Philip Helenius

Efficiency analysis of transit signal priority using microsimulations:
A case study in Turku City

Master's Thesis
Information Systems
Supervisors: Prof. Anssi Öörni
Faculty of Social Sciences, Business
and Economics Åbo Akademi
University October 2020

Research field: Information Systems	
Author: Philip Helenius	
Title: Efficiency analysis of transit signal priority using microsimulations: A case study in Turku City	
Github link to simulation files: https://github.com/phelenius1/Efficiency-analysis-of-transit-signal-priority-using-microsimulations-A-case-study-in-Turku-City	
Supervisors: Dr. Anssi Öörni	

Abstract:
This paper studies the effectiveness of implementing a transit signal priority (TSP) system in Turku city using microsimulations. TSP is a technology that prioritises buses at traffic lights to make the public transport system faster and more punctual. The research was conducted by building a microsimulation using real-world data from induction loops placed along the route and with the help of bus logs. The paper tests different variables by altering the detector distance between 60,90,120,150 and 180 metres. The effect on non-prioritised traffic is also studied in the paper.

The TSP system being implemented in Turku is a centralised GPS based TSP system. While GPS based TSP systems exist around the world, there are many other technologies also used for TSP systems. Therefore, the paper reviews and compares the result from previous studies on what differs between the technologies. The literature chapter also reviews studies that can help Turku to improve their system further, once the TSP system has been implemented.

According to the results, a TSP system reduces the total route duration by 7.8-10.8% for the east direction and 10.5-15.8% for the west direction. The lower limit is when the detectors are placed 60 metres from the intersection, and the upper limit is when the detectors are placed 180 metres from the intersection. Moving the detector distance 30 metres farther from the intersection showed a significant improvement. However, the returns were diminishing after 150 metres from the intersection. Placing the detector farther away from the intersection also doubled the number of extensions given, and after 150 metres, the other priority types started to reduce. Extensions are preferred to red light shortenings and extra greens, as they have the least amount of disturbance on the general traffic. The result also showed that implementing a TSP system did not have a significant effect on general traffic.

The paper also reviews whether simulations should be used more in traffic planning within Turku city. Therefore, the paper explains in great detail all the steps necessary to build a simulation in SUMO. From the result, it is recommended that Turku city does not focus on large scale simulation as they are very time consuming, and their accuracy can often be questioned. However, it is recommended that Turku city continues to use small scale simulations as they are much quicker to build compared to large scale simulations. With simulations, the traffic planning department can quickly test many different variables and gain a more comprehensive understanding of the system, which could take up to years to test in the real world.

Keywords: Transport signal priority, split-phasing, SUMO, microsimulation	
Date: 19.10.2020	Number of pages: 83

Table of Contents

Table of figures

Table of tables

1. Introduction

The number of vehicles in cities continues to rise with urbanisation. Therefore, city planners must try to minimise the amount of congestion within cities. One of the best ways to reduce congestion is by having a working public transport system, but how does one achieve a highly functioning public transport system?

The optimal public transport system depends on the size of the city. In larger cities, there are often local trains or a metro system by which the passenger can quickly travel from one district to another. Another popular solution is tramways, which can either travel with the traffic or have its separate lane. However, the most common public transport system are buses, which can be found in almost all cities. Buses are more common than the others because they are less expensive to implement compared to the other examples. Bus routes can also be changed with ease, while the others require large new rail networks to be built.

De Oña et al. (2013) concluded in their study that frequency, punctuality, speed, safety, cleanliness and accessibility are the most important factors according to passengers when riding a bus. Frequency and accessibility can be improved by adding more buses or more bus routes, but this is expensive. Therefore, the local bus department must determine an appropriate number of buses depending on how many passengers will ride the bus. Speed and punctuality are more challenging to control as the bus drivers might drive at different speeds. Additionally, the congestion is impossible to control, and the bus stop duration will fluctuate depending on how many passengers aboard the bus.

For buses, increased congestion is especially problematic because it means extra dwell time at traffic lights. This causes variability in the bus schedule, which causes further dissatisfaction among passengers because they cannot reliably plan their trips. To counter the variance in trip times buses have implemented some slack time to their bus routes, which is used on designated bus stops where they can hold or leave early, depending on whether the bus is ahead of schedule or behind (Anderson & Daganzo, 2019). The amount of slack time depends on how significant the variances in route-times are; a large variance requires more slack time to counter it. More slack time means that there will be more time planned for the bus route, which causes dissatisfaction among the passengers. Buses being ahead of schedule might cause even more dissatisfaction as it would mean that the bus must stop and wait at one of the designated bus stops until it is back on schedule.

One of the most common strategies to decrease variance in bus route times is to implement transit signal priority (TSP) system for buses. Because a significant share of the variance in bus route durations happen during dwelling times at traffic lights, it is beneficial to implement a TSP system to counter this. By having a TSP system implemented, the bus dwell time should theoretically be close to zero, because in a perfect world it would never have to wait at a red light. In reality, many different factors decide whether the bus will gain priority or not. It is also important to remember that a significant proportion of the variance in route lengths comes from bus stops. Nevertheless, TSP systems have been proven to decrease the variance in travel speeds and reduce the route duration, which is why Turku city is now considering implementing a TSP system for their bus routes.

1.1 Context

Turku is a small city located by the sea in south-west Finland. In 2019, Turku's population was estimated to be roughly 193 000 (Turun väestökatsaus, 2019). The city continues to grow yearly; hence the congestion in the city increases. Therefore, Turku is continuously researching new technologies and potential solutions that would decrease congestion in the inner city.

In 2009, Turku decided that they would build a light-rail. A light-rail is a fast going tram-line, that is not driving among the cars, which allows it to go faster than the regular traffic (Turku, 2020). The problem with light-rail is that it is expensive, and in a small-sized city, there is a risk that the rail will not have enough user to become profitable. However, in 2020, Turku decided to continue developing a potential light-rail for the city. Even if the light-rail project is executed, there will still be a need for bus traffic as the rail will just cover the most densely populated areas around the city centre. Therefore, improving the bus system in Turku city will still be relevant.

In Turku, Föli is the actor who handles all local traffic and bus routes to nearby municipalities. Föli has continuously been researching on how they could increase the popularity in bus use. One of the latest decisions is to change the trunk lines for 2022. The idea is to increase the efficiency, availability and the number of passengers among Föli buses. Another research area has been bus rapid transit (BRT), which is when there is a designated bus lane on the road. By having a bus lane, the buses would never have to queue among personal vehicles during congestion. The problem is that constructing an extra bus lane is expensive due to infrastructure changes and, in some cases, it might not even be possible to add extra lanes to the road because it is an inner-city area. Föli is now interested in implementing a TSP system because it is an inexpensive change that could potentially decrease the average route duration and variance, which both would make buses more attractive to the population.

Transit signal priority is a well-established concept, which has been implemented in several cities around the world in the USA, the UK, Japan, France, Denmark, Sweden, Switzerland, Finland, Germany, Australia, Austria, Italy, New Zealand (Gardner et al., 2009). Even if the technology exists in several countries, it still differs somewhat between the countries. This is because the parameters must be optimised individually to each city. The specific configurations also depend from intersection to intersection and can differ tremendously.

Therefore, there are many different priority strategies within TSP systems. TSP systems can be divided into a passive or an active TSP system. A passive TSP system is when the signal phases are weighted or optimised to predict when a stream of buses will approach the traffic light. In a passive TSP system, everything is pre-planned, thus the system is not optimising itself in real-time (Ahmed, 2014). These systems are outdated and unreliable as it is impossible to predict when exactly the bus will be in the intersection. On the contrary, an active TSP system reacts to incoming buses in real-time, which makes it much more reliable in giving signal priority to buses. The bus can either signal the traffic light by using a beacon that continuously emits a signal to the traffic light or by using GPS to keep track of all buses in the traffic light central.

Gardner et al. (2009) list four different priority strategies that exist, these are extension only, priority to late buses, a mix between extension and priority for late buses and full priority. “Extensions only” means that the traffic light will extend the green phase for a specific timeframe when a bus approaches the intersection. An extension can only happen if the traffic light is about to turn red in the bus direction. The second strategy will only give priority to buses that are behind schedule. This includes the option that the traffic light will change to green when a bus is approaching the intersection. In the mixed strategy, the traffic lights will give full priority to buses that are late on schedule, while buses that are on schedule, will only receive traffic light extensions. The full priority strategy gives all buses priority, independent on whether the bus is late or not. In some cities, there might be other strategies implemented, but they are all built based on the same core strategies.

The optimal priority strategy depends on what the traffic planning department consider is most important with the TSP system. For example, if the effect on the general traffic is to be minimised, then the extension only strategy is the best. On the contrary, the full priority strategy has the most negative effect on the general vehicle flow. However, it reduces the bus route duration the most. The optimal priority strategy also depends a great deal on where the intersection is located and what the vehicle flow in the intersection is.

Today, it is relatively easy to measure how much the buses are delayed because they often have a time system implemented to them. This is not possible for personal vehicles because there often no data available that could measure the delay caused by a TSP system, which is one of the reasons why simulations are excellent for measuring the effects of a TSP system.

1.2 Objective

In this paper, a simulation will be built to understand what effects a TSP system would have in Turku. The simulation results will then be compared to actual data from the same intersections to verify that the simulation describes the real-world results.

Different optimisation possibilities can then be analysed, which could take years to test in the real world. A simulation will give us a better understanding of how different factors affect the system. The requirements to build a simulation are time, data and a computer, which makes it much less expensive for testing compared to implementing a TSP system and then testing whether the system has any benefits.

Today, many TSP systems function by having a radio transmitter installed on every bus. The radio transmitter emits a signal with a given interval. When the bus approaches the intersection, the traffic light will receive this signal and therefore know that a bus is approaching the intersection. The traffic light will then choose the optimal strategy depending on the conditions.

In this study, a GPS based TSP system will instead be analysed. While the system itself will be almost identical to other TSP systems, there are complications which do not exist when using a radio signal, such as signal delays and imprecision of the actual position. In the worst-case scenario, the signal delays are in such a high magnitude that the traffic lights will not be able to adequately react to the incoming bus, which would render the TSP system useless. In the literature chapter, other TSP systems will be examined, then TSP systems in Finland will be discussed. Finally, previous researches about GPS based TSP system will then be revised.

Building a simulation of the whole city of Turku is not feasible because of time constraints, but also because it would be impossible to make a realistic simulation with the data available today. Simulating the inner city is also not feasible because of the large number of pedestrians. It is possible to simulate pedestrians, but there is zero data about pedestrians in Turku. Additionally, pedestrians are more random compared to vehicles (speed, direction) which makes them harder to simulate. Therefore, a bus route called route 99 was chosen for this paper, which goes around the city centre.

Today, the traffic planning department in Turku already uses simulations to simulate single intersections, but seldom use them to simulate several intersections simultaneously. This study will give insight into whether simulations can produce accurate results even if there is a lack of data in larger simulations. This will be done by comparing the simulation results with actual data from the same intersection as in the simulation. This information can then be used to deter whether simulations should be used in the future again for similar projects.

