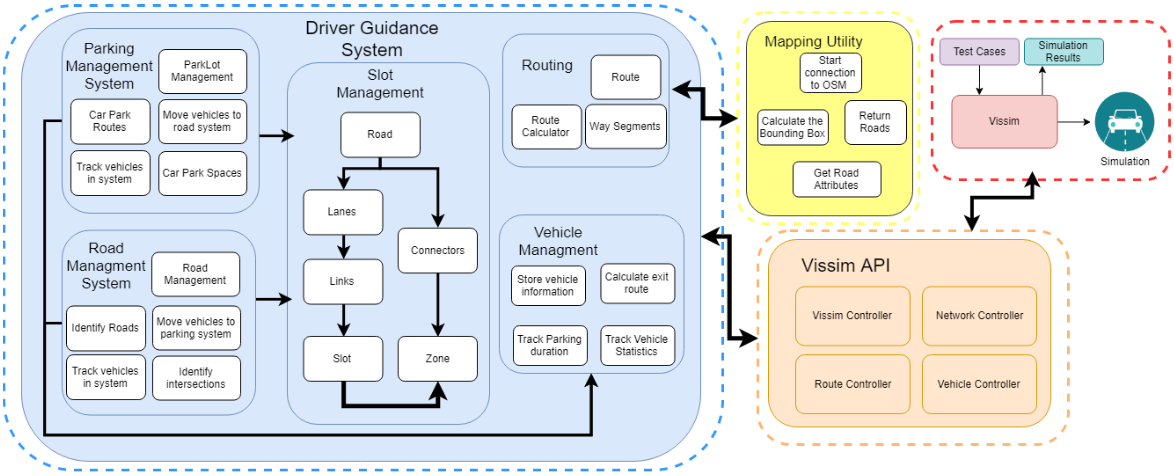

Figure 3.3: Final system architecture.

was removed and the database became redundant during development. The mapping utility, while not used in the final simulation, still held enough successful functionality to include in the final architecture as the only limitation was importing the OSM maps into Vissim.

3.4 Driver Guidance System Entities

To fully understand the system, insight is required into the design of the main entities of the guidance system before the system's functionality can be discussed. This provides the necessary background information regarding the composition of the main entities that shape this slot-based driver guidance system. Relationships between the entities are briefly covered before a more in-depth discussion in later sections.

3.4.1 Vehicle

Vehicles are one of the two most important objects in the system. The vehicles are an abstraction of real world vehicles and represent the 'users' - without them there is no need to provide guidance. Each vehicle is a unique entity, which is created by the user or in this case the simulation software. The vehicle symbolises a driver on the road using the system. The main attributes that define a vehicle are:

- ID
  - This a unique reference to each vehicle, to easily identify them individually.
- Origin
  - The location at which the vehicle joined the road network controlled by the slot-based system.
- Destination

    - The desired final location of the vehicle. This is either a parking space within the parking lot of the event or a specific destination away from the event. On first entry to the system, if a vehicle provides the event as the desired destination, this destination is updated to a parking space, if one is available.

- Current Position
    - This is the vehicle's current position in the road network or parking lot. This is updated on a regular basis through real-time communication with the vehicle.
- Speed
    - This is the current speed of the vehicle at a specific period in time. This is used to provide guidance to the vehicle on maintaining its position within the assigned slot.
- Is Parking
    - The system needs to know if the vehicle is completing a parking manoeuvre or not, as additional space and time needs to be accounted for to allow the car to successfully move in and out of the parking space.
- Allocated Parking Space
    - If the occupants of the vehicle are attending the event and the vehicle's destination is the event, an available parking space should be assigned to the vehicle.
- Stay Time
    - If the vehicle intends to stay in a parking space, the stay duration must be declared.
- Route
    - The current route the vehicle is instructed to follow.
- Start Time
    - The simulation time at which the vehicle first entered the road network controlled by the slot-based system.
- End Time
    - The simulation time at which the vehicle left the road network controlled by the slot-based system.

3.4.2 Slot

Slots are the second most important component in the system, as without slots the concept of a slot-based driver guidance system could not be implemented. Slots are specific regions spread over the width of a lane and long enough to match the length of a vehicle. However, by definition, slots are created to be longer than the vehicle that has been assigned to it. This is to account for potential delay or error by the vehicle in maintaining a central position within the slot. A safety distance is added between each slot. This safety distance is proportionate to the speed at which vehicles travel at and the larger distances required to slow down or stop. The positions of each slot are tracked via the road management system and the position is the distance in metres from the start of the road segment.

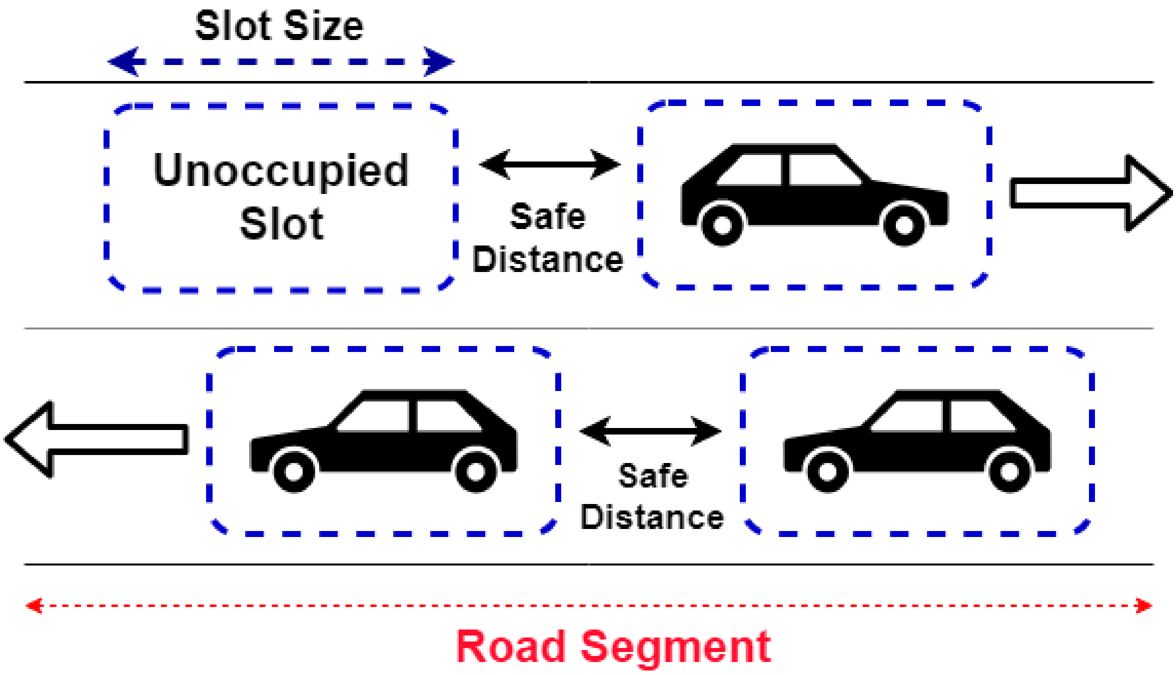

Figure 3.4: Example of slots on a road.

The main attributes that define a slot are:

- ID
  - This is the unique identifier for each slot.
- Road ID
  - This is the reference for the road to which the slot is currently assigned.
- Lane
  - This is the lane number on the road to which the slot is currently assigned.
- Size
  - Each slot is constructed to be of a specific size; big enough to contain the vehicle by which it is occupied.
- Occupied By
  - The vehicle currently occupying the slot.
- Reserved By
  - If no vehicle currently occupies the slot, a reservation can be made to prevent multiple vehicles from attempting to occupy the slot at the same time. This acts as a form of lock or mutex, similar to those used in concurrency control.

3.4.3 Road Segment

In this system, a road is a specific polyline within the road network which is positioned between two intersections. Vehicles cannot leave a given road segment before the end of that road segment is reached.

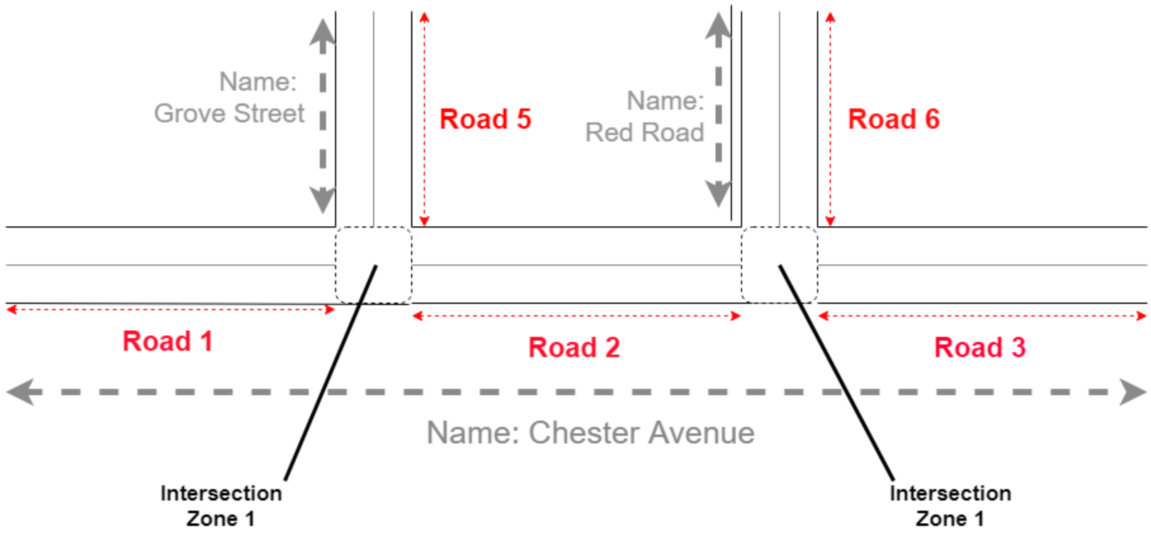

Figure 3.5: Example road segments in Vissim.

In Fig.3.5, the road architecture is shown. While in the physical world, a road usually covers a long distance with multiple turns every few meters. The slot-based system design requires roads to be divided into segments. This division is required for both the simulation and the OpenStreetMaps as there is a need to separate the intersections from the straight roads. The intersections require more complex management than the road segments. A road segment is defined as being the distance between each pair of subsequent intersections. Initially the road segment entity was created for the road networks imported from OpenStreetMaps, however, the link and connector style design that is used within Vissim meant that this road segment design was also required for Vissim simulation networks. The main components of each road segment are:

- ID
  - Unique identifier for each road.
- Name
  - The name of a given road as per the road naming system in Ireland.
- Length
  - The length of a road segment, from start to end.
- Number of Lanes
  - The number of lanes that the road has of the same direction.
- Speed Limit

- The maximum speed limit for the given road.

- Nodes
  - All nodes from the OSM map data, which are used to create this road.
- Requests
  - A list containing all current access requests from vehicles that are attempting to enter a given road controlled by the slot-based guidance system.