The research questions in this study are the following:

1. *Would a transport signal priority system be effective in Turku?*
2. *Could GPS signal cause a problem in the system?*
3. *Is it beneficial to move the detectors farther away from the intersection?*
4. *How is the other traffic affected by the TSP system?*
5. *Should simulations be used more in traffic planning in Turku even if there is a lack of data?*

Research questions 1, 2, 3 and 4, will be studied with the help of the simulations. Research question 5 will be answered based on personal experience from working with SUMO. Since research question 5 is not being studied in this paper, it will be answered in the practical conclusion chapter.

2. Turku bus route 99 details

This chapter will explain why this specific route was chosen for this study. First, the route will be illustrated and some details about it will be given. Secondly, the bus stops included in the simulation will be described, and some details about exceptional cases will be explained. This chapter helps the reader understand what is being discussed later.

2.1 Route 99

In Figure 1, bus route 99 is marked with a blue line. Only the part between the first (839 and 1559) and the last (1636 and 1637) bus stops will be included in this paper. This segment was chosen because it includes all the intersections where it is possible to implement a TSP system on route 99. Additionally, by not including the segments outside these bus stops, the number of random factors can be minimised.

The route in this study begins at bus stop 1559/839 at Länsikeskus on the road called Markulan tie and continues along the road until the intersection after bus stop 1588/1590. At intersection 410 in Figure 2, the bus turns right to Vanha Tampereentie. The route then continues on Maunu Tavastin katu, which changes name to Gregorius IX:n tie between bus stop 1406 and 1405, and then changes name to Halistentie after crossing the water stream. The simulated route ends at the intersection after bus stop 1636/1637. The journey is roughly 6.4 km long with the speed limit being 50 km/h, except for the part between bus stop 1636/1637 and the intersection with Hämeentie. Since this part is not included in the simulation, the speed limit was set to 50 km/h throughout the route.

This specific route was chosen because it is a popular bus route that includes several intersections where TSP can be implemented. The studied road is one of the busiest roads around Turku city. Additionally, the route is situated outside of Turku city centre, and therefore there will be less disturbance from pedestrians. This is important because otherwise, it would be more difficult to analyse how much reduction a TSP system brings in route times. Pedestrians are complicated to simulate because they add a new layer of randomness to the simulation. For example, in zebra crossings, there is often an interplay between the pedestrians and the vehicles in the real world, which is impossible to simulate accurately. There is also no data available on pedestrian traffic, which means that it would have required field studies.

For bus route 99, there are four different buses with different starting positions or different last stops. These four buses go in both directions. The first one is called 99IP and goes from Ilpoinen-Skanssi-Lauste-Länsikeksus-Pansio, and then backwards. 99UT goes from Uittamo-Skanssi-Lauste-Länsikeskus-Perno. 99IT goes from Ilpoinen-Skanssi-Lauste-Länsikeskus-Perno and 99UP goes from Uittamo-

Skanssi-Lauste-Länsikeskus-Pansio. The different routes for bus 99 are irrelevant in this paper because the studied segment is part of every bus route.

2.2 Bus stops

All bus stops in the simulated route can be seen on the map in Figure 1. Most of the bus stops are placed within approximately the same distance from each other. There are in total 18 bus stops in both directions, which makes the total number of bus stops 36.

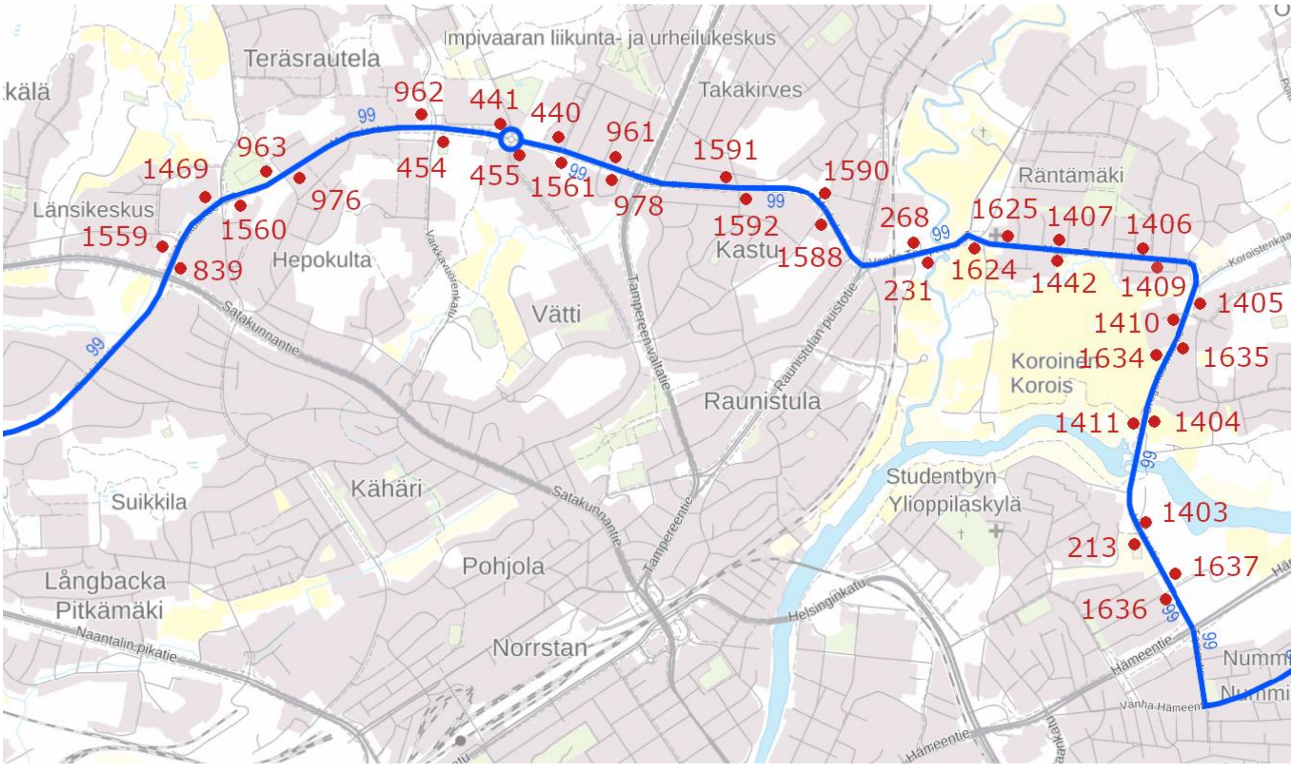

Figure 1: The simulated bus route and its bus stops

Throughout the route, some of the bus stops could be overcrowded due to services offered close-by. The first ones are bus stops 1559 and 839 at Länsikeskus. Länsikeskus is a small centre for stores outside of Turku. Therefore, these bus stops might have more passengers compared to other stops. The main road to Turku centre from the North is also situated next to Länsikeskus. Therefore, the bus stops work as a layover for people coming or going to the city centre. Bus stop 1559 and 839 were ignored in the simulation because they are two of the four slack bus stops in this segment. A slack bus stop is a bus stop where the bus can stop for a longer time in case it is ahead of schedule. Because there is no data on how long the slack time was, it is impossible to determine what the actual duration of the bus stop is.

Bus stop 440 and 1561 are the bus stops for the swimming hall in Turku and bus stop 441 and 455 are the bus stops for the sports stadium. Bus stop 213, 1403, 1405, and 1410 are close to public schools. These bus stops are all likely to have extreme values because of large groups going together.

2.3 TSP intersections

There are in total eight traffic light intersection that can be equipped with a TSP system in the studied route. These are 333,334,336,405,409,410 and 420, and can be found in Figure 2. The naming of the intersections is according to how Turku city internally names them. Therefore, the same method will be used in this paper.

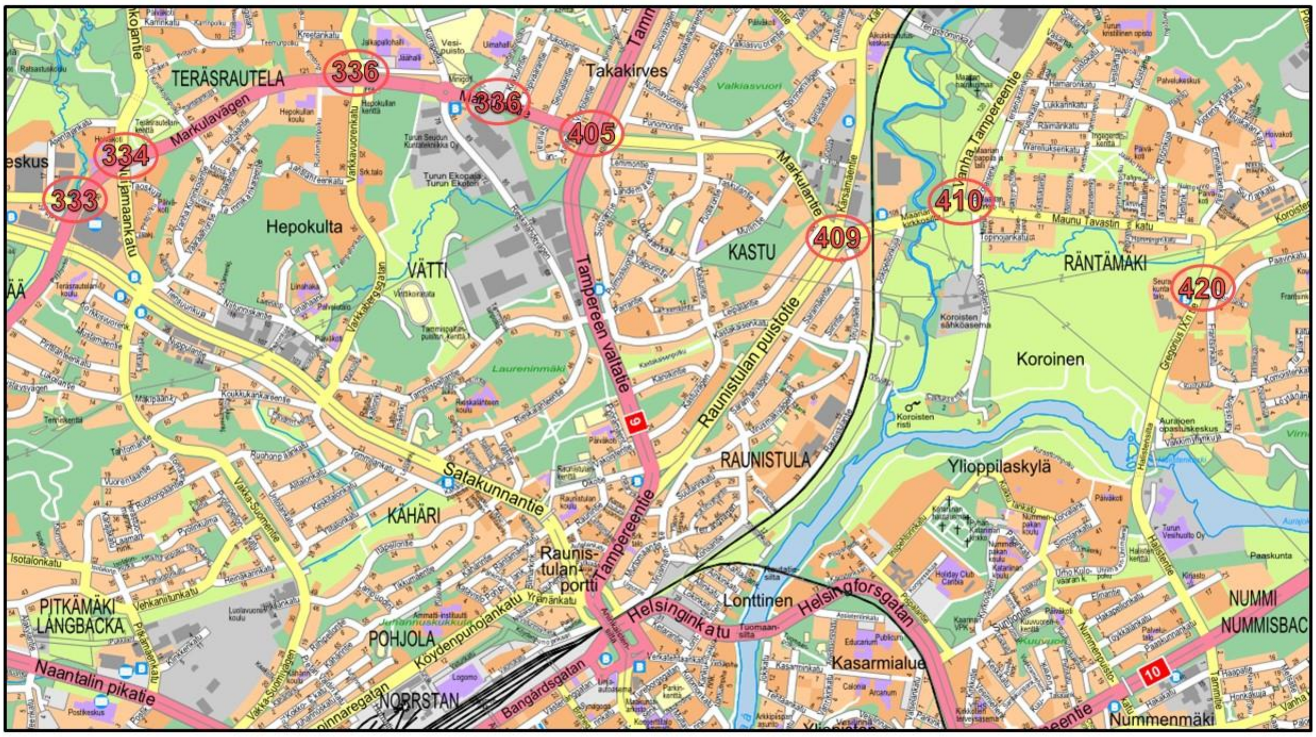

Figure 2: All TSP equipped intersections

The largest intersections in the studied route are 405 and 409. These intersections are both crossing with main roads to Turku city centre from the North. In contrast, the least congested intersections are 336 and 420. Intersection 336, is the intersection between Markulantie and Kekkerintie. It is the intersection which turns to the swimming stadium. Intersection 420, is between a grocery store and residential area. More detailed data about the intersections will be given in the data collection and analysis chapter later.