3.4.4 Lane

Each road segment, contains one or more lanes. A lane is a specific stretch of road, used to separate lines of traffic. Lanes are intended to divide traffic based on direction or speed. This system limits each road segment to containing one lane of one direction or two lanes of two different directions. The main components of a lane are:

- Lane Number
  - The unique identifier for each lane of a given road.
- Road ID
  - The reference for the road to which the lane belongs.
- Lane Length
  - The length of the given lane, as it could be shorter than the entire road (e.g. it merges into another lane).
- Slot Positions
  - The position of a slot in a given lane, the distance the slot is from the start of the lane.
- Slots
  - All slots currently travelling on the given lane.
- Lane Waiting
  - This is used to determine whether or not the lane is held up by other vehicles, either awaiting access or due to an unexpected event.
- Minimum Slot Gap
  - Used to maintain a safe distance between slots travelling in the lane.

3.4.5 Route

A route is defined as being a path that any vehicle can take, providing both the route and vehicle have the same points of origin in the slot-controlled road network. A route consists of multiple roads and may contain multiple intersections.

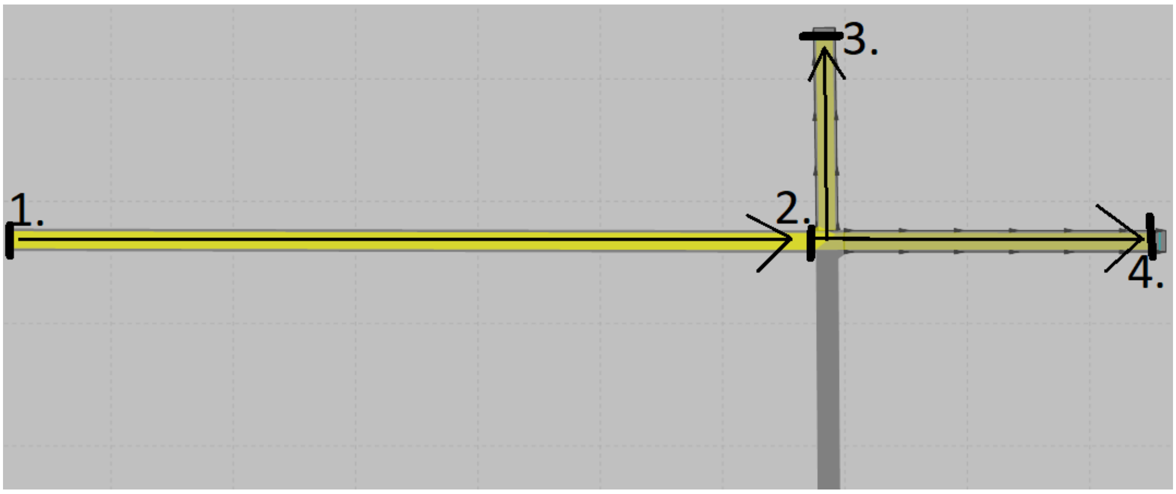

Figure 3.6: Example of routes in PTV Vissim.

In Fig.3.6 using a one-way road with an intersection of 3 different turns, a route is shown as the arrows on each link and connector. The origin point in this case is at the line labelled **1**. The arrows from line **2** to lines **3** or **4** are variations of the route from the same point of origin (line **1**). The lines **3** and **4** indicate the destination point of a route. The slot-based driver guidance system considers each variation to be a unique route, for simplicity purposes. The main components of a route are:

- ID
  - Unique identifier for a route.
- Origin
  - The point from which any vehicle on this route first joined the slot-controlled road network.
- Destination
  - The final destination point where the route ends.
- Link Sequence
  - A list of all the roads (links) and intersections (connectors) that a vehicle must travel along in order to complete that route.

3.4.6 Parking Lot

In this system, a parking lot is an area containing many parking spaces. Vehicles will only be routed towards a parking lot if they intend to park and attend the event. The parking lot has been designed in Vissim to be a one-way system with several aisles of parking spaces in a diagonal parking space layout. The parking lot has been design as a one-way system to reduce the complexity of intersections within the road network of the parking lot. As previously mentioned, the diagonal spaces are used to increase vehicle change over efficiency by reducing the difficulty of completing a parking manoeuvre.

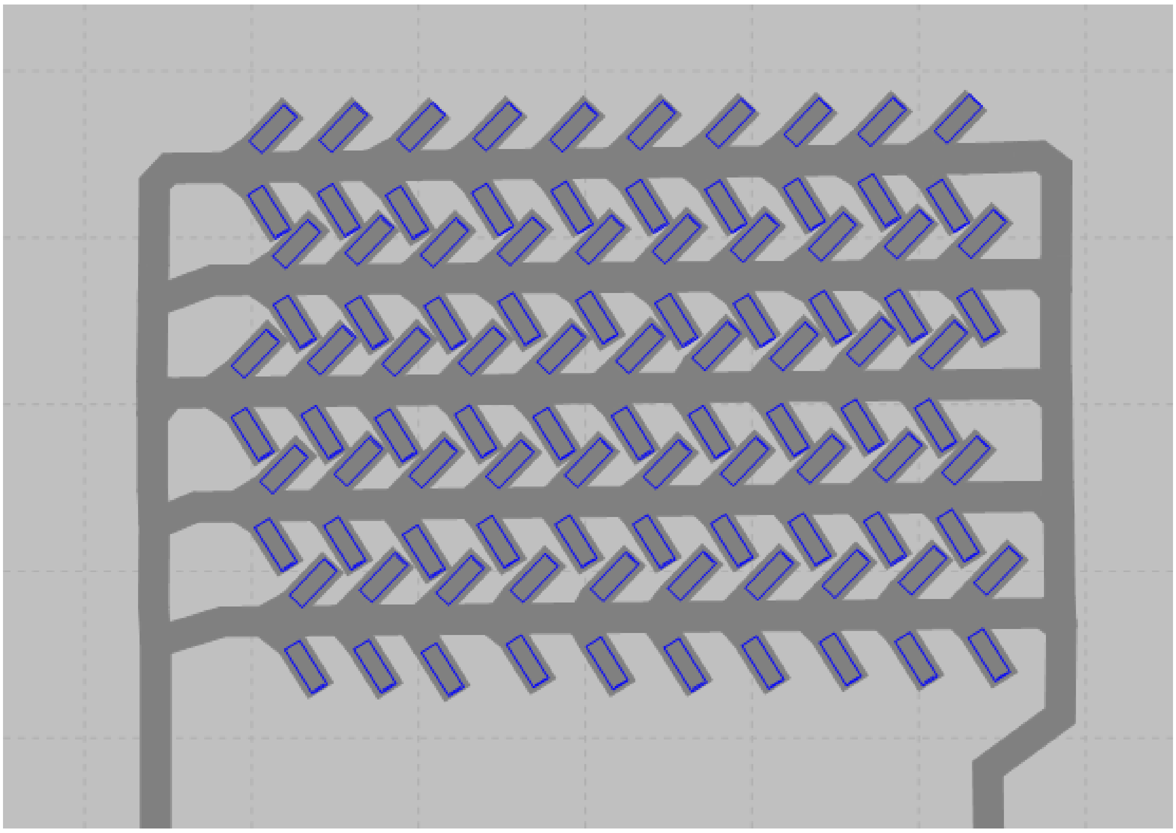

Figure 3.7: Parking lot design in Vissim.

Fig.3.7 shows the design of the parking lot which was used when testing this system. The grey polylines are the combination of links and connectors to form the roads and intersections throughout the parking lot. The blue rectangular squares positioned on short grey links are the parking spaces.

- ID
  - The unique identifier for the parking lot.
- Spaces
  - All parking space objects in the parking lot.
- Routes
  - Routes to each parking space in the parking lot.
- Entrance
  - The road segment that holds the entrance to the parking lot road network.
- Exit
  - The road segment that holds the exit from the parking lot road network.

- Free Spaces
  - The current number of unoccupied parking spaces.
- Occupied Spaces
  - The current number of occupied parking spaces.

3.4.7 Parking Space

A parking space is similar to the slot mentioned before. However, the parking space does not move. It is in a fixed location on its own unique link, as shown in Fig.3.7.

- ID
  - The unique identifier for the given parking space.
- Link ID
  - The Vissim link / road where the parking space is located.
- Parking Lot ID
  - The reference for the parking lot where the parking space is located.
- Occupied By
  - If the parking space is occupied by a vehicle or not.
- Reserved By
  - If the parking space has been reserved by a vehicle, but not yet occupied by one. This prevents the system from assigning this parking space to other vehicles.

3.4.8 Intersection Zone

An intersection zone is defined as being one or more connectors between two or more road segments (links). However, if multiple connectors overlap or form conflicting paths, these are considered to be part of the same intersection zone rather than individual intersections.

In this system, there is only one **type** of intersection zone. All intersections with different amounts of incoming and out-going links are consider the same **type** of intersection zone, there is no **type** difference between "forks" and "T junctions", which are differentiated in the real world. In Fig.3.8, the intersection zones for both A and B are denoted by the rectangles with dotted lines. Intersection zone A is shown to be have only one incoming link and two out-going links. In the slot-based system, this intersection is considered the same object as zone B, even though B is more complex with multiple incoming and outgoing links on conflicting paths and would be managed differently to zone A in the real world. This system only allows one vehicle access to an intersection zone at any given time, due to the complexity involved in coordinating vehicles on conflicting and non-conflicting paths which is discussed in a later section.

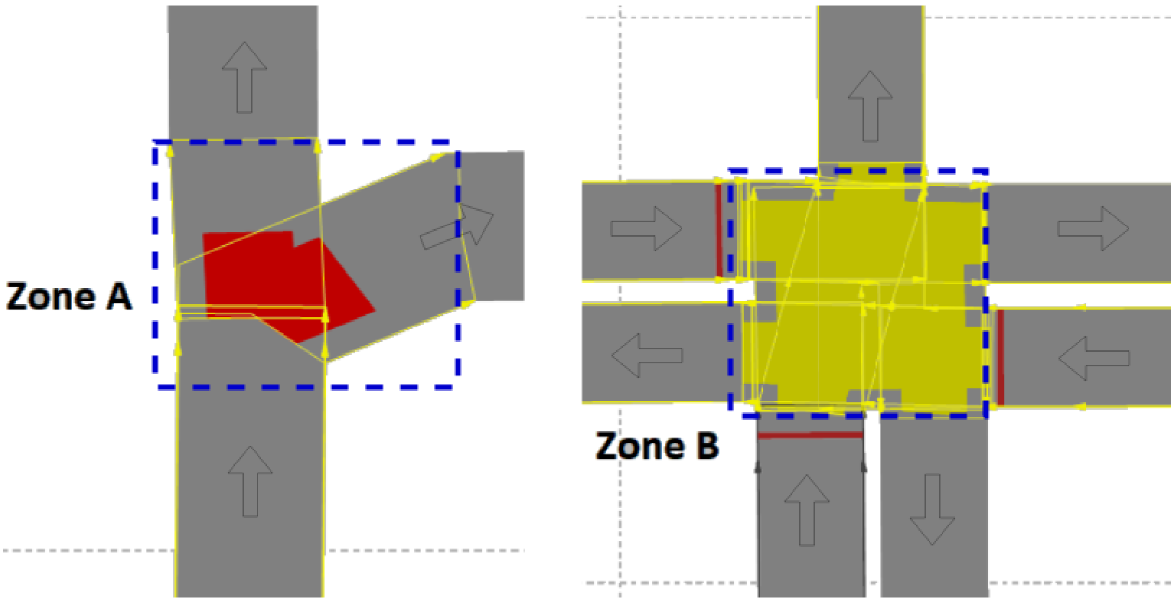

Figure 3.8: Examples of intersection zones as per definition for the slot-based system.

The main components of an intersection are:

- Intersection ID
  - The unique identifier for each intersection zone.
- Links
  - A List of IDs for all incoming and outgoing road segments which create this intersection zone.
- Occupied By
  - If the intersection zone is occupied by a vehicle, the ID of that vehicle is stored.
- Reserved By
  - If a vehicle has reserved the intersection in the future of the simulation, the ID is stored of the vehicle, alongside the future reservation it has made for the intersection zone.

3.4.9 PTV Vissim Communication

To allow for the Java implementation to communicate back and forth between the simulation software, a Vissim controller, network controller, route controller and vehicle controller were created. A controller in this system is an object that contains many functions for making calls to the COM interface and reduces the overall complexity involved in making calls to the Vissim COM interface. Unfortunately, the controllers took longer than expected to implement, as the Vissim COM interface documentation lacked clarity when making API calls through Java. The attribute names used in API calls were slightly different to the attributes used in every other language and could only be found through online research or trial and error.

Vissim Controller

This controller is used to handle the invocation of the connection to the PTV Vissim software and to load one of several types of test road networks. The Vissim Controller is also responsible for starting the simulation when required. This controller is called once at the very start of the simulation to connect to the Vissim software and then continuously during the simulation to run the next time step.

Network Controller

The Network Controller is responsible for retrieving all the information required by the slot-based guidance system, to recreate the road network in the system's own objects. This includes retrieving all links, link attributes (length, speed, name, ID), connectors and connector attributes (FromLink, ToLink, length). This controller is called once at the start of the simulation to import the Vissim road network into the guidance system.

Route Controller

The Route Controller is responsible for retrieving all possible "static routing decisions" from the simulation. A static routing decision in Vissim is a route which is predefined during the creation of the road network. Vehicles generated during a simulation may be assigned to follow that static routing decision, if the point where the vehicle enters the network first is on a link that is part of that route. Routing was intended to be configured by the slot-based guidance system, however due to restrictions on static vehicle routes being read-only attributes in Vissim, a selection of predefined routes were configured in Vissim. This gave the responsibility of routing vehicles entirely to Vissim. Vehicles are assigned routes based on their point of entry and the relative flow of each route variation. The relative flow is an attribute stating the proportion of vehicles that should be assigned a given route. Routing was also intended to be dynamic, however, the COM interface lacked the functionality to retrieve dynamic routing decisions made by Vissim and hence, this had to be static. A way to circumvent this routing shortfall of the COM interface was discovered during the latter stages of the project. This solution was to export the routes from Vissim as CSV files and then using move vehicle functions via the COM interface to override the route assigned to a vehicle by Vissim. Due to time constraints, this dynamic routing solution was only applied to the routing of vehicles in the parking lot.

Vehicle Controller

The Vehicle Controller is responsible for retrieving information about all vehicles in the simulation at a given time. The retrieved information includes: the current route, speed, position, current link/connector and vehicle ID. However, unlike the other controllers, the vehicle controller also sends commands to the simulation, including changing the speed, merging to another link, changing the vehicles position, and adding or removing vehicles. This controller is used continuously throughout the simulation. These continuous calls are to send information to and receive information from each vehicle in the network, at every time step.

3.5 The Slot-based Solution

As discussed in section 1.2, high volumes of vehicles in a small urban environment, accompanied by chaotic driving behaviour negatively effects traffic flow and causes congestion. The goal of this project is to improve the flow of traffic in the vicinity of an event and to increase the accuracy of predicting vehicle journey times. The proposed traffic shaping solution was to allow vehicles to travel in slots for the duration of time that they are in the vicinity of the event. The initial challenge was in architecting a solution to accomplish this. Several design iterations were analysed including using a single data structure to track slots on a coordinate based system and using a system similar to the nodes used by OSM, where additional nodes were created and each node in a network represented a slot. However, these were far too complicated for solving this slot movement problem, due to the complexity of the data structures and the limitations of data available. To further reduce the intricacy of the system, the compliance rate of the drivers in the system is assumed to be 100%. This removes the need to handle on the fly scheduling as unguided drivers enter and move around the event's road network.

The final design is similar to a production line in a factory. Factories utilise conveyor belts to move product across large floor distances. When applied to a road network, each road segment is thought of as a conveyor belt. Road segments move each slot along until the end of the road is reached, where the vehicle is then passed onto the next road in the vehicle's route. This solved the problem of moving slots for road segments. However, in contrast to a conveyor belt in a factory, each product (slot) should be capable of moving at different rates along each belt. Hence the design was altered to allow each slot to move along the conveyor belt data structure in two ways:

- All slots under the control of the conveyor belt (all cars moving at the same rate).
- Or selections of slots allowed to move at different rates (vehicles moving at different speeds).

The main conveyor belt road segment implementation utilises two arrays to keep track of the slot objects on the road and also the position of each slot on that road. This enables the system to adjust the movement of slots as required, whilst maintaining safe distances and slot locations throughout. An array was a suitable data structure as it enforced the restrictions on the number of slots which could be located on a road at any one period of time, similar to the maximum number of vehicles that can fit on a stretch of road while maintaining a safe driving distance. The implementation uses the concept of traffic shaping as discussed in the related work by Marinescu et al. [2], to form traffic into a more manageable pattern. This slot-based solution intended to schedule slots at the start of the vehicle's journey upon entering the controlled road network surrounding the event. Access to every intersection would be pre-booked and the vehicle's slot on approach to each intersection would adjust to allow for it to reach the intersection at its scheduled time. The slot-based system uses look-ahead algorithms to check for all vehicles ahead of the vehicle. If there is no occupied slot ahead of the current slot, the current slot is allowed to drive at the maximum speed for that road. However, if there is an occupied slot ahead of the vehicle, the distance and speed difference between them is calculated. If the slot ahead of the vehicle is travelling at the max speed limit for the current road, the current slot is allowed to drive at maxmimum speed for that road, otherwise calculations are done to control the movement of the current slot to allow it to catch up to the next slot while maintaining the safe distance.

Simple tests within the console and Vissim were completed using this conveyor belt system for one-way, single lane road segments for coordinating the slots. The test cases included slot scheduling and position adjustment within the guidance system and for vehicles within Vissim. The tests were completed for different vehicle volume levels, increasing from a low vehicle input to a heavy vehicle input. These tests successfully scheduled vehicles within slots, adjusting their position and propagating each slot (vehicle) towards the end of the road segment. The driver guidance system did not lose control over each vehicle in the network for the entirety of the tests, also showing a successful 100% driver compliance rate. Slots maintained a safe distance between them. The success of these tests meant work using this solution could continue, the concepts of the coordinate based system and OSM node style solution were dropped. The next iteration of the system was to develop features to handle more complex road networks including the transfer of vehicles across two road segments (intersections) and other networks with bidirectional traffic flow.

The intended design when adding intersections was to have an intersection zone reservation system. The slot based system, would use a more complex look ahead algorithm to check for occupied slots ahead of the current vehicle, but also oncoming intersection zones. The movement of the current slot would be adjusted to allow the slot to reach the intersection at a time when it is available to enter and pass through. At the start of a journey, calculations were done to schedule reservations for each intersection along the new vehicle's assigned route. This would ensure a time would always be available for the vehicle to pass through each intersection without stopping, although they may have to slow or speed up to reach the intersection at the correct reservation time. The scheduling of slots would no longer be primarily down to the slots ahead of the vehicle, but instead scheduled based on the available access times to the intersections along the route.

3.5.1 OSM Mapping Utility

OpenStreetMap was intended to be the source of all road network data. This was to ensure the most realistic network possible on which to test the slot-based system. Hence, the mapping utility was designed to retrieve road networks from the OSM database using the OSM-Commons Java library [28]. The OSM-Commons library enables Java code to query the OpenStreetMap API at api.openstreetmap.org. Initially, an OsmConnection had to be created. Once the connection was successfully established, the API could be continuously queried. OSM-Commons utilises a function called 'getMap', which returns all map elements within a given area. However, a

BoundingBox object needs to be provided as an argument. The BoundingBox object was created from 4 coordinates representing the top, bottom, left and right edges of the box. For the purpose of this project, this was modified to only require one coordinate, which would return all map elements within a 3km radius of the given coordinate. Each element returned is stored for future use. The tags of OSM Ways are filtered and the required information to create Road Segments were extracted including (ID, name, speed limit, OSM Nodes which make up the way and the type of way). The direction of each OSM Way is determined by checking the order of the OSM Nodes that make up the way. Any tags stating if the way is a one-way road or in reverse order are also checked, as this is how direction is stored in OpenStreetMap.

Upon retrieval of the road networks, OSM ways 2.3.2 are converted into road segments. However, the OSM-Commons library lacked functionality for determining the length of ways. Hence, research was completed on how to calculate the length of roads based on latitude and longitude coordinates of each OSM node.

Haversine Formula

The haversine formula calculates the shortest (straight) distance between two given points on a sphere. These two points have longitude and latitude coordinates. When applied to the earth, this is only an approximation due to the earth not being a perfect sphere shape. Hence, an assumption was made for this project whereby the distance of each road used in this system is only an approximation.

$$\mathrm{haversin}(\frac{d}{r}) = \mathrm{haversin}(\theta_1 - \theta_2) + \cos(\theta)\mathrm{haversin}(\lambda_1 - \lambda_2) \quad (1)$$

Where haversin is a reference to the haversine function:

$$\mathrm{hav}(\theta) = \sin^2(\frac{\theta}{2}) = (\frac{1 - \cos\theta}{2}) \quad (2)$$

d = the distance between the two coordinates.

r = the radius of the sphere (in this case the radius of the earth, approx $6,371$km).

ϕ_1, ϕ_2 = the latitude of $coordinate_1$ and the latitude of $coordinate_2$ in radians.

λ_1, λ_2 = the longitude of $coordinate_1$ and the longitude of $coordinate_2$ in radians.