3. Literature review

As discussed earlier, TSP systems have already been implemented in several countries. This chapter will first explain how a TSP system works, and then study to what degree a TSP system can improve the bus duration in a city.

The chapter begins with exploring previous TSP systems studies around the world and then goes into more detail about TSP systems used in Finland. Additionally, papers on different signal technologies will be examined. The chapter will review how the results differ depending on the solution used and why it is crucial to choose the correct solution depending on what effect one desires to achieve with the TSP system.

3.1 Brief history of TSP

The use of TSP systems has already been around for almost 50 years. One of the first adopters was London, who installed their TSP system in the 1970s in isolated intersections (D'Souza et al., 2010). The system was built on their vehicle actuated system, which was called the "D-system". Bus detection was handled by installing transponders to the underside of the bus. These detectors would then continuously emit a signal and the traffic lights would then receive this signal when the bus was close to an intersection. The traffic lights used the full priority strategy, which meant that it gave priority to all buses. After a successful trial period, the system was extended to 50 intersections and has since been implemented to many more.

In 1997 Toronto city implemented a TSP system to 10 intersections using infrared technology (Toronto Transit Commission, 2004). After analysing the data retrieved, the city concluded that they were able to reduce the transit delay by up to 46% by implementing a TSP system, which means that the delay reduced by 46%, not the total travel time. The data also showed that no significant delay was caused to the crossing traffic. After reviewing the results, several recommendations were issued to improve the system further. One recommendation was to use loop-based detection, which was less expensive than the original detection system. The city also decided to use radio signal to send the priority request from the bus to the traffic light instead of infrared because it frequently failed. In 1998, Toronto expanded the system to 33 intersections and reported that the system saved over $235 000 in operating costs annually.

3.2 Current studies on TSP

Since TSP has existed for several decades already, the technology has seen significant improvements since it was first introduced. However, optimising a TSP system is difficult because of the many different factors being affected. If the TSP system causes too much delay for crossing traffic, it might have an unfavourable result on the bus too due to increased congestion in the whole intersection. The system is also dependent on how well the traffic light in the intersection works. The question then becomes, how does one implement the perfect solution? The ideal solution is heavily dependent on the location of the intersection because of the different amount of traffic, and intersection layout affects the effectiveness of the system. Therefore, several studies have been conducted on how to optimise and choose the correct settings for the TSP system.

The goal of the studies about TSP systems has shifted during the last decade. Before the actual TSP system was being studied, but today most studies are about optimising a TSP system. The papers often use a TSP system with other technologies to achieve a more optimal system. VANET is one of the most popular trends currently, which stands for vehicular ad hoc network and means that the buses can give information to each other (Hu et al., 2016; Lee et al., 2017; Yang et al., 2019). These studies were ignored in this paper because VANET technology is not relevant to this study.

An evaluation report done by the Boulevard in Miami shows why it is so difficult to optimise a TSP system. The report was done by using both manual labour and GPS location to calculate the dwell time, route time, signal delay on a roughly 10-mile route (Pessaro & Van Nostrand, 2011). The results show the change signal delay when using a TSP system. In most traffic lights the signal delay shrank, but the number differed a great deal between intersections. At one intersection, the signal delay shrank by over two minutes, while there was no significant difference at others. There is also no correlation between the length of the signal delay before implementing a TSP system and the benefits of implementing a TSP system. The results also showed that for some traffic lights, the signal delay even grew after implementing a TSP system. For example, at one intersection the signal delay was 0 seconds with TSP system off, but after turning TSP system on, the signal delay rose to 26 seconds. The results in the paper show that on average, there was a 4% reduction in signal delay for the whole route. The bus route time shrank by 12.1 % by implementing the priority system and the total bus route duration by four minutes.

In Portland, Albright & Figliozzi (2012) studied how the effectiveness of the TSP system depends on the intersection. They were specifically interested in how the effectiveness of the TSP system changed between congested intersections and less congested intersections. In the paper, a model was built to identify which factors affect performance the most. Another research objective was to study the lateness recovery of the bus. The studied route was an 8-kilometre corridor that has 14 signalised intersections and 54 bus stops. The TSP system was configured to only give priority for buses that were more than 30 seconds late. The different factors studied were: dwell time at intersections, duration at bus stops, time, passengers, lift for disabled used, segment distance, segment travel time. The result showed that a longer distance between the bus stops resulted in a lower chance to recover from the lateness. Time of the day, the number of bus stops and the number of passengers were all found to be significant variables. Results showed that the major intersections had the least improvement from implementing a TSP system. The conclusion from the study was that the TSP systems performance is highly localised between intersections. The buses with the most severe lateness showed the most significant improvements. While having a lower dwell time at traffic lights and a lower passengers number showed a better recovery rate from the schedule lateness.

In 2008 a new TSP system was installed in London called iBus (D'Souza et al., 2010). The new system uses GPS technology instead of a beacon to locate the bus. The bus sends its location every 30 seconds to a central computer system situated in London using the General Packet Radio System (GPRS). The bus has a virtual detector implemented to it that consists of inbuilt detector points, which trigger when they match the onboard navigation system. When the virtual detector is triggered, the bus sends out a priority request to the approaching traffic light. By using GPS and GPRS, it became possible to add multiple detection points when approaching a traffic light. Having multiple detection points means that it is possible to more accurately estimate when the bus will arrive at the traffic light. At the first detection point, the bus gives an estimate of when it will arrive at the traffic light. The second detector is then able to re-estimate the arrival time based on how long it generally takes to arrive at the traffic light from that point. Additionally, the data between the first and second detector can be used to give an even more accurate estimation. The two-detector system showed in a simulation environment that there is an increase in bus delay savings. Most of the increased delay savings came from buses that gained priority extension at the traffic light.

Hounsell et al. (2008) discuss in their paper the challenges GPS errors bring in the new iBus system. GPS errors are troubling because they cause more variability in journey time between the detection point and the upcoming intersection. Therefore, the bus might miss the green light extensions it was given. To test the effects of GPS error, the authors built a theoretical model and a simulation. The GPS error was set to 10 metres in the model. The result showed that the variability increased by 56.6%, and 0.47% of the buses missed the green light extension due to GPS error, when the detector was 50 metres from the intersection. If the detector was set to 150 metres from the intersection, then the variability shrank to 7.6% and missed green light extensions shrank to 0.33%. The simulation results showed a 2-5% decrease in bus delay savings, compared to zero GPS inaccuracy. The decrease was also shown to be relatively consistent for all detector ranges varying from 10 metres to 150 metres. This is explained by the fact that GPS errors mainly affect green light extension, and these generally occur when the detector is placed farther away from the light, but the standard error in journey time to the intersection also increases as the detector is farther away and therefore the effects balance each other out. The simulation results were consistent with the theoretical model built earlier.

While the iBus system is similar to the one researched in this system, there is still one significant difference. In iBus, the system is built onboard the bus, which means that the bus sends the priority request directly to the traffic light controller. The TSP system researched in this paper will send the signal to a private company where the data will be analysed, and then a priority request will be sent to the company that handles the traffic lights. This could cause signal delays that do not exist in the iBus system. Otherwise, the iBus system is almost identical to the system used in Finland. In London, they use SCOOT (Split Cycle Offset Optimisation Technique), which is practically identical to the Finnish traffic lights system (Gardner, D´Souza, Hounsell, BShrestha, et al., 2009; Salonen, 2010). In Finland, the system is called SYVARI, and it will be discussed in chapter 4.

Yao et al. (2009) tried a different approach in their study. They studied how to optimise signal priority with the help of the queuing model. To test this, they built a simulation model of an oversaturated intersection where ten cars are able to pass the intersection per green light phase. If the 11th vehicle were a bus, then the light would extend its green time. In the paper, they tested three different parameters and concluded that the best strategy would be to extend the green light if the 11th vehicle is a bus. If the 12th vehicle is also a bus, then it would also be let through the intersection. Increasing these values showed little improvement and caused more disruption in the general flow. An interesting observation made from watching traffic cameras was that the traffic was less likely to be interrupted by late pedestrians if the first vehicle was a bus.

In Toulouse, France, traffic signal priority was tested on two bus routes under the European project called CIVITAS MOBILIS (Monzón et al., 2016). The project is meant to bring together different actors around Europe to share their expertise within traffic planning (Doucet, 2008). The results from the test runs showed that route time improved by 5-24%. The average dwell time at traffic lights also reduced by 52%, which averaged to 9 seconds per traffic light.

In Novrich, the UK, two different strategies for bus priorities were tested (Symonds, 2014). One of the strategies was to give only the buses that were behind schedule priority, while the second strategy gave priority to all buses. The results show that the TSP system was able to decrease the dwell time at intersections by up to one minute per intersection. When surveying the bus drives, they reported that they preferred the strategy which gave priority to all buses. However, the result showed that this strategy caused more variance in bus route duration compared to the priority-for-late-buses-only strategy. The decision-makers concluded that reliability and punctuality are more important than speed because they give a positive incentive to use buses. The paper suggests that one potential solution for the full priority strategy could be to inform the driver when he is running early or late and therefore minimise the variance in bus route times.

3.3 TSP systems in Finland

HSL is the bus actor in the capital region of Finland. In Helsinki, the TSP system is called HELMI. HELMI is an abbreviation for Helsinki public transport traffic signal priority and passenger information in Finnish. The system was implemented in 1999 and has now been in use for over 20 years

(Liikennevalot.info, n.d.). When the system was first introduced, it had many sceptics questioning its functionality. This was because the system used radiofrequency instead of a beacon, which was the standard at that time. The sceptics were proven wrong by a large margin, mainly because they were not able to predict that telephones would become as prevalent as they are today. The TSP system uses the same radio frequency technology as mobile phones but instead works on a private network. On implementation, three base receivers were placed on Hanasaari, Salmisaari and Ruskeasuo power plant chimneys. The buses receive and send live updates to the traffic control centre via these base receivers. The transit signal priority system is an onboard system, which works by having GPS hardware on the bus. When the bus is approaching an intersection, it sends out a radio signal to the base receivers, which is forwarded to the central computer. The traffic light will then be informed that a bus is approaching from this direction and give priority accordingly. The traffic light will only give signal priority to buses that are late on schedule.

In Tampere, Mattersoft handles the transit signal priority, which is the same company as in Turku (Inkiläinen, 2012; Mäkelä, 2020). They use GPS signal to recognise when a car is approaching a traffic light. Inkiläinen (2012) studied how to optimise transit signal priority by applying different parameters in the Mattersoft software. The experiments included six intersections on Pirkankatu, just outside of the inner city. The tested variables were: Priority value, Observation distance and late-bus values. The priority value decides which bus will receive priority when multiple busses are approaching the traffic light. The observation distance is the distance from the intersection where the Mattersoft software gets notified that a bus is approaching. The observation point is to determine which bus will obtain priority first and has nothing to do with the priority trigger point, which has been discussed earlier in this paper. Changing the observation distance can have a significant effect because if it is too large, then the lower priority bus will have to wait longer until the higher priority bus has passed the intersection. Late-bus values decide which buses will gain a priority based on if they are ahead or behind of schedule.

In total, six tests were done, and they collected data for one week for each test. The test results showed that the observation distance had little effect on the results. The two variables tested were 300 metres and 500 metres. When testing different priority values, the results showed that the optimal solution was to give the crossing buses a higher priority compared to the ones driving along the main road, Pirkankatu. By doing this, the study was able to reduce the travel time for every bus except one. For the late-bus value, the optimal value was -30 seconds as the minimum value and 600 seconds as the maximum value. This means that buses within 30 seconds ahead of schedule or up to 600 seconds behind schedule, will gain priority at the traffic light.

3.4 TSP effect on traffic safety

Implementing TSP systems can have unexpected results due to change in the traffic flow, which causes the drivers to become impatient because they must wait slightly longer at traffic lights due to priority request. Today, the safety aspects are seldom studied, but there are some studies on the subject.

Li et al. (2017) built a simulation based on traffic data from Toronto from the years 2006-2010. The results showed that all vehicle crashes decreased by 0.1%, if red shortening was implemented, but if both red shortening and green extensions were implemented then the number of crashes increased by 1.6%. The paper concludes that implementing a TSP system should cause an increase in the total number of crashes, but it does not address the severity of the crash.

Song & Noyce (2019) extended their study on TSP effects on traffic safety to pedestrians. This is interesting because pedestrians might also become frustrated from waiting longer times at traffic lights after TSP implementation and therefore cross the road in risky situations. In their study, they analysed real traffic data from 1995-2010 in Portland, Oregon. The TSP system had been implemented in 2002. For the study, they used a method called ITSA, which measures the effectiveness and changes in the effectiveness of the measure before and after implementing a TSP system. In the study, they concluded that the total number of crashes decreased, while the fatal crashes increased. Pedestrian-involved crashes and bike-involved crashes also decreased. The paper admits that several factors might skew the results, such as only the number of crashes were used without more detailed data, the data is also gathered from police and hospital records which might not always be correct. Traffic volume was also not taken into account. The result still agreed with other studies on traffic safety that a TSP system reduces the number of crashes (K. Goh et al., 2013; K. C. K. Goh et al., 2014; Naznin et al., 2016).

Based on the contrary results from the different studies, there is no definite answer to whether TSP systems affect traffic safety. However, the study in Portland is more credible due to actual data being analysed.

3.5 Schedule-based or frequency-based bus service

An important question when planning bus routes is whether passengers will care or even know about timetables (Gentile et al., 2016). If the service is infrequent or irregular, then it is likely that the passengers will try to time their arrival at the bus stop according to a specific bus. However, if the bus arrives frequently, then the passengers are unlikely to plan the arrival.

Whether to show the time-schedule or not is an important decision that the bus operator must make when planning the bus routes. The operator must decide what schedule information should be displayed to the public, and this decision should be made based on how the passengers perceive the schedule. If the bus operator chooses to publish all schedules, then there is much more weight on regularity. This form of information publishing is called schedule-based. On the contrary, if the bus operator does not display the schedules, then the journey time becomes more valued, while punctuality loses its importance. The service would then be frequency-based.

This decision should mostly be made based on how frequent the buses are. However, other factors can influence the decision too. If the journey times have a high variance, then it might be better to opt for the frequency-based service, because buses that are late or early according to the schedule might cause dissatisfaction among the passengers. High variance can be caused by unpredictable variables, such as driver behaviour, passenger number, congestion or traffic light signals. It is also possible that the bus is so frequent that it becomes unnecessary or almost impossible to memorize the schedule, for example, if a bus arrives every 3 minutes. On the contrary, if seat reservations are possible, then it becomes necessary to show schedules because otherwise, the seat reservation system does not work. Furthermore, if the bus arrives every 30 minutes, then the passengers will want to know when they should be at the bus stop.

It is possible to combine the two service types within cities. For example, it might make sense that the trunk lines are frequency-based, while the not as frequent bus routes going to remote areas would be schedule-based. A possible complication with using both systems is that it becomes more difficult for the passenger to plan routes that require multiple bus rides with different service systems.

Whether the bus operator chooses to have a schedule or frequency-based service is essential when deciding what transit signal priority should be implemented. If the service is frequency based, then the variance loses its importance, while speed becomes significant. Therefore, it is logical to give all buses priority. On the contrary, if the service is schedule-based, then the bus operator should aim to minimise the variance. The best strategy to minimise the variance is to implement a conditional TSP system, which has a preference for late buses according to the previous studies in this paper.

3.6 Evaluation report from other TSP systems

Evaluations from other implemented TSP system were collected in Table 1 to gain a better understanding of how much a TSP system can benefit the bus traffic. The table includes columns about whether the priority system was conditional and what sort of technology was used. It should be noted that the delay savings only calculate how much the delay reduced. For example, if there was originally a 30-second delay and with the TSP system, the route time shrank by 15 seconds, then the delay saving would be 50%.

Table 1: TSP effect after implementation in other cities

City	Priority for late buses only	Bus detection	Delay saving	Travel time	Variability	Disturbance on other traffic
Aalborg, Denmark	No	GPS and odometer	5.8 sec./Int.	4% reduction		
Auckland, New Zealand	No	GPS	11 sec/bus/Int.			
Brighton, UK	Yes	GPS		Reduced	Reduced	
Cardiff, UK	Yes	GPS		3-4% reduction	Reduced	1-2% increase
Chicago, USA	No	Loopdetection		15% reduction		Minimal
Genoa, Italy	Yes	GPS		7-10% reduction		
Gothenburg, Sweden	No			13-15% reduction		
Helsinki, Finland	Yes	GPS, door opening sensor and odometer		11% reduction	20% improvement in regularity and 58% in punctuality	
Kawasaki, Japan	No	Infrared beacon and GPS		5.1% reduction		
King County, USA	No	RFID	25-34%	5.5- 8% reduction	35-40% reduction	Minimal effect
London, UK	Yes	GPS and odometer	9 sec/bus/Int. at isolated intersections and 3-5 sec/bus/Int. at SCOOT intersections			
Los Angeles, USA	Yes	Loopdetection		19-25% reduction		1 sec/veh/Int. increase
Malmo, Sweden		GPS			Headway reduced from 10 min to 7.5 min.	
Miami, USA	No	GPS	4 %	12.1% reduction		
Oakland, USA	Yes	Encoded infrared	9 %			Almost non-existent
Prague, Czech Republic	No	Infrared beacon		2% reduction		
Seattle, USA	No	Passive radio frequency		35-40% reduction	5.5-8% reduction during peak hours	
Southhampton, UK	No	Infrared beacon and odometer	9.5 sec/Int.			3.8 sec/Int. increase
Stuttgart, Germany	No	Infrared beacon and GPS		Speed increased from 14.48 to 16.25 km/h		
Sydney, Australia	Yes	GPS		Up to 21% reduction	Up to 49% reduction	
Tallinn, Estonia		Infrared beacon		Speed increase by 2km/h		
Tacoma, USA	No	Encoded infrared	40% reduction			Minimal
Toulouse, France	Yes	GPS and odometer		5-24% reduction	Reduced	
Turin, Italy	No	Infrared beacon		19% reduction		Up to 2% increase
Vancouver, Canada	Yes	Encoded infrared and visual recognition			40-50% reduction	No noticeable impact

Note. Adapted from "Review of Bus Priority at Traffic Signals around the World", by Gardner, D´Souza, Hounsell, Shrestha, et al., 2009 *and from "Transit signal priority control at signalized intersections: A comprehensive review" by* Lin et al., 2015, *Transportation Letters, 7(3), p.168-180.*

The number of conditional TSP system in Table 1 was a surprise. In total, 22 different evaluations reports are listed in the table, and nine of them use conditional priority. It was expected that using non-conditional priority would have a more significant improvement in the route time reduction, but the values seem almost identical between the two conditional types. It should be noted that the improvement is dependent on the traffic flow before the TSP system was implemented. Therefore, cities with an enormous improvement likely had an inefficient traffic light system from the beginning. Furthermore, the improvement depends on how long the route was and how much space there was between the intersections.

According to the evaluations reports done after implementing a TSP system, the route time reduction can be anything between 2-40%. On average, the reduction seems to be somewhere between 5-15%. According to the evaluations the variability also shrank in every single city, where it was calculated. For conditional TSP systems, it was expected that the variability would reduce a great deal, while non-conditional was expected to see less improvement in variability. This is confirmed in Table 1, which shows that on average the variability reduced more in the conditional TSP system. However, the variability was seldom calculated in the non-conditional systems.

4. Reviewing the traffic light systems used in Finland

In this chapter, TSP systems and traffic lights systems in Finland will be discussed. First, the differences between the technologies will be discussed and what their effects are on the TSP systems effectiveness. Secondly, the traffic light system used in Finland will be explained in detail.

4.1 Priority request processing and delays

For transit signal priority, different technologies might be used. Today most TSP systems in Finland use a central database to process the priority requests. The old systems use radiofrequency receivers and priority request handlers to control the priority requests, which must be installed in every intersection. Hence, the old system is much more expensive compared to using a central database. Just the hardware for the old system costs 3800 euros per intersection, and that does not include the installation costs (Korvenmaa, 2020). This is the cost according to Swarco, which is the largest traffic light retailer in Finland.

Due to the high cost of the old system, the most common method today is to use a central database for processing. With the old system, the bus in Figure 3, would send its signal directly to the traffic light control box. However, the old system still has the same or even more delay compared to using a central database, even if it communicates directly with the traffic light (Korvenmaa, 2020). This is because a central database has much more processing power compared to a single traffic light controller. Another problem with the old system was that the processes that took place on the bus could only be updated when the bus was at the depot. Which meant that if there was a bug, then it required the bus to go back to the depot to fix it. By using a central database, all buses can be updated simultaneously regardless if the bus is in use or at the depot.