As shown above, the latitude and longitude must first be converted into radians, which are then used in the haversine formula and haversine function. Tests were conducted on the Java implementation of this formula by comparing the distances of results from Google Maps along certain roads. Calculations by Google Maps are also an approximation, however, they were seen as a good base case for this project. While there was minor differences in the results, it was on average less than a 2 metre difference. This was an acceptable difference, as the road network used in this project was intended to be as realistic as possible, it did not need to be exactly correct and the slot-based system was capable of adapting slot provisioning and scheduling to roads of any size.

3.5.2 Connecting Road Segments

The OSM data was used to form road segments. Multiple OSM Ways (a sequential series of OSM nodes, creating a road) may share the same node, hence creating an intersection. Therefore, all nodes where an intersection was created, were tracked in a list. After the entire road network and required OSM Way-specific information was imported from OSM, road segments were created using the remaining nodes in each OSM Way that were not intersection nodes. The intersection nodes are therefore the start and end nodes for road segments.

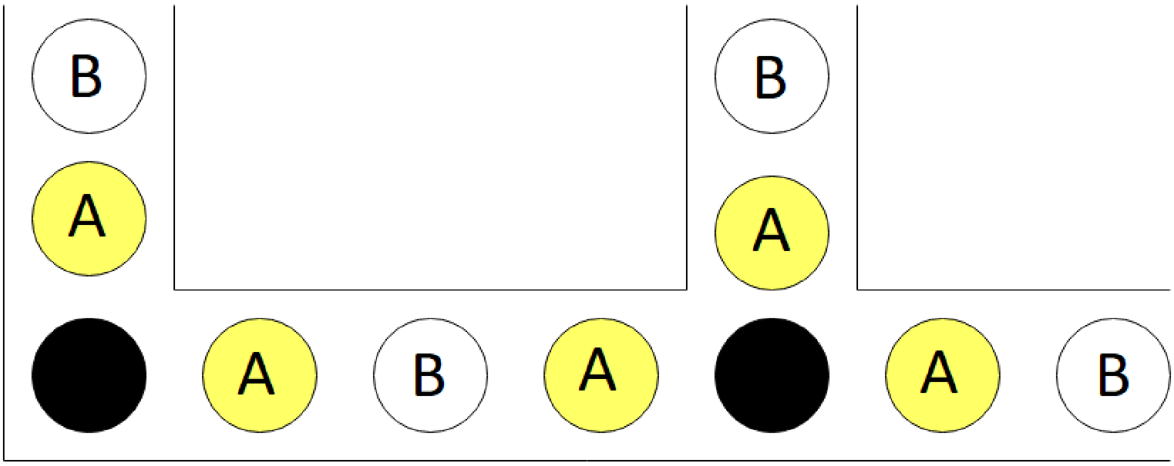

Figure 3.9: OSM Network of interconnected Ways.

In Fig.3.9, Black circles are intersection nodes and circles labelled A are the nodes closest to the intersection node. The layout of the intersection was determined from the coordinates of the closest node to each intersection node. Intersection objects were created for each node to keep track of whether intersections were occupied or unoccupied.

Unfortunately, this entire OSM mapping system could not be used for the OSM road networks imported into PTV Vissim. This was because the original OSM map data was difficult to import to the simulation software, without manually creating every link and connector to be the exact length and position replicating the OSM map, which is extremely time consuming. Previous research on determining which simulator was the most suitable for this project found that another software offering from the PTV group called Vissum had a built-in feature for importing OSM Maps. This Vissum import had features to export the OSM map as a .ANM file which could be imported to Vissim. However, when this was tested, the imported .ANM network was far too complex and different to the original OSM data retrieved from the OSM database. Due to pressing time constraints of the project, the decision was made to move onto implementing custom road networks within the simulation software instead of using OSM real world maps.

3.5.3 The Intersection Zone

The decision to use custom-made Vissim networks instead of OSM maps, led to another challenge in determining where intersections were formed in Vissim. The link and connector design used by Vissim meant that only the potential intersections were known, as each connector was definitely part of an intersection. However, determining which connector overlapped other connectors to form an intersection, from my research, could not be identified. The useful information relating to each connector was limited to the *from* and *to* links.

The solution to finding the overlapping connectors (intersections) in Vissim is to use the basics of set theory and a key-pair data structure, to form an 'intersection zone':

1. Each unique FromLink is checked for all corresponding ToLinks. A set is then generated for all of these corresponding ToLinks, with the FromLink as the key for the set. This new set is the connectedSet.
2. Multiple connectedSets are generated for each unique FromLink.
3. Every connectedSet is compared to all other connectedSets to find sets with matching elements (this is done using the intersection function within Java).
4. If a matching element is found for two sets, those two connectedSets are combined, replacing the two original connectedSets. A list is formed of all keys (FromLinks) of the two original connectedSets.
5. This happens repeatedly until no new combined sets are formed. The lists which were created containing all keys and respective connectedSets are now the links which form an 'intersection zone'.
6. For this method to work, **all incoming links must be connected to all out-going links, and no turning restrictions should apply**. This is therefore stated as another assumption for this project.

Example:

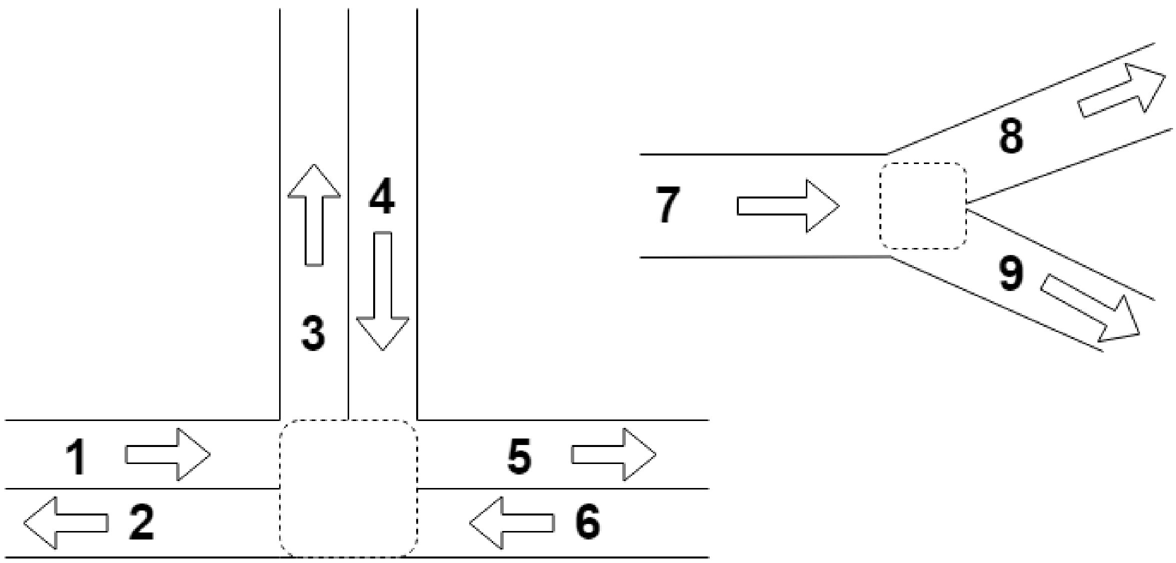

Figure 3.10: Example of two intersections in Vissim

ConnectorID	FromLink	ToLink
10001	1	5
10002	1	3
10003	4	2
10004	4	5
10005	6	4
10006	6	2
10007	7	8
10008	7	9

=>

From Key	ToLinks Set
1	{3, 5}
4	{2, 5}
6	{2, 3}
7	{8, 9}

	Intersection	New Connected Set
$1 \cap 4$	{5}	{2,3,5}
$1 \cap 6$	{3}	{2,3,5}
$1 \cap 7$	{}	{}
$4 \cap 1$	{5}	{2,3,5}
$4 \cap 6$	{2}	{2,3,5}
$4 \cap 7$	{}	{}
$6 \cap 1$	{3}	{2,3,5}
$6 \cap 4$	{2}	{2,3,5}
$6 \cap 7$	{}	{}
$7 \cap 1$	{}	{}
$7 \cap 4$	{}	{}
$7 \cap 6$	{}	{}

Any sets with no matching elements with any other set, are then added as ConnectedSets. In this case "From Key" 7:

FromKeys	ConnectedSets
{1,4}	{2,3,5}
{1,6}	{2,3,5}
{4,1}	{2,3,5}
{4,6}	{2,3,5}
{6,1}	{2,3,5}
{6,4}	{2,3,5}
{7}	{8,9}

All matching ConnectedSets are combined, the final result is the sets of links which form an intersection zone:

FromKeys	ConnectedSets
{1,4,6}:	{2,3,5}
{7}:	{8,9}

3.5.4 Phasing

Phasing is a concept discussed in section 2.2.2 of related work. It is identified as a time period whereby only a certain number of vehicles travelling on non-conflicting paths are allowed to enter and pass through an intersection. This was an important concept to implement as multiple vehicles should be allowed to enter an intersection, on the condition that they will not collide or cross paths. To successfully implement phasing, the layout of an intersection is required. In earlier work, the layout of an intersection was successfully implemented using OSM nodes and their corresponding coordinates. An attempt was made to recreate the layout of the intersection based on the ConnectedSets generated when forming intersection zones from Vissim. However, it was found that multiple variations of neighbouring intersections could be formed. This is shown in Fig.3.11, as both Intersection A and Intersection B are potential layouts of the ConnectedSet and it was impossible to determine which one was the correct layout with the information currently in use. Due to time constraints, a solution could not be found and phasing was chosen not to be implemented. Hence this meant the system could only successfully allow one vehicle to enter an intersection zone at any one period of time. Therefore, this created another assumption for the project that **only one vehicle can access an intersection at any given time**.

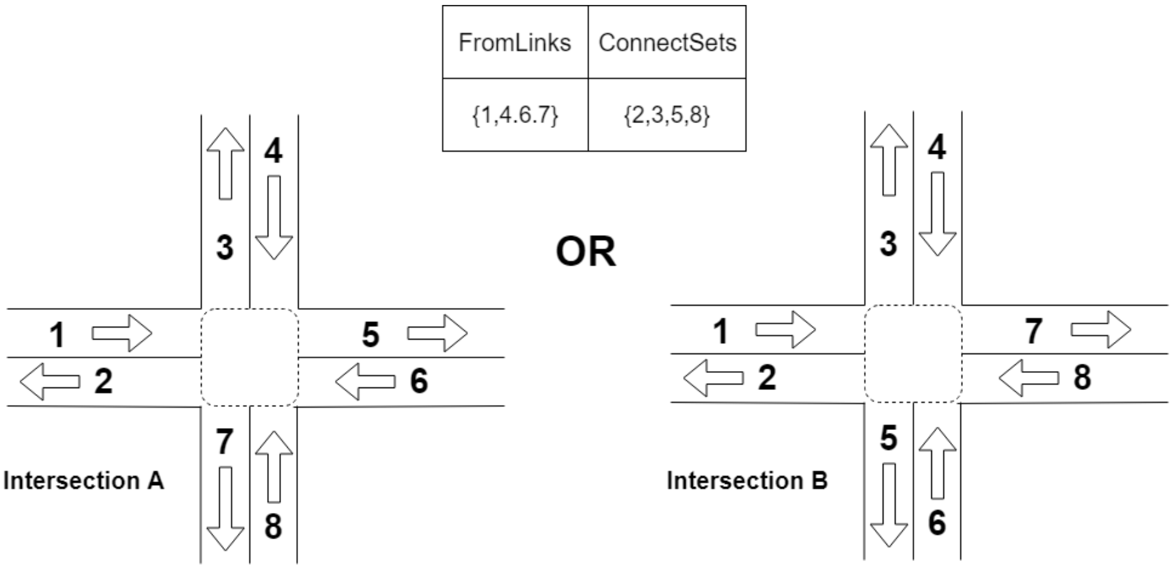

Figure 3.11: Multiple variations of intersections can be made from one connected set

3.5.5 The Parking Lot

In section 2.2.4, Abdullah et al. discusses how to optimise the available space, to create the maximum number of parking spaces using a perpendicular style parking space layout. Yet, [22] provides specific instances where a diagonal layout creates more spaces than the perpendicular layout for a given area. This is due to diagonal spaces requiring less lane width on approach to each space. The diagonal spaces also essentially enforce a one-way system in a parking lot.

For simplicity and improved flow of traffic around the parking lot, the goal was to design a traffic flow optimised system. Events have many attendees and hence require a greater number of parking spaces. However, the flow of traffic was determined to be a more important factor as the goal of the slot-based system is to improve traffic flow and allow journey times to be more accurately predicted. The diagonal layout reduces the time spent manoeuvreing in and out of parking spaces [21], alleviating congestion and reducing the number of slots which are either stationary or moving slowly during these manoeuvres . The one-way system created by the diagonal layout is more beneficial to the flow of traffic in comparison to a two-way system as this reduces the number of conflicting paths and the complexity involved in scheduling slots on conflicting paths within the parking lot.

As shown in section 3.4.6, Fig.3.7 presents the final parking lot design in PTV Vissim. Vehicles enter the parking lot in their respective slots through the left-hand side of the parking lot and exit out the right-hand side. Diagonal spaces are interleaved between aisles and two spaces are positioned opposite each other on every aisle.

The parking lot is created as a separate parking lot network to every other external road network in the slot-based guidance system. Nonetheless, it is composed of a similar road and intersection zone design. Due to the network within a parking lot consisting of multiple turning choices following each other in quick succession, there is a significant increase in intersection zones for the parking lot. Slots are scheduled the same as the external road network, however, additional complexity is added when vehicles request to exit the parking space. To allow a vehicle to exit a parking space, the vehicle must reverse out of the space and manoeuvre so it is in a central position and parallel to the edges of the aisle. To account for this manoeuvre, temporary slots are added to the road segments leading up to the parking space if they are available. If they are unavailable, the slot scheduling system makes a reservation for them to become available at the next best opportunity. All slots leading up to the reserved road segments and newly added temporary slots are rescheduled to account for this change. Once the vehicle has completed the parking space exit manoeuvre, the temporary slots and road segment reservations are removed from the system.

3.5.6 The Handover between Parking Lot and Road Network

The completed parking lot management entity and road management entity were the two major slot control components of the system. The next challenge was to combine each of the entities and ensure a smooth transfer of the control of the vehicles as they are passed from one to the other. The initial scheduling of slots for new vehicles entering the slot-controlled road network was reused for this purpose. As a vehicle entered the road leading up to a parking lot, the vehicle was reassigned a slot within the parking lot entity. A similar process was followed when exiting the parking lot: a slot was assigned to the vehicle. This process replicates the very first slot-based process experienced by the vehicle when it first entered the event's surrounding road network under the direction of the guidance system.

No Availability Re-Routing

When a new vehicle enters the slot-controlled road network, the destination of the vehicle is evaluated. If the destination matches the parking lot of the event, the system searches for a parking space. The parking lot entity tracks all parking spaces determining which ones are occupied, unoccupied or reserved. When a vehicle requests a parking space, the furthest parking space available from the entrance is reserved for the vehicle. The furthest space available is returned as this alleviates any potential congestion closer to the entrance of the parking lot, causing the schedules of slots entering the parking lot to be rescheduled.

If no parking space is available in the parking lot, the vehicle is re-routed out of the test network by the shortest available exit route. The re-routing of vehicles is intended to be to another parking lot with available spaces, or else away from the event as there is no legal parking available.

3.5.7 Intended System vs Implemented System

The original intention of the slot-based system's 'conveyor belt' style design was to schedule slots on entry to the network. Upon entry, the route of a vehicle would be analysed and all intersection zones that the vehicle needed to pass through would be scheduled appropriately. Each road segment's management system would then adjust the rate at which the vehicle arrives at the intersection to meet the reserved time. Due to time constraints and the complexity of this scheduling challenge, this was not completed. However, a successful implementation was completed for scheduling slots on each road, as previously mentioned using the look-ahead algorithms. This implementation successfully propagated slots along each road segment and if the intersection zone was available upon reaching it, the slot would also be propagated through the intersection or delayed to prevent collisions.

A collision prevention system was implemented early on in the project to test the intersection zones in the simulator and prevent multiple vehicles from entering an intersection at once. As a result of the scheduling of intersections for the entire journey of each vehicle being incomplete, the collision prevention system was once more used in its place. In this implementation, road segments are responsible for scheduling all of their currently assigned slots. Therefore, the scheduling of the slots is not based on when each slot's next intersection booking is and the current slot circumstances for the given road segment. The slots are instead scheduled based on the current road segment information (distance between slots, speed of slots etc.) and also when the next intersection will be available. If the intersection will not be available before a vehicle reaches the intersection without travelling below a threshold speed, the vehicle must make a request to obtain access to enter the intersection and then wait until access becomes available. The parking lot management system was also successfully implemented as originally intended with parking space reservations to the furthest available parking spaces from the entrance and slot scheduling accounting for the manoeuvres required by vehicles to drive into and reverse out of the diagonal spaces.

3.6 Security and Privacy Concerns

The slot-based guidance system is a centralised control system, which will utilise each participating driver's built-in location tracking feature of the smartphone device or another solution whereby each vehicle can be tracked on the road. If a smartphone is not the primary form of delivering guidance information to the user, this could be in the form of a heads up display as suggested in [2]. Location tracking on smartphones is carried out by using sources like GPS, Wi-Fi, mobile networks and sensors to help estimate the device's location [29]. There is also an opportunity to use V2I technology and road-side units to track vehicles and collect real time information, as outlined in 2.1. Due to time constraints, the mobile application has not been developed, however, the slot-based control system has been developed to control test road network simulations running in PTV Vissim. The analyses of privacy and security concerns are briefly examined in the context of this mobile app and control system being built in the future.

3.6.1 General Data Protection Regulation

The General Data Protection Regulation (GDPR) has most recently led the charge in ensuring individuals and businesses who deal with personal data are conscious of both the security and privacy aspects of obtaining, using and storing this data. The developers and manufacturers of devices, services and utilities that collect and consume personal data have an ever-growing responsibility to ensure the data is not only secure, but that the data privacy factors are maintained for the entire duration of the data being in their possession. GDPR ensures the rights of users are the number one priority for all parties involved.

The slot-based driver guidance system described previously collects location- and movement-specific personal data from each driver. Therefore, the terms of GDPR must be strictly adhered to. The regulation ensures each data controller is given definitive consent from the user, before any data can be collected or used. The users must be provided with in-depth specifications on how the gathered data will be used, controlled and protected [30]. As the personal data is collected from the driver in real time, the driver must be immediately informed of the specific data being collected. Immediately means at the time the data is obtained. As this system currently does not transfer this data to any other party, the system is not required to inform the driver of any other recipients or any intention to transfer the data outside of the EU. However, if the infrastructure of the system changes and third party tools and services are used, the user must also be informed of the intention to transfer this data to the third party. Information on the retention period of the data and processing which will be carried out by the third party must also be provided to the user. Under the new regulation the user has the right to withdraw any consent given to data controllers, at any period in time, unless contractual obligations state otherwise [31]. Therefore, the data collected must be well organised to allow seamless removal of data at the request of the user without disrupting the system.

The users of the driver guidance system must be given explicit information on how the data is being collected, used and stored. The system will only store information for the duration of the journey. Under the Right to be Forgotten clause, attention must also be given to various other circumstances from which a user may withdraw consent without specifically stating the withdrawal. Any users who stop their journey, become inactive or delete the app, should have their information removed from the system. This is under the terms of GDPR. The data controller must erase all personal data which is no longer needed for its original purpose [31]. In the case of the driver guidance system, the original purpose of the data is solely focused on the driver's current journey travelling to and from the event. No data should be retained after the journey is complete or new data collected for any subsequent journeys without consent being given again.

3.6.2 Security Concerns

There is a strong incentive for manipulating the system for personal gain. As the nature of the system is to facilitate a more cooperative approach to driving to improve the overall flow of traffic. This approach is not guaranteed to lead to shorter journey times. Hence, by maliciously manipulating the system, those malicious users could generate better routes and more beneficial slots for themselves, in turn shortening the journey times for specific users and increasing the journey times of all other road users. This would be similar to the idea of maliciously targeting traffic lights to always be green for your route.

The future implementations of this project could make use of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technology. The latter is currently simulated using Vissim, mimicing vehicle communication with the overlying global slot-based guidance system (V2I). Therefore, if deployed outside of the simulator, there would be many access points to the central control system, which all need to be highly secure, as even vehicles themselves could be potentially used as a hop to gain access to the central slot-based guidance system.

In the paper [32], the researchers outline various potential threats and associated attacks which also apply to this project. These threats include:

- **Passive vs Active**: Passive attacks are those which are used to monitor rather than change the operation or data of the system. However, active attacks could be used to inject packets and modify the slot generation process.
- **Malicious vs Rational**: Malicious attackers intend to disrupt the system in some way, while rational attackers seek to gain benefit from the attack. Vehicles could be routed to crash or a slot could be scheduled to have preferential treatment through intersections and at parking lots.
- **Intentional vs Unintentional**: Intentional acts are threats or attacks which are intended to disrupt the system or achieve a specific outcome. Unintentional attacks are those which are caused by accident, a lack of knowledge or system malfunction. The attacker could take down the system disrupt the flow of traffic in road network.

- **Local vs Extended**: Local attacks target a specific area of the system i.e. jamming vehicle to vehicle communication. On the other hand, extended attacks that cover a large area generally aim to carry out attacks over a longer period of time e.g. tracking or worms.
- **Insider vs Outsider**: Insider attacks are carried out by entities which belong to the slot-based system. This includes any person or user with authenticated access to the system. Outsider attacks are the opposite, and are caused by users who are external and have no authenticated access.