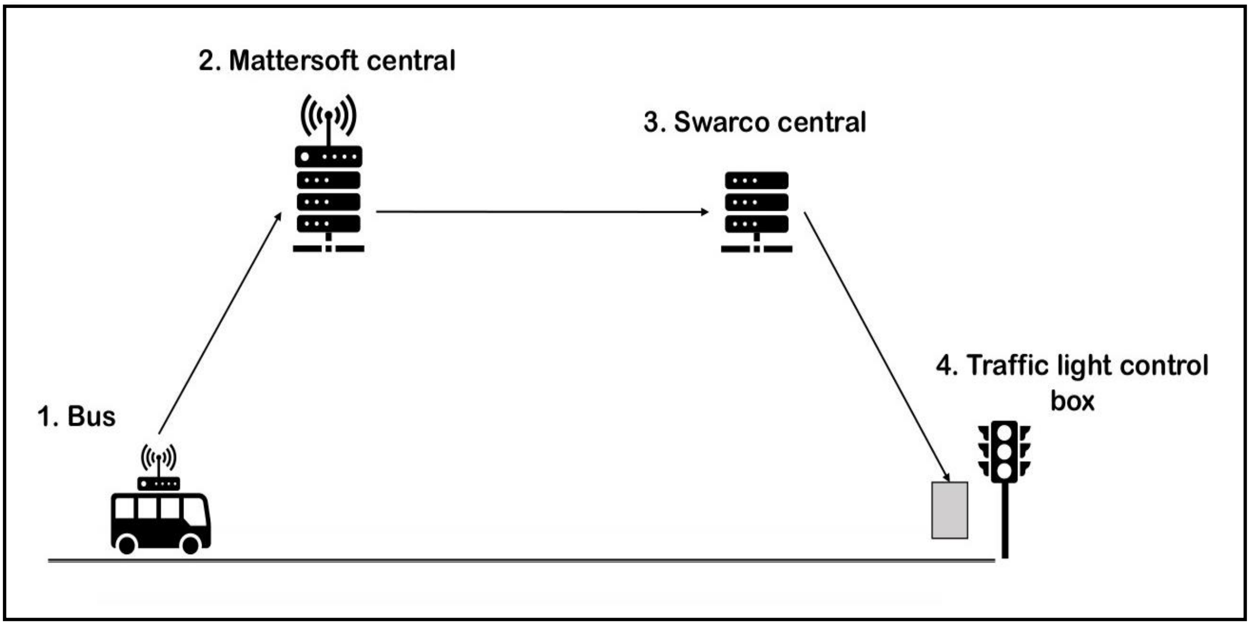

Figure 3: Signal routing for priority requests

In the new system, the bus sends a signal via a private IP connection or by radiofrequency to the central. In Turku, Mattersoft central handles the priority request (Mäkelä, 2020). Once the priority request has been processed, the request is sent to the company that processes the traffic light, which in Turku is Swarco. Swarco will then receive the priority request and send it to the traffic light control box in the correct intersection.

In Turku, the placement of the priority request in the phase cycle is processed at the Swarco central database (Korvenmaa, 2020). The traffic light control box will additionally check that everything is correct. However, it is possible to have the priority request processed by only the traffic light control box. The traffic control box simply orders the priority request depending on settings configured beforehand. For example, an emergency vehicle will gain priority before public transport. In the control box, there is a simple I/O switch, which is activated when a priority request is received

The signal between the bus and Mattersoft is sent either through a private IP connection or by radiofrequency. The delay between sending the signal from the bus to Mattersoft and Mattersoft processing the priority request and then forwarding it to Swarco central takes on average about 0.5 seconds (Mäkelä, 2020).

Between the Swarco central and the traffic light control box, the priority request can be transmitted in many different ways, such as through copper, optical fibre or IP connection. Most often, the system works with a private IP connection within the city. The data can then be sent via 4G, 3G or even older variants because the data size is so small that the connection has no trouble handling it. With a private IP connection, it takes around 400 milliseconds from Swarco central receiving the signal and for the traffic light control box to then receive the signal (Korvenmaa, 2020). Some older system systems that are still using serial communication transmission might reach up to 1-second signal delay, but there are almost none of these left in Finland.

In the end, the differences in signal delay between having a beacon or using GPS signal is negligible. The more problematic issue is that GPS is not 100% accurate. On the official US government website, it says that the GPS in a smartphone is typically accurate within a 4.9-meter radius, but it can worsen a great deal due to buildings or other blockages (GPS.GOV, 2020). For a bus approaching a traffic light, this would mean that the variability is 9.8 metres because the request can be triggered both 4.9 metres too early or late. The new Galileo system is said to have an accuracy of one to two metres while in motion, and under 1 meter when static (European GNSS Agency, n.d.). It should also be acknowledged that today smartphones use the telephone network, map data and Wi-Fi to verify the location, which makes it more accurate than using only GPS.

The signal delay discussed earlier also means that the data the traffic light control box receives becomes even more deceiving due to the GPS error. From the expert interviews, we know that the bus to Mattersoft central the delay is about 0.5 seconds, and the delay from Swarco central to the control box is 400 milliseconds. The only delay that is unknown is the delay between Mattersoft central and Swarco central, but it was assumed to be 0.5 seconds in accordance with the other signal delays. The total delay would then be around 1.5 seconds in total. At 50 km/h, the bus travels 13.89 metres per second. Thus, with a 1.5-second signal delay, the control box will believe that the bus is 20.83 metres farther away from the intersection than it is. Then, there is the inaccuracy, which can either remove or add 4.9 metres. This would create an error between 15.93-25.73 metres. Of course, this is just an estimate, but it gives a perspective of how large the error can be.

In the end, the signal delay is not too troublesome because it is possible to move the detector backwards to counter the delay. One must estimate how far the bus will travel on that specific intersection within the signal delay window. The GPS inaccuracy is a bit more complicated because its effect is in both directions. One solution could be to implement the two detector points, similarly to the iBus system used in London. It is also essential to consider this when marking the detector points. Because of the inaccuracies, the detector points should not be made too small. Otherwise, the bus could accidentally miss the detector point. However, Table 1, in the literature chapter, shows that several other cities have implemented a GPS based TSP system without problems.

4.2 SYVARI

The first TSP system was implemented in Finland over 40 years ago, but implementing a TSP system in Finland was a challenging task up until 2009 (Salonen, 2010). This was because there was no standard strategy within the country. This meant that in each city, the urban planning department had to design the system themselves, which was time-consuming and challenging. The initiative to build a standard traffic light configuration method in Finland began in Turku 2006. This system was called SYVARI, and while the system itself was built to configure the traffic light settings and how the cycle works, it also made it effortless to implement a TSP system. The project was officially finished in 2009.

The objective was to find a solution that could be implemented with ease. Therefore, a new traffic light control software was built. One of the most important factors was to avoid the logic in the traffic light controller from becoming something that could be implemented in just a few specialised companies to prevent monopolies.

The first version of SYVARI finished in 2007, and version 1.5 was released 1.5 years later. The 1.5 version included the possibility to grant priority green phases for buses, even if there is a regular green ongoing for another direction. Version 2.0 was released in 2010 and is today the newest version. The latest version included more optimised solutions for transit signal priority.

The initiative gained most of its traction through the JENKA project, whose objective was to implement a TSP system in every city in Finland. The project included further developing SYVARI and compiling a manual about how to implement the system. Today the manual has a step for step guide on how to implement the system with explanations on how the strategies can be modified.

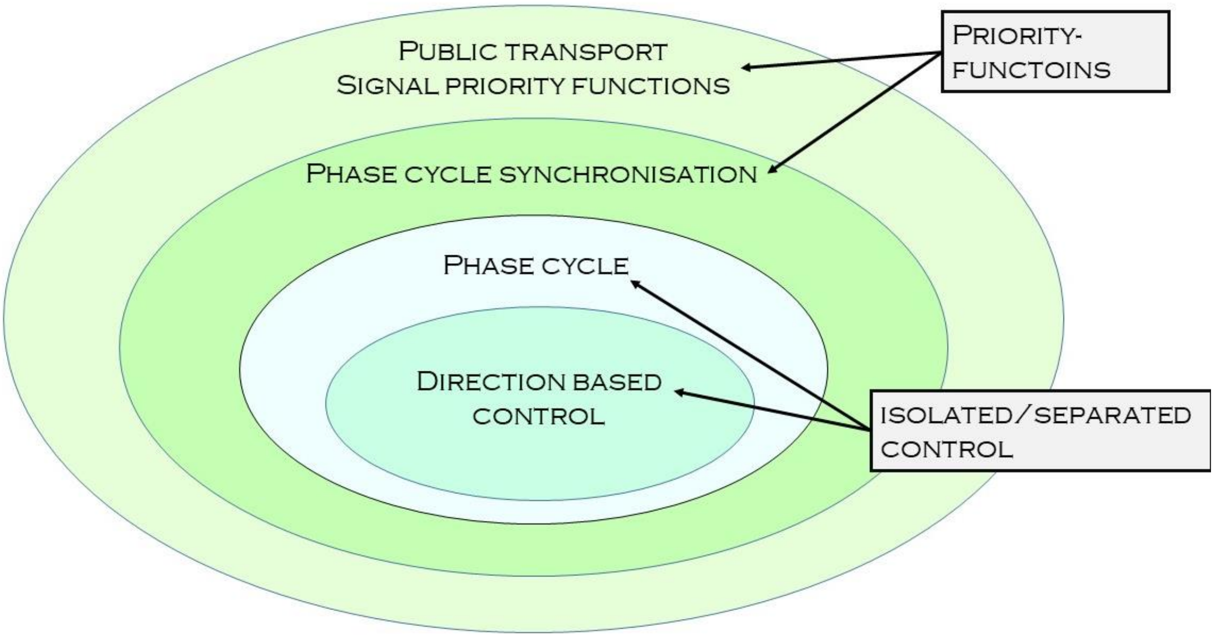

Figure 4: SYVARI structure

How SYVARI works together with the phase cycle is shown in Figure 4. The two inner circles are part of the normal phase cycle, which decides the current state of the lights when there is no priority request. Phase cycle synchronisation synchronises the phase cycle back to its normal state. The phase cycle can be off-sync because of a priority request or because the previous phase was longer or shorter than normal.

For example, when a new priority request is sent, the controller stops the phase cycle from following the normal state. During the priority request, the normal phase cycle continues according to its normal phase, even if there is a priority request. This means that if there is a 10 second extension, then the traffic lights will be behind the normal cycle by 10 seconds. The system will then automatically calibrate itself back to the normal cycle by either shortening or lengthening the traffic light phases. The phase cycle synchronisation handles this process. The system works like this to create green waves, which enables cars to drive through several intersections without stopping. Therefore, the normal phase cycle must be coordinated with the upcoming intersections.

The different synchronisation functions are synchronisation extension, synchronisation reset, synchronisation delay. Synchronisation extension, synchronises the cycle to its normal phase by extending a light within the phase cycle. Synchronisation reset, resets the current phase by accelerating the speed of the phase cycle and shortening the green light phases in the cycle. Synchronisation delay will delay the phase cycle until it is back to its normal state.

This model differs a great deal of how priority requests were initially configured in Finland, due to SYVARI continually working together with the normal phase cycle. With this model, the realisation of the priority requests depends to a high degree on the settings of the normal phase cycle, which makes it easier to implement a TSP system.

Even if there is a priority request in the normal cycle, the controller will ensure that every direction will gain a green phase within each cycle. This means that the next phase cycle is not able to begin before the previous phase cycle has finished. This is to ensure that other directions will not become congested because of priority requests.

In SYVARI the traffic lights are configured separately for each direction, which is called split phasing. This allows the traffic phases to have different lengths depending on the direction. Therefore, lanes can have different phase lengths even if they became green at the same time. Figure 32 in the appendix shows an example of the configuration file. The old system used literal phases for traffic light planning, instead of having the directions defined individually. This meant that once the phase ended, all green lights within that phase switched to red. With split-phasing, it is possible to implement active green and passive green. It also makes the traffic lights more optimised to changes in the traffic amount.

Active green

In SYVARI, a normal defined green phase in one direction is called active green. Active greens cannot be shortened below their minimum green time value. Therefore, it will stay green at the minimum for the minimum value, but less than the maximum value. The duration is dependent on the traffic amount for that direction. The light can only remain green for longer than the maximum time if there is an extension request.

Passive green

Passive greens are potential greens that can coexist while another direction has an active green. For example, if there is an active green for the direction going west, then the opposite direction can have a passive green as they are not conflicting directions.

Passive greens will stay green until there is a conflicting active green. Therefore, it is crucial to define passive greens within SYVARI and to set parameters for when passive green is not permitted. Otherwise, it will stay green or red throughout the cycle.

Safety times

Safety time is the time between one light turning red and the next light turning green. Because of safety reasons, there is always a minimum number of time that all conflicting directions are red to empty the intersection before the next green light. It should be noted that the yellow light before turning green is included in the safety time, but the length of the yellow phase can differ between 3-5 seconds depending on the speed limit and size of the intersection.

All the traffic light configurations in SYVARI are defined based on a single lane. This means that each lane must know the safety time to every conflicting lane. This includes the zebra crosswalks.

The safety time is calculated based on the speed limit and the distance that the vehicle must clear before there is no crash risk with the conflicting lane. Figure 38 in the appendix shows how the safety time is calculated.

Offsets

Offsets are used to create green waves between intersections. For example, if the first intersection turns green at time 0, then the next intersection will turn green in the same direction a couple of seconds later to such extent that the first intersection's traffic can reach the second intersection in time for its green phase. The offsets are calculated based on what the speed limit is and how long the distance between the two intersections is.

4.2.1 Configuring priority requests

SYVARI has implemented different priority functions in the system. These functions are green light extension, red light shortening (shorten phase length), extra green and green rotation. When planning the priorities in SYVARI, the traffic planner does not need to consider if the priorities will fit in the normal cycle as the software can adapt in such a manner that all the priorities fit. Even if there are multiple priority request present, the software ensures that each traffic light gains green for their minimum time. This means that if there happen to be several priority requests within a single phase cycle, then the cycle time will be longer than the normal cycle time. However, once all the priority requests are processed, the phase cycle will start to synchronise itself back to its normal state. In SYVARI, there is no maximum time of how delayed the phase cycle can be, however, the minimum cycle time should be set at least 15 seconds shorter than the normal phase cycle to make the synchronisation effective (Salonen, 2010).

Restricting the maximum amount of priority request that can sequentially appear is not encouraged in SYVARI, even if the priority requests can cause the cycle to be entirely off sync with the normal cycle. This is because a priority request causes less disruption to the general traffic if the cycle is already off sync compared to a new priority request in a new cycle that is in sync with the normal phase cycle. If there is a need to add some restriction to the number of subsequent priority request, then it is recommended that this is done by instead adding a minimum number of normal cycle phases before the controller accepts a new priority request.

Guaranteed maximum time

Guaranteed maximum time defines the minimum time the light can be green within a single cycle. Guaranteed maximum time exists to prevent unexpectedly or dangerously short green phases from existing. It also ensures that there is adequate time reserved in each direction such that enough vehicles are allowed to pass the intersection, without oversaturating the intersection in a specific direction. The guaranteed maximum time should be no less than the minimum pedestrian green light time in the same direction.

Synchronisation maximum time

Synchronisation maximum time defines the maximum length for the active green in a given direction. Synchronisation maximum time includes all extensions except for priority requests. For example, if the lane is congested, then the direction will have a green light for the duration of the maximum synchronisation time. However, this is dependent on the traffic amount in the other directions.

Maximum priority time

Maximum priority time is the parameter that defines how long the bus priority extension can maximally be. Maximum priority time is always defined longer than maximum synchronisation time such that the difference is at the least as long as the time it takes for the bus to travel from the request trigger point to the intersection. Maximum priority time explains how long the extension is allowed to be after the maximum synchronisation time has ended.

This is the essential vocabulary to understand when working with SYVARI. Without them, the system will be impossible to understand. Next, the different priority types that exist in SYVARI will be discussed and how they work. The priority functions in SYVARI are extension, red light shortening, extra green and green rotation.

Green light extension

Green light extension means that the traffic light will extend the time the light stays green. It is possible to alter the maximum length, to minimise the disruption from a green light extension. This can be done by changing the maximum priority time value.

Red light shortening

Sometimes the bus is not able to catch the green light and will then have to wait until the next cycle. One of the solutions for this is to shorten the length of the current phases such that the bus can continue more quickly. In SYVARI, green phases can be shortened to their guaranteed maximum time.

Extra green

SYVARI also has a function which allows it to add an extra green for buses. This works in the system by adding an extra phase to the normal cycle, which gives the bus a green phase. For extra green, it is vital to set up suitable parameters because it can potentially reduce the disruption by a great deal. For example, in Figure 5, the number of phases does not change even if there is extra green. This is because of the passive green the traffic, which allows non-colluding lanes to gain green even during the priority. Even if it easy to implement extra green to SYVARI, it is crucial to carefully consider the placement of the extra green because it could potentially reduce the traffic flow by a great deal.

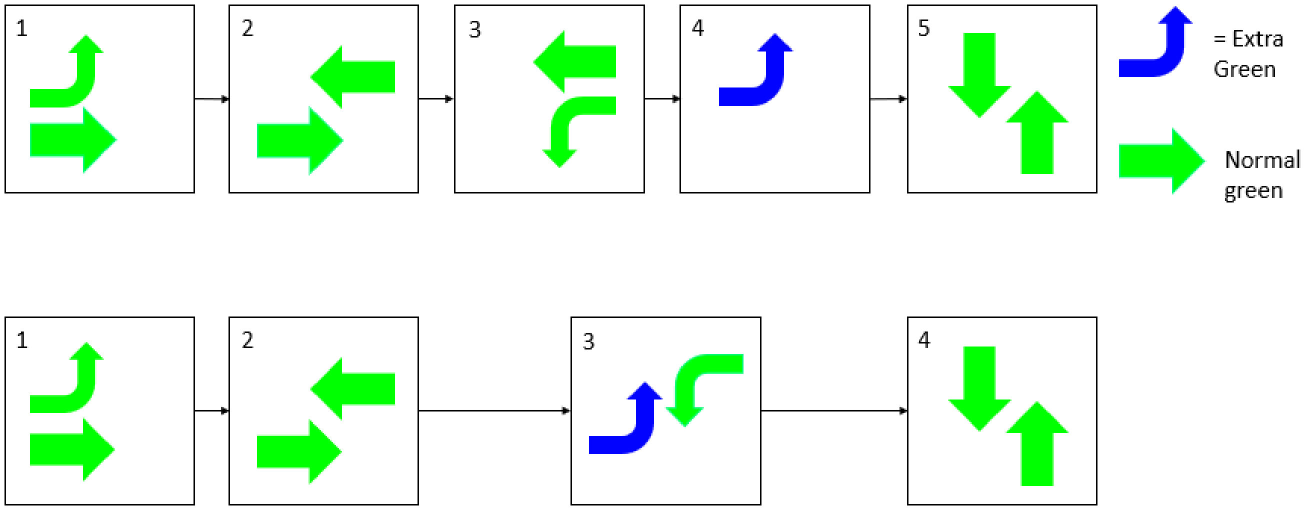

Figure 5: Extra green phases

These are the different priority types that exist within SYVARI. What and how the priorities are applied depends on the state of the intersection. One of the inbuilt functions in SYVARI is green rotation, which allows the phases to rotate their position in the cycle. This function can be used for both an extra green or to move the bus green phase earlier in the cycle. The rotation of green phases must be defined separately in the configuration file. Otherwise, SYVARI does not know which green phases are allowed to rotate. This is done with the hollow blue arrow found in Figure 6.

An example of the green rotation function is shown in Figure 6. There the third square is a priority request for the south direction. Initially, this direction would gain green after the north direction. However, because of the priority request, these two phases now rotate, and therefore the south direction will gain its green phase before the north direction. The software will always try to find the most optimal way to rotate the phases such that the normal cycle is minimally disturbed.

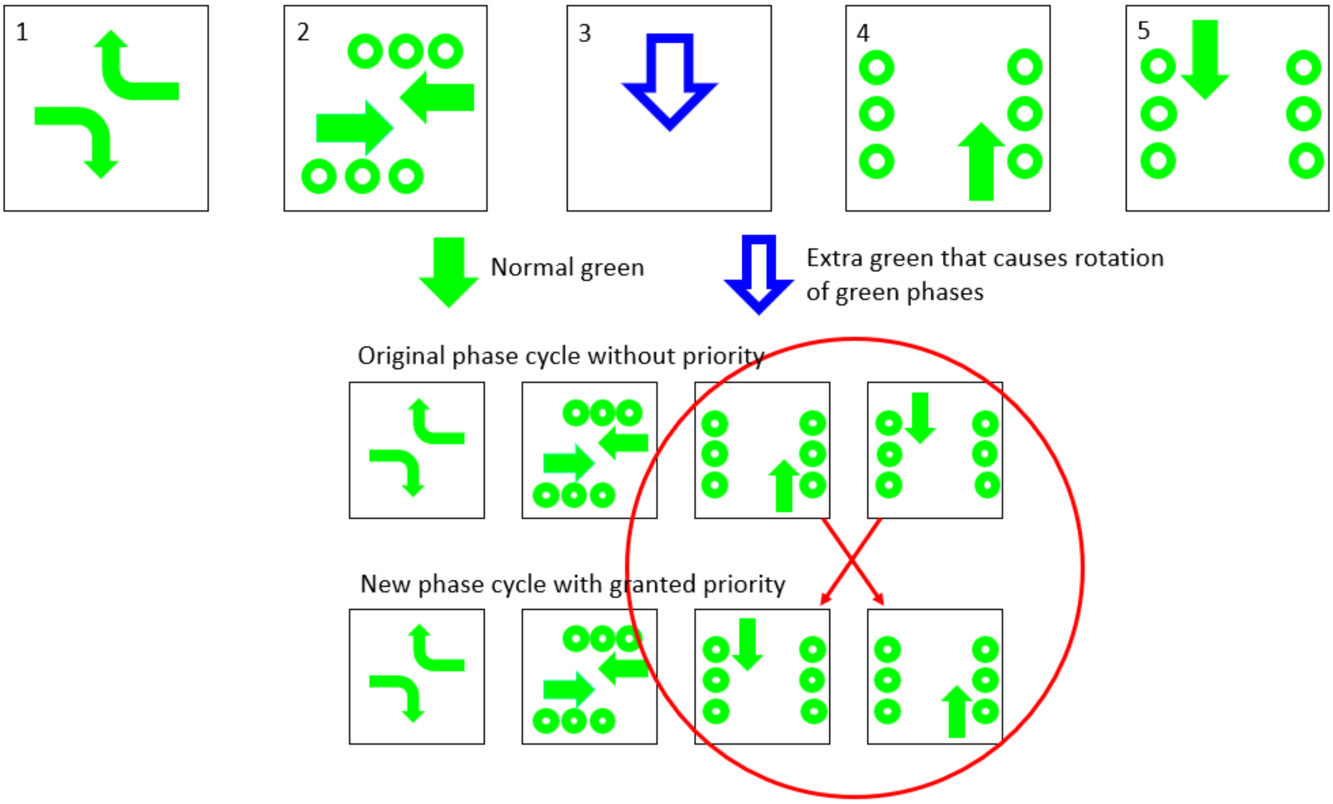

Figure 6: Rotation of green phases source

5. Methodology

In this study, a quantitative data study will be conducted with the help of simulations. A quantitative data analysis is great at answering "what" or "how" questions because of the quantitative data (Goertzen, 2017). Quantitative data analysis consists of the following steps:

1. The first step in quantitative analysis is collecting the data. The data can be collected from both primary and secondary sources. The sources should be investigated to ensure that they are reliable.
2. Recognize the type of data that is being handled: nominal, ordinal, interval or ratio. The data should then be familiarised and presented with various figures or tables.
3. Present the data with a descriptive statistic such as, mean, median, mode, minimum and maximum values, percentages and frequency. This step works as a validation process because outliers will be found.
4. The relationships, trends and differences between multiple samples should be analysed. This allows for hypothesis testing and generalising the results. Inferential statistics that can be used are correlation, analysis of variance and regression analysis.