Potential solutions to a selection of these security challenges faced by system include:

- **Authentication**: An efficient and reliable method for authenticating each user must be in place. This will prevent unauthorised access to the system and provide confidence to the vehicle that messages received from the slot-control system are authentic.
- **Integrity**: The integrity of messages should be verified. This will ensure that the data sent from the user or slot central control system has not been modified. In principle, this can be achieved by making use of HMAC (hash based message authentication code) using any cryptographic hash function, one of such is SHA-256.
- **Trace-ability**: Any user or system which abuses the network should be identifiable.
- **Revocation**: If a user or system begins to abuse or attack the network, measures should be in place to revoke the abusive entity's access to the system.

This proposed system will have a direct connection and access to the personal information of users. The collection, processing and storage procedures need to be transparent and the specifications of each must be provided to users. This will ensure GDPR compliance and mitigate against potential legal and financial ramifications of the system. Security concerns are also extremely relevant, as there are multiple access points to the system from the users, vehicles, roadside infrastructure and the central control itself. These all need to be addressed through various solutions including authentication, encryption and message integrity. Outlined above are concerns relating to both the privacy and security aspects of this slot-based cooperative driver guidance system, if it was to be developed into a production system. The current implementation completed in this project lacks the same direct risks, due to it being simulated in a controlled environment.

4 Evaluation

In this section, the performance of the slot-based system is examined. The performance of the system is analysed in comparison to the set of control cases control cases developed in Vissim, controlled by the *Wiedemann 74* car-following model [5]. The evaluation is conducted on a single road network and parking lot design. All restrictions stated in chapter 3.2 are applied to all slot-based test cases. The modelled road network specifically targetted the entrance and exit of a parking lot, the approach roads to the parking lot and within the parking lot itself. An event attendance ratio of 25% was tested for each simulation. The evaluation of the slot-based cooperative guidance system's performance is based on the following metrics:

- Average journey times for a selection routes
- Average number of vehicles passing through the network
- Observed congestion levels: The apparent build up of traffic on each road segment.

4.1 Testing of the Slot-Based System

To test the performance of the slot-based driver guidance system, the test cases were all run using the same volumes of traffic per hour and under the same simulation seed. This simulation seed *(42)*, ensured the vehicle input generation was the same across all simulation runs for each test variation. Vissim has built in metric evaluators, including vehicle count and average vehicle travel times. Three variations of the single test network were examined: two control cases and the slot-based system.

The two control test cases were attempts at replicating real-world scenarios. The first control case modelled had operational traffic lights (signalling) at the intersections. The second control case had a round robin style system (priority rules) at the intersections, with yielding vehicles. The slot-based system controlled the same road network and vehicle inputs with a 100% driver compliance rate. This was to showcase whether any benefits were gained from implementing this slot-based system at all.

4.1.1 Testing Metrics

- Vehicle journey time measurements

- Vehicle Travel Times are calculated by using the Vehicle Travel Time network object within Vissim. Each vehicle travel time object requires a start point and destination point. The travel time is calculated by the time taken for the vehicle to travel from the start section to the destination section. Vissim calculates an average travel time for vehicles travelling on this route alongside the total number of vehicles to have travelled from the start section to the destination section.

- Vehicles passing through the network.
  - Vissim has a feature for tracking all the vehicles currently in the network (vehicle throughput). As a new vehicle enters the network, it is added to the list. If a vehicle leaves the network, it is removed from the list. Vehicles that are generated are given IDs increasing from one. Hence, the total number of vehicles that have passed through the network at any given time can be retrieved by checking the latest vehicle ID to have entered the network and subtracting the count of vehicles which are currently in the network and have yet to leave. This measures the rate at which vehicles consistently flow through the system, the less vehicles passing through the network indicate higher congestion levels, due to vehicles taking longer to pass through.
- Observed congestion levels
  - This metric is a visual metric which showcases when specific routes become heavily congested during a simulation. This visual metric is the apparent build up of vehicles on the road segments. This can be seen by a large increase in the number of vehicles moving slowly or remaining stationary on each road segment.

Signalling Control Case

Within PTV Vissim, the concept of signal controls are used to implement traffic lights. These signal controls follow the repetitive pattern of green, amber and red. Vehicles moving towards a signal control unit detect the status of the next signal control unit and adjust their speed accordingly. The signal control units for the test network are slightly optimised. After multiple simulation runs, it was found that specific roads experienced higher congestion levels than others. This was caused by one signal control unit remaining green, while no vehicles were passing through this green light and all other signal control units at the intersection were preventing vehicles from accessing the intersections with queues of vehicles forming. This inefficiency was reduced by setting the signal times to reflect the road priority and alleviating the queues. This simulation not only simulated traffic lights, but also stewards at an event, directing platoons of vehicles to enter and pass through the intersection at any given time. Signal controlled intersections had a distinct advantage over the slot-based approach, as multiple vehicles were allowed access an intersections at any given time.

Priority Rules Control Case

Another feature within Vissim for modeling right-of-way without signal controllers at intersections is the use of priority rules. Priority rules are used in circumstances whereby vehicles on conflicting links are required to consider each other. They are used to ensure choreographed, free flowing traffic at a non-signalised intersections. The use of priority rules in the test event network implements a round robin style control system. It operates on a first come, first served basis. However, vehicles slow down and yield if conflicts occur before they are given access to the intersection by the car-following model. This control case also simulated stewards at an event directing traffic using a round robin style approach to direct traffic through an intersection. The benefit of the priority rules strategy is that if an intersection is available the vehicle is given access to pass through, it does not have to wait for the signal controller to go green.

Slot-Based System

The slot-based driver guidance system had full control over the movement of each vehicle within the test network. The vehicle input generation and initial routing assignment were configured by Vissim. However, once the vehicle entered the network it was assigned a slot and re-routed to an available parking space, if the vehicle was attending the event. The expected improvements from the slot-based approach over the control cases were predicted to be at the intersections, with less delays and vehicles coordinating in a improved free flowing manner; each vehicle was provided with specific guidance on when they would be granted access to the intersection. Improvements were also expected at the entry to parking lots, due to vehicles being routed away to the furthest parking space available. The final improvements were expected upon the parking lot reaching capacity, as vehicles in the control cases would be still routed towards the full parking lots. However, vehicles controlled by the slot system would be re-routed from the very beginning reducing the congestion at roads leading up to the event and also the time wasted driving towards the full parking lot.

4.2 Results

The test cases were conducted over the duration of 10 simulation runs, which is similar to that of [2]. The simulations were run at four different driver volume levels; the same vehicle input ratio was maintained for each link across all volume levels. This ratio was 4:1:2:1 for incoming links, numbered 1, 2, 3, 4 in Fig.4.1. The driver volume levels that were tested were 1000, 2000, 3000 and 4000 vehicles per hour. The combined event-attendance ratio from all routes was 25% of the driver volume levels.

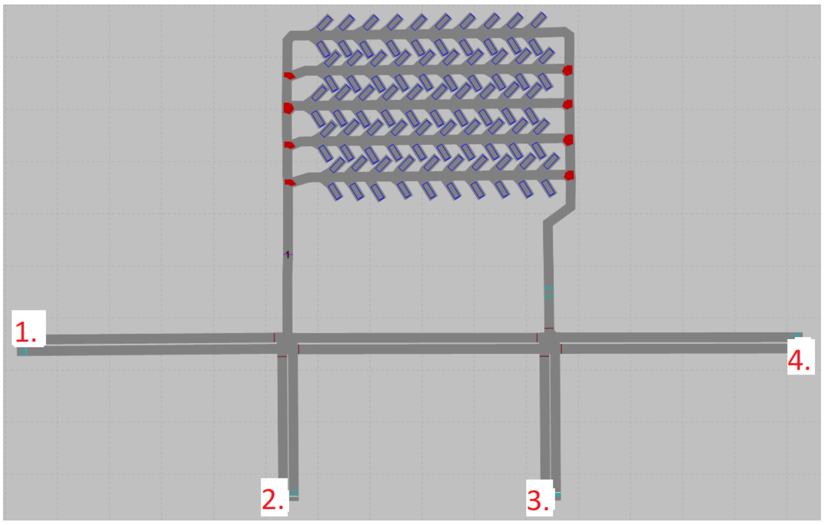

Figure 4.1: The test road network

Average Journey Times

The journey times of vehicles were evaluated over three distances. As the specific tests cases were focused on the entrance and exit of the parking lot, the differences between the journey times are not very large. The longest distance examined was 267m, the middle distance was 224m and the shortest distance was 137m. These variations of distances were specifically chosen as they provided 3 unique scenarios to analyse including travelling through one intersection on a non-conflicting path, travelling through two intersections on both conflicting and non-conflicting paths, and also driving through two intersections on two conflicting paths. A non conflicting path is defined as one whereby the vehicle does not have to pass through the intersection (e.g. taking a left hand turn). A conflicting path is one where multiple paths must be crossed to access the other side of the intersection.