In qualitative research, the studied material is non-numeric, and interviews and observations are often used to answer the research question. In a qualitative analysis, one tries to understand what or why people believe or do something. The differences between quantitative and qualitative research will be described in Table 2 below (Barnham, 2015; Moore, 2016):

Table 2: Comparison of quantitative and qualitative research

	QUANTITATIVE	**QUALITATIVE**
OBJECTIVE	To answer what consumers do or think by creating a copy of the real world in data format	To gain an understanding of why and how consumers think by analysing beyond the surface and responses.
RESEARCH TYPE	Exploratory	Conclusive
METHOD	Structured questionnaires	Open-unstructured questions
DATA	Quantitative data	Unmeasurable data

In this study, there will be a couple of expert interviews about the system in Turku. The interviews will give an understanding of how the simulation should be built. The remaining data will be in a quantitative format from bus logs or induction loops that are placed throughout the route.

5.1 Simulation

As stated previously, for this research paper, a microscopic traffic simulation will be built within SUMO. The traffic priority system will then be implemented to the simulation, whereafter the results from the simulation will be analysed and studied.

A simulation is a model of the system. The model is used to predict the outcome of the real system under various conditions without testing the variables in the real world (Barcelo, 2010). The simulator must have an understanding of how the system works and have assumptions of the mathematical and logical relationships within the system. Because simulations are built on assumptions, one must accept and realize that the assumptions will have some effects on the results.

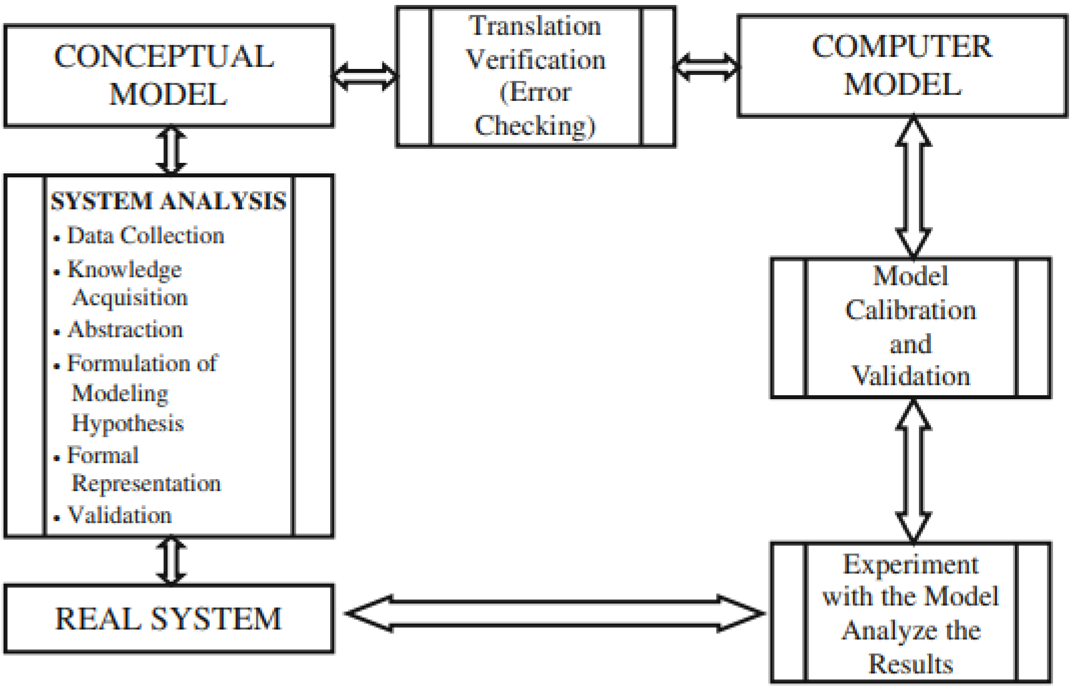

Figure 7: Methodological steps for building a model

Note: Reprinted from Fundamentals of Traffic Simulation (p. 5) by J. Barcelo, 2010, International Series in Operations Research & Management Science, Copyright 2010 by Springer Science and Business Media LLC.

In Figure 7, the steps for building simulations is shown. The first step is to analyse the system and the corresponding data (Barcelo, 2010). A conceptual model should be designed after the input data is collected. A conceptual model is a prototype of the final simulation, which is built in the simulators mind or by drawing it on a paper with the relationships marked. Additionally, the logic between the different variables is conceptualized. The conceptual model will help the simulator to understand the system and streamline the building process of the simulation. The model should be verified by discussing with experts.

Next, the computer simulation will be built, which is often done in small segments. This makes it possible to test that everything works correctly at each step, which is much easier when there are just a few variables. Once the simulation is complete, the calibration and validation process begins. Since the simulation is built on assumptions about the real system, the calibration and validation of the simulation is invaluable. A model is valid when it faithfully represents the real system. This can be confirmed by testing that the output differences between the real system and the simulation are within the tolerable significance level (Miguel et al., 2004). Once the model is confirmed to be an accurate representation of the real system, the experiments can be tested in the simulation

5.1.1 Simulation software

The simulator must decide which simulation program to use before the simulation can be built. This is a crucial decision because there are many different traffic simulation software, and they have different functions depending on their purpose. For example, PTV group, who owns the most popular simulation software Vissim, offers in total 17 different traffic simulation software tools (PTV GROUP, n.d.).

The simulator must decide the scale of the simulation to choose the appropriate simulation software. The scale is not about the area of the simulation. Instead, it defines to what extent the variables can be changed within the simulation. Traffic simulations can be divided into four categories: microscopic, macroscopic, mesoscopic, submicroscopic. Microscopic and Macroscopic are the most common definitions.

In a microscopic simulation, the simulator is able to input vehicle data to a single vehicle and change the driving characteristics independently from the other drivers (Helbing, 2008). This allows the simulator to create a more random and realistic simulation. The simulation software can give outputs about single vehicles. Microscopic simulations are used for detailed studies where the small details are important. Because there are many different variables in the simulations, a microscopic simulation also requires a great deal of CPU power. To the contrary, macroscopic simulations are excellent at simulating large areas. In a macroscopic simulation, it is not possible to design all vehicles individually; therefore, averages are used for the variables. Macroscopic simulations are useful for understanding the overall effects and are often used to analyse traffic densities. Mesoscopic simulations are a mixture of microscopic and macroscopic simulations. In a submicroscopic simulation, the vehicle functions are also explicitly simulated, such as gear shifting.

In this study, a microscopic simulation software will be used since the route is so small that the extra CPU power required will not be a problem (Lopez et al., 2018). Next, some of the potential simulation software will be reviewed.

Vissim

Vissim was the most popular simulation program to use in scientific papers when studying TSP systems. This could be because it has many functions built in it. For example, in Vissim, it is possible to connect the actual traffic light controller to the simulations such that the traffic lights are identical to the real world. Vissim also has a function for transport signal priority, which can be used to test bus priority systems. Vissim offers several tools to make the building of a simulation simple, such as using a map as a layer and then drawing the road network on the map. Vissim also has a 3D feature, which does not exist in the other simulation software presented in this paper. However, the wide variety of tools makes the software more complicated to use. The drawback of Vissim is that it is expensive compared to the other simulation software.

SUMO

SUMO, which stands for Simulation of Urban Mobility, is a freely available open-software traffic simulator. In the papers that were studied for this research, SUMO was one of the most popular simulation software after Vissim. In total SUMO is downloaded over 35 000 times per year (Arellano & Mahgoub, 2013). A reference publication of SUMO was published in 2012, which is an excellent tool for understanding the basics of what can be built in SUMO. (Krajzewicz et al., 2012). Since then, SUMO has gotten several updates and new tools to simplify the process of building simulations or to test new technologies. SUMO is especially prevalent in Vehicular ad hoc networks (VANET), where the vehicles share information (Lim et al., 2017). SUMO being open-source means that the simulator can change and modify almost everything in the software and even build entirely new functions within the software. This can often be a deciding factor for someone who is about to build and test a new traffic system. SUMO is also able to utilize many other tools that exist in the simulation market that have inbuilt tools to integrate them into the SUMO simulation environment.

Synchro

Synchro is a simulation software that has previously been used in the traffic planning department in Turku (Salonen, 2020). Synchro is also a licenced software, but it is much less expensive than Vissim. When searching for scientific papers that include Synchro and traffic simulation, the result shows that synchro is often used to plan the optimal traffic signal timings. Some papers even used synchro just for the optimisation of the traffic signals and then used the outputs in another traffic simulation software (Singh et al., 2017; Stevanovic & Martin, 2006; Udomsilp et al., 2017). The lack of vehicle simulation capabilities in the software makes it inferior to the other simulation software for this study.

MATsim

MATsim is a microscopic multi-agent transport simulation software, which is open-source (Horni et al., 2016). The software is built to simulate large scale logistic simulations. The software has been used to plan new bus routes in other studies (von Flavio, 2017). Because it is open software, custom tools can be built in the programming language Java.

SUMO was chosen as the simulation software because it is most suitable for this study. One of the research questions is whether simulations can be used more within the Turku city, and for them, the cost of the software is a significant factor. Furthermore, SUMO offers almost all the functions that the other software have. It has the advantage of being open-software, which means that the simulation can be programmed according to how the system in Finland works. Additionally, according to a review done by Ejercito et al. (2018), the differences between SUMO and Vissim are not significant, with both software having some unique features. Synchro was not chosen because Turku city already uses the software and because it is designed for traffic signal planning. The other open-source software was MATsim, but it was not chosen because SUMO was more popular and had been used several times in similar studies.

5.2. Data collection

The data for this study will be collected from various sources from Turku city. The traffic data for the simulation will be collected from induction loops placed at intersections throughout the route. These induction loops are used for the adaptive traffic lights, but the vehicle data is collected simultaneously. The traffic planning department at Turku handles the traffic data.