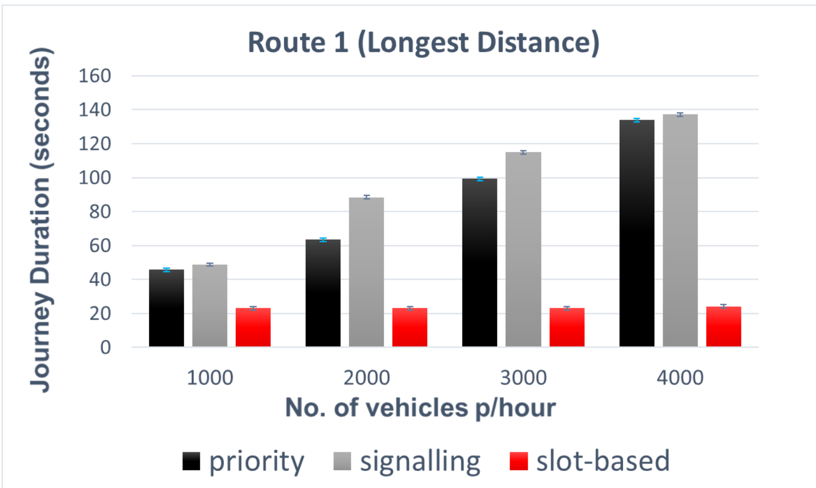

Figure 4.2: Average travel times - Route 1

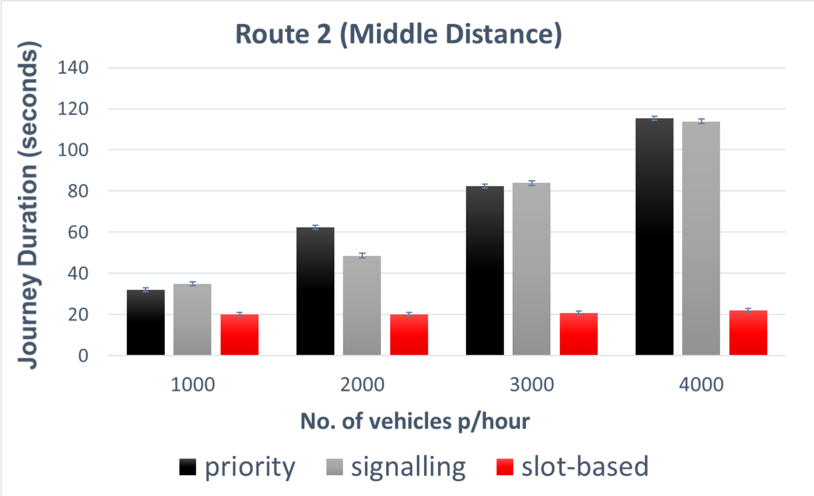

Figure 4.3: Average travel times - Route 2

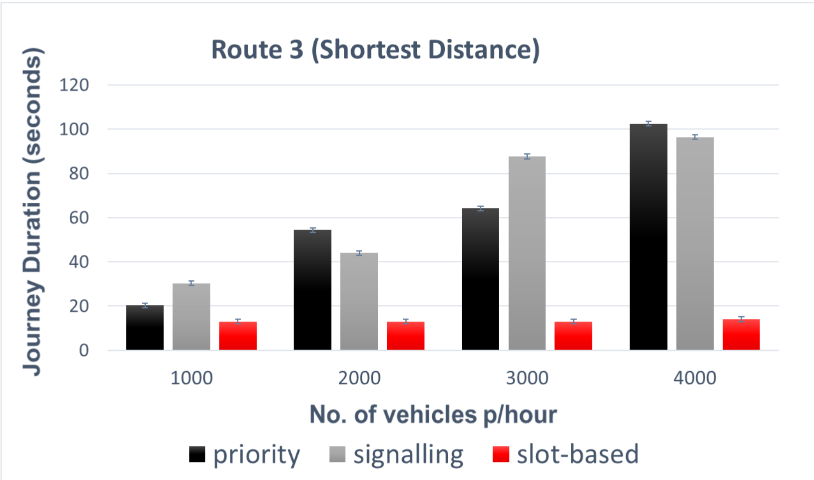

Figure 4.4: Average travel times - Route 3

In all three route variations, there is an improvement for the slot-based driver model. The two control cases suffer from delays in either stopping at a status red signal controller, or slowing down to check for right-of-way to the intersection. At lower volume levels, the journey times are similar for all three test cases. However, as the volume levels increase, so does the difference between the journey times. This is caused by the high congestion levels where the slot-based system is not used, as vehicles remain stationary or travel at a reduced speed for much longer. This difference remains consistent across all three of the routes tested, however, there are slight differences in the performance between the priority rules and signal controllers at different distances and volume levels, with both alternating between which of them has the best performance.

Average No. of Vehicles Passing Through

The vehicle throughput can be seen in Fig.4.5. Both the control cases had a near linear increase in vehicle throughput between 1000 and 2000 vehicle volume levels. However, upon reaching 3000 and 4000 vehicles per hour, this began to plateau, as it seems a threshold was being reached, caused by the congestion in the network and new vehicles being unable to enter the network. The slot-based system, continued on a near linear increase for all four volume levels. The error rate for each simulation run was +/- <1 vehicle. This was negligible and could not be shown on the graph.

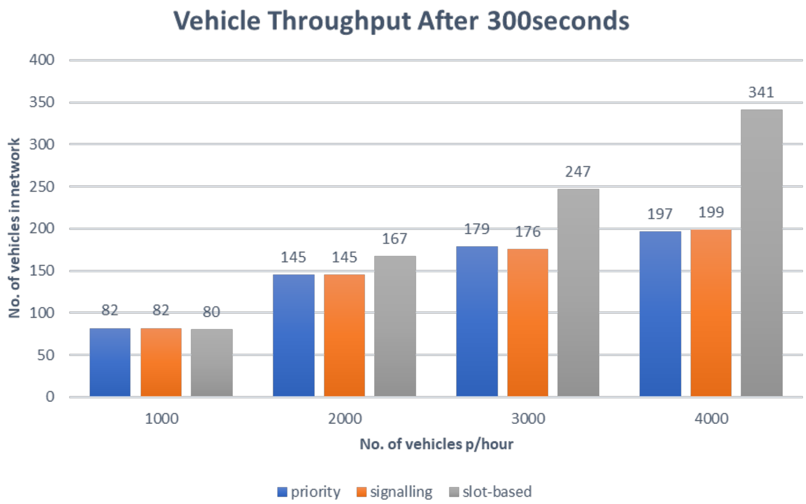

Figure 4.5: Vehicle that have passed through the network after 300s

Congestion Levels

As previously mentioned, congestion levels were examined as a visual metric. Fig.4.6, Fig.4.7 and 4.8 are all snapshots of each simulation after running for 114 seconds.

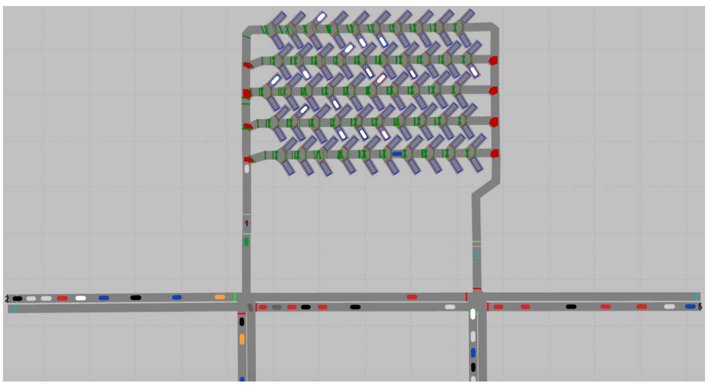

Figure 4.6: Congestion levels: Signal controlled simulation at 114s

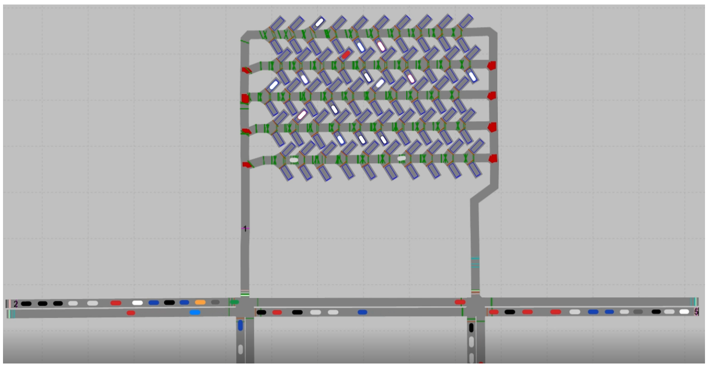

Figure 4.7: Congestion levels: Priority controlled simulation at 114s

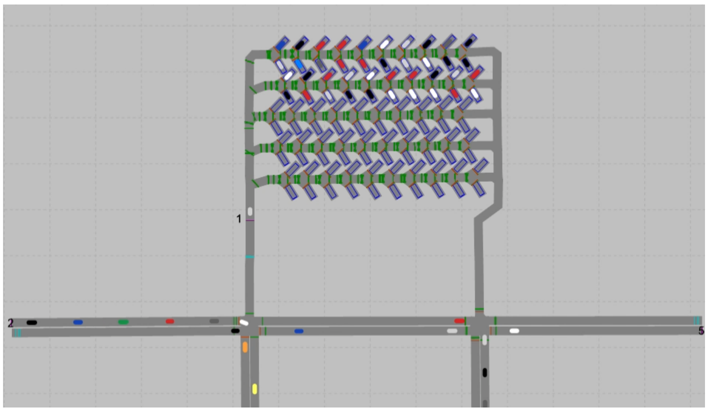

Figure 4.8: Congestion levels: Slot controlled simulation at 114s

Fig.4.6 and Fig.4.7 (control cases) have a significantly higher build up of vehicles on each of the 4 incoming links. However, the slot based simulation 4.8 has less vehicles in the network as they are all scheduled to travel through the intersections in a more efficient manner. This leads to lower congestion and traffic build up over time, as shown after the simulation has run for 114 seconds.

4.2.1 Conclusion

Each metric evaluated during this project under the restrictions listed in chapter 3.2, showed improvements when the slot-based system was used, over the two control-cases. The limitations of the tests, were primarily that the slot-based system was only tested for 100% driver compliance, roads were restricted to one lane, the attendance ratio was limited to 25% and the slot-based system restricted intersection access to only one vehicle at a time. Even with all these restrictions, the slot-based system still returned lower average journey times, higher number of vehicles passing through the network and a reduced build up of vehicles on road segments.

5 Summary

This project focused on developing a slot-based co-operative driving solution for events, to alleviate congestion by organising the flow of traffic. This project explored whether this slot-based system is a more beneficial form of driving, rather than the current unco-operative approach. The system that was developed focused on analysing the feasibility and the benefits gained from incorporating this driver guidance into event management plans. The benefits were in respect to reduced average journey times, higher number of vehicles passing through the network and also the reduced build up of vehicles on road segments.

This project implemented a dynamic slot provisioning and control system for all vehicles entering the specified road network surrounding an event. The system simulates the desired driver guidance system with one hundred per cent driver compliance. Each road was allocated a number of slots based on the length of the road. Vehicles that entered any of the roads within the boundary were assigned a slot. The driver guidance system, in turn, provided guidance to the driver to maintain a central position within the assigned slot, as it moved. The guidance provided directed each vehicle in the simulation to their destination. The slot was propagated along the road at each time step in the simulation until the destination was reached. Vehicles that were attending the event were assigned a parking space in the parking lot and vehicles that were not attending the event were directed away from the event towards their destination. To reduce the complexity of the system, roads were restricted to one lane per direction. There were no turning restrictions at intersections except u-turns and routes were predefined within PTV Vissim.

A custom parking lot with diagonal spaces and a one-way system was designed, accompanied by an entrance and exit to the parking lot. The road network had four incoming roads, where vehicles initially join the network. Intersections were only capable of granting access to one vehicle at a time to prevent collisions. All vehicles tested in each scenario were considered homogeneous.

6 Conclusion

From the evaluations of the implemented slot-based driver guidance system, there is a clear improvement to the flow of traffic in these scenarios. The road network where the guidance system was tested was specifically designed to replicate the road network at the entrance and exit of the parking lot of an event. Under 100% compliance with the system, there are significant improvements to the average journey times of vehicles travelling in the network. These improvements are in comparison to the two control cases tested. At higher volume levels, the slot-based system reduces the congestion levels experienced by the road network over time, leading to an increase in the overall vehicle throughput in the network. This higher vehicle throughput benefits drivers by enabling parking lots to fill up faster and drivers not attending the event to pass through in a shorter time. Accompanied by the intelligent parking lot management system, it reduces the time wasted by drivers on-approach to a full parking lot or even searching for available spaces within the full parking lot.

The overall biggest improvements were to the average journey times being notably more consistent for each vehicle input volume level tested. During the testing of Fig.4.2, the slot-based simulations average journey times only increased by a maximum of 5% across the four vehicle volume levels. On the other hand, the priority rules average journey times increased by up to 290% and the signalling simulation by 280%. This provides valuable evidence towards proving the hypothesis that the slot-based driver guidance system improves the flow of traffic around events and allows for journey times to be more accurately predicted.

6.1 Future Work

There are plenty of avenues left to explore for this project, as this implementation only scrapes the surface. The immediate future work concerning this project should be to successfully implement the scheduling of vehicles for the entire journey, preventing the need for vehicles to stop moving at any point. This accompanied by handling unexpected events and varying degrees of driver compliance with the system are the next steps in exploring the slot-based guidance system for events. The road networks should also be made more complex, with several lanes for each direction and turning restrictions. More road networks should be designed and tested, to ensure the results from this system are consistent. The more ambitious future work for this project could include phasing, allowing more than one vehicle on non-conflicting paths access to an intersection at any one period of time. There is also an opportunity to develop the mobile app, or explore the road-side units (V2I) and create a physical real-world implementation of this slot-based system.

Bibliography

[1] Vinny Cahill, Stefan Weber, Aline Senart, Anthony Harrigton, Douglas C.Schmidt, and Barbar Hughes. The managed motorway: Real-time vehicle scheduling. 2008.

[2] Dan Marinescu, Jan Curn, Mélanie Bouroche, and Vinny Cahill. An active approach to guaranteed arrival times based on traffic shaping. 2010.

[3] K Abdullah, N Kamis, N Azahar, S Shariff, and Z Musa. Optimisation of the parking spaces. 2012.

[4] H Wang, F Zhang, and Peng Cui. A parking lot induction method based on dijkstra algorithm. 2017.

[5] M Saidallah, A Fergougui, and A Elalaoui. A comparative study of urban road traffic simulators. *MATEC*, 2016.

[6] Philip Lieberman. *The Unpredictable Species: What Makes Humans Unique*. Princeton University Press, 2013.

[7] Orit Taubman-Ben-Ari, Mario Mikulincer, and Amit Iram. A multi-factorial framework for understanding reckless driving—appraisal indicators and perceived environmental determinants. 2004.

[8] Sasan Amini, Eftychios Papapanagiotou, and Fritz Busch. *Traffic Management for Major Events*. 2016.

[9] World Bank. Egypt - cairo traffic congestion study - phase 1 (english). 2010.

[10] TASS. Cooperative Driving. `https://tass.plm.automation.siemens.com/cooperative-driving`.

[11] TDMA. TDMA. `https://www.techopedia.com/definition/5089/time-division-multiple-access-tdma`.

[12] Chieh-Chang Li, Shuo-Yan Chou, and Shih-Wei Lin. An agent-based platform for drivers and parking lots negotiation. 2004.

[13] Shin Kato, Sadayuki Tsugawa, Kiyohito Tokuda, Takeshi Matsui, and Haruki Fujii. Vehicle control algorithms for cooperative driving with automated vehicles and inter vehicle communications. 2002.

[14] R Tachet, P Santi, S Sobolevsky, LI Reyes-Castro, E Frazzoli, and D et al. Helbing. Revisiting street intersections using slot-based systems. 2016.

[15] Dan Marinescu, Jan Curn, Mélanie Bouroche, and Vinny Cahill. On-ramp traffic merging using cooperative intelligent vehicles: a slot-based approach. 2012.

[16] R Roess, E Prassas, and W Mc Shane. *Traffic engineering*. Pearson, 2010.

[17] S Lämmer and D Helbing. Self-control of traffic lights and vehicle flows in urban road networks. 2008.

[18] B Lee, Jand Park. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. 2012.

[19] S Shladover. Cooperative (rather than autonomous) vehicle-highway automation systems. 2009.

[20] B Chen, Z Jia, and P Varaiya. Cooperative (rather than autonomous) vehicle-highway automation systems. 2001.

[21] Anthony Chrest, Mary Smith, Sam Bhuyan, Mohammad Iqbal, and Donald Monahan. *Parking Structures: Planning, Design, Construction, Maintenance, and Repair*. Springer, 2001.

[22] B Wolshon and A Pande. *Traffic Engineering Handbook*. Wiley, 2016.

[23] K Dsouza and S Hussain. Smart parking - an integrated solution for an urban setting. 2017.

[24] Fu Jiabin, Chen Zhenxiang, Sun Runyuan, and Yang Bo. Reservation based optimal parking lot recommendation model in internet of vehicle environment. *Communications, China*, 2014.

[25] R Tomar, M Prateek, and G Sastry. Vehicular adhoc network (vanet) - an introduction. *International Science Press*, 2016.

[26] Kodapan. OSM-Common Library. `https://github.com/kodapan/osm-common`.

[27] T Tettamanti and M Horváth. A practical manual for vissim com programming in matlab. 2015.

[28] Oracle Corporation. Java Programming Language. `https://www.java.com`.

[29] Google. Google Android 9.03 OS - Location tracking information.

[30] GDPR-INFO. GDPR - Right to be Informed, . `https://gdpr-info.eu/issues/right-to-be-informed/`.

[31] GDPR-INFO. GDPR - Right to be Forgotten, . `https://gdpr-info.eu/issues/right-to-be-forgotten/`.

[32] K Fysarakis, I Askoxylakis, V Katos, S Ioannidis, and L Marinos. Security concerns in co-operative intelligent transportation systems. 2017.

A1 Appendix


```xml
<way id="88614655" visible="true" version="6" changeset="48889214" timestamp="2017-05-22T13:45:44Z" user="VictorIE" uid="2008037">
 <nd ref="1028990940"/>
 <nd ref="2995124668"/>
 <nd ref="2995124674"/>
 <nd ref="1028990804"/>
 <nd ref="1028990905"/>
 <nd ref="2995124677"/>
 <nd ref="1028990992"/>
 <nd ref="1028990826"/>
 <nd ref="1028990930"/>
 <nd ref="1028991017"/>
 <nd ref="2995124680"/>
 <nd ref="1028990893"/>
 <nd ref="1028990900"/>
 <nd ref="1028990988"/>
 <tag k="highway" v="residential"/>
 <tag k="lanes" v="2"/>
 <tag k="maxspeed" v="30"/>
 <tag k="name" v="Monastery Crescent"/>
 <tag k="surface" v="asphalt"/>
</way>
```


Figure A1.1: Example of OSM Nodes data


```xml
<node id="1028990992" visible="true" version="2" changeset="24526102" timestamp="2014-08-03T23:29:54Z" user="mackerski" uid="6367" lat="53.3239343" lon="-6.3845313"/>
<node id="1028990993" visible="true" version="2" changeset="24526102" timestamp="2014-08-03T23:29:54Z" user="mackerski" uid="6367" lat="53.3220165" lon="-6.3863166"/>
<node id="1028990996" visible="true" version="1" changeset="6591590" timestamp="2010-12-09T02:30:08Z" user="mackerski" uid="6367" lat="53.3213777" lon="-6.3836261"/>
<node id="1028991003" visible="true" version="1" changeset="6591590" timestamp="2010-12-09T02:30:09Z" user="mackerski" uid="6367" lat="53.3228746" lon="-6.3854822"/>
<node id="1028991005" visible="true" version="2" changeset="24526102" timestamp="2014-08-03T23:29:55Z" user="mackerski" uid="6367" lat="53.3221216" lon="-6.3862833"/>
<node id="1028991013" visible="true" version="1" changeset="6591590" timestamp="2010-12-09T02:30:09Z" user="mackerski" uid="6367" lat="53.3221982" lon="-6.3840052"/>
<node id="1028991017" visible="true" version="2" changeset="24526102" timestamp="2014-08-03T23:29:55Z" user="mackerski" uid="6367" lat="53.3241216" lon="-6.3856781"/>
<node id="1028991022" visible="true" version="1" changeset="6591590" timestamp="2010-12-09T02:30:09Z" user="mackerski" uid="6367" lat="53.3205866" lon="-6.3831584"/>
<node id="1269046126" visible="true" version="2" changeset="24526102" timestamp="2014-08-03T23:29:55Z" user="mackerski" uid="6367" lat="53.3234581" lon="-6.3846993"/>
<node id="1269046152" visible="true" version="2" changeset="8406719" timestamp="2011-06-11T14:59:02Z" user="mackerski" uid="6367" lat="53.3240976" lon="-6.3821397"/>
<node id="1269046160" visible="true" version="2" changeset="8406719" timestamp="2011-06-11T14:58:50Z" user="mackerski" uid="6367" lat="53.3236914" lon="-6.3812975"/>
<node id="1269046163" visible="true" version="4" changeset="57956186" timestamp="2018-04-10T00:59:19Z" user="VictorIE" uid="2008037" lat="53.3197141" lon="-6.3834807">
 <tag k="traffic_calming" v="table"/>
</node>
```


Figure A1.2: Example of OSM Way data

	Vehicle Input Volume			
	1000	2000	3000	4000
priority	46.029	64.3294	102.2947	139.9857
signalling	51.10392	89.38439	118.1398	142.6378
slot-based	23.11234	23.22342	23.4349	24.31211

Maximum journey times (seconds)

	1000	2000	3000	4000
priority	45.68911	63.38845	99.3418	133.9182
signalling	48.74344	88.47378	115.0357	137.156
slot-based	23.00282	23.03836	23.0361	24.0836

Average journey times (seconds)

	1000	2000	3000	4000
priority	45.34921	62.44749	96.38887	127.8507
signalling	46.38296	87.56317	111.9317	131.6743
slot-based	22.8933	22.8533	22.6373	23.85509

Minimum journey times (seconds)

Table A1.1: Route 1 results

	Vehicle Input Volume			
	1000	2000	3000	4000
priority	32.20394	64.22314	84.56732	120.32451
signalling	35.21341	49.30013	84.99291	117.2956
slot-based	20.21345	20.34947	20.23953	21.29271

Maximum journey times (seconds)

	1000	2000	3000	4000
priority	31.952569	62.417594	82.498847	115.418357
signalling	34.915355	48.613101	83.853705	113.803708
slot-based	20.02837	20.10289	20.58263	21.937326

Average journey times (seconds)

	1000	2000	3000	4000
priority	31.905138	60.835188	80.497694	110.536714
signalling	34.73071	47.926202	82.80741	110.407416
slot-based	19.95674	19.90578	20.96526	21.974652

Minimum journey times (seconds)

Table A1.2: Route 2 results

	Vehicle Input Volume			
	1000	2000	3000	4000
priority	20.4	55.94831	66.91303	106.3922
signalling	30.4	44.96816	89.232	100.3847
slot-based	13.06	13.0923	13.1284	14.73211

Maximum journey times (seconds)

	1000	2000	3000	4000
priority	20.267117	54.45352	64.23511	102.5226
signalling	30.230158	44.07181	87.67112	96.38716
slot-based	13.00382	13.03826	13.00982	14.06937

Average journey times (seconds)

	1000	2000	3000	4000
priority	20.134234	53.90704	61.57023	98.65296
signalling	30.060316	43.24362	86.11024	92.38962
slot-based	12.94764	12.98422	12.89124	13.43874

Minimum journey times (seconds)

Table A1.3: Route 3 results