The bus data will be retrieved from the bus department in Turku (Föli). They can retrieve the bus logs with MySQL workbench. However, only the latest four months of bus data is stored in the database because it is too memory intensive, but it should be enough to create a realistic simulation.

Traffic light settings will be retrieved from the traffic planning department. For these, there are existing excel files, which contains the configurations for each intersection. Additionally, the traffic planning department will consult on how to make modifications to the simulation for functions that cannot be included. Furthermore, the company that handles the traffic control system (Swarco) in Turku will be interviewed to verify that the system is built correctly and to gain knowledge about how the system works. For example, there might be delays in the system that must be considered in the simulation as well.

6. Data collection and data analysis

In this chapter, the data collection for this study will be reviewed in more detail. Afterwards, the data will be studied and analysed. The data discussed and analysed is bus data from the bus logs and vehicle data from detectors placed at intersections in the route. Additionally, the collection of map data and traffic light data will be briefly discussed.

6.1 Bus Data

With the new GPS system installed in Turku buses, the buses log data on every bus stop. From this data, it is possible to calculate how long it takes for a bus to travel from point A to point B, with the different points being bus stops. The data includes the arriving time and departure time from a bus stop, which can be used to calculate the duration of the bus stop.

Only the last four months of bus logs can be stored at Turku due to data server capacity. Therefore, the data used in this study is from 22.10.2019-19.2.2020. This data contains logs from the entire day, but the rush hours are most interesting because that is when the TSP system will bring the most benefits. The traffic flow outside rush hours is smooth, which means that there is less dwell time at traffic lights. Therefore, the results from implementing a TSP system are less beneficial.

A small segment of the bus data logs is shown in Table 3: *Bus log*. The bus stop ID column shows is the bus stop the data were logged. The pattern code explains which route the bus is on. For bus 99, there are four different routes, which have a different ending or starting location. The last number in the pattern code (1 or 2) explains which way the bus is going. Stop type tells us whether the bus stopped at the bus stop or not. If the value is zero, then the bus stopped there, and if it is 4, then the bus drove past it. Vehicle ID is a unique ID number of the bus and nominal arrival time is the scheduled arrival time at the bus stop. The actual arrival time is when the bus arrived in reality. All columns that are in time format are measured in seconds in the bus logs. For example, the actual arrival time of the first bus is 35312, which is 09:48:32 in standard date-time format.

Table 3: Bus log

Line code	Pattern code	Date	Bus stop ID	Stop type	Vehicle ID	Actual arrival time	Nominal arrival time	Actual departure time
99	99I2	22.10.2019 0:00	454	0	200066	35312	35280	35322
99	99I2	22.10.2019 0:00	961	0	200066	41950	42090	41960
99	99I2	22.10.2019 0:00	1636	0	200066	45364	45420	45409
99	99I2	22.10.2019 0:00	1624	0	200066	45109	45160	45122
99	99I2	22.10.2019 0:00	839	0	200066	54047	54060	54096
99	99IP1	22.10.2019 0:00	1442	4	200066	54713	54560	54713

6.1.1 Bus stop time

The data were first cleaned with pandas in python and then converted to a boxplot. As stated earlier, this study focuses on rush hours because that is when the buses are most likely to be stuck at traffic lights. Therefore, the bus stop logs were filtered to only included hours 6:45-9:00 and 14:00-17:00. These are the hours that are most congested based on the traffic data.

Bus stops 839 and 1559 were also removed from Figure 8 and Figure 9 because they are slack stops, which means that they are bound to have a great deal of variance and outliers. Bus stop 839 and 1559 can be found in Figure 30 in the appendices. Slack stops have many outliers because if the bus is early, it must stop and wait until it is on schedule again on these stops. This is not a problem in the simulation because the route begins at bus stop 839 and ends at bus stop 1559, and therefore they are excluded from the simulation. Additionally, some other extreme values were removed because they are presumably because of technological failures. Bus stop 961 had an outlier with a value of 391 (6.51 minutes) during rush hours. Bus stop 440 also had an outlier with the value 238 (3.96 minutes).

In the original data, many of the bus stops that had a duration value of less than five seconds. This is not realistic because opening the door, letting passengers in and then closing the door requires more than five seconds. The short bus stops could generate because of inaccuracies in the GPS signal. Therefore, these were removed from Figure 8 and Figure 9. The minimal bus stop duration was set to five seconds. Furthermore, buses that drove past the bus stop were removed because they were the majority in most bus stops. Therefore, the median value would be zero, which does not produce any meaningful information. In the simulation data, all the values are included.

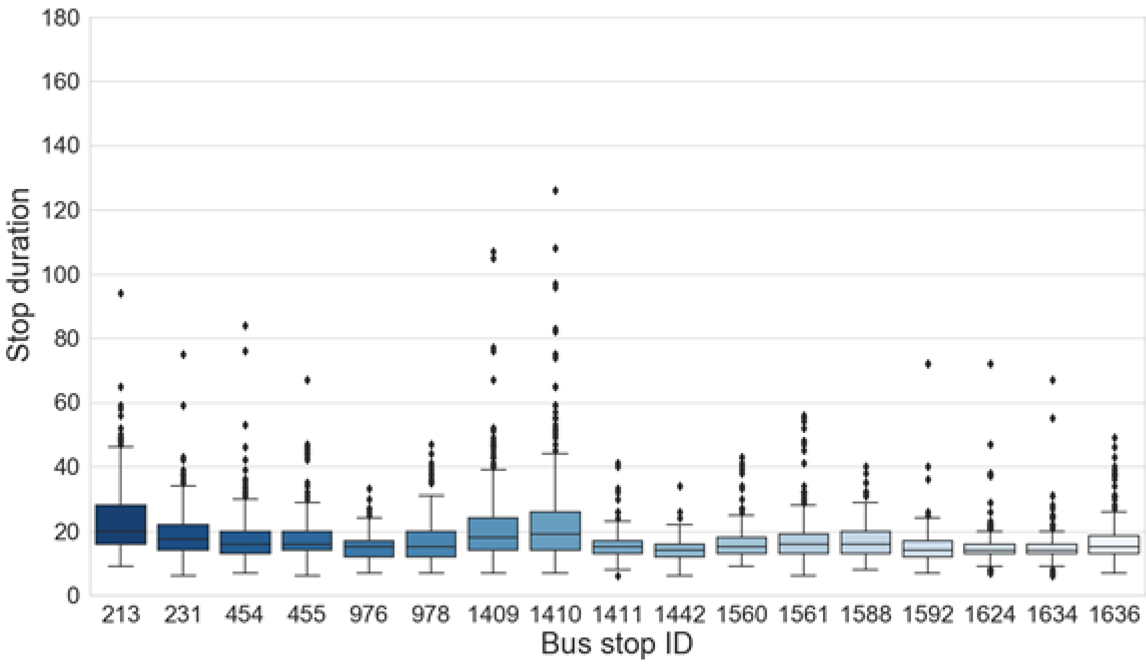

Figure 8: Bus stop duration during rush hours (east)

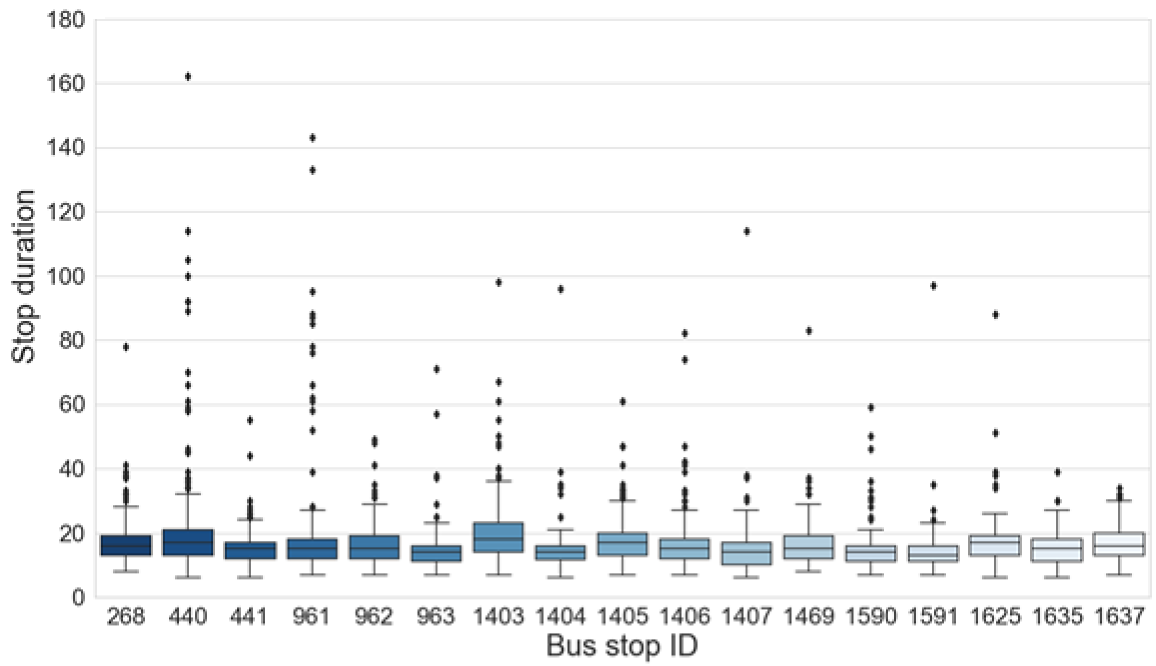

Figure 9: Bus stop duration during rush hours (west)

From Figure 8 and Figure 9, it can be concluded that every bus stop has a median stop time of under 20 seconds. Additionally, the variances seem to differ a great deal between different bus stops. However, the median length is almost the same for all bus stops, with a difference of less than five seconds between the maximum and the minimum median value. This is not surprising because in most cases, there will only be a couple of person per bus stop, thus the median values become similar.

However, the maximums differ a great deal depending on the bus stop. For some, it is less than five seconds more than the median bus stop time, while for others it is more than two times higher compared to the median. For example, bus stop 213, 1409 and 1410 have a much higher upper maximum compared to the rest in Figure 8. This means that those stops have a more inconsistent bus stop time, and therefore, it becomes more challenging to plan the bus schedule.

Figure 8 and Figure 9 show that the west direction generally has more outliers compared to the east. The variance seems higher for buses going East according to the maximum values. However, according to Table 4, the variance is higher for west-bound bus stops. This is mainly because the east direction has a couple of bus stops with extremely low variance.

For scheduled buses, it would be best if the variance between stop time on a specific bus stop would be minimal. This is impossible to achieve because there will always be different amounts of people boarding the bus. One solution would be to have designed time at each bus stop, but this would mean that there is added slack time to each bus stop which increases the route duration.

Table 4: Bus stop duration statistics

	East morning	West Morning	East evening	West Evening
count	2108.0	1465.0	2982.0	2519.0
mean	19.84	16.93	17.01	16.99
std	9.99	6.09	7.52	11.60
min	6.0	6.0	6.0	6.0
25 %	14.0	14.0	13.0	12.0
50 %	17.0	16.0	15.0	15.0
75 %	22.0	19.0	19.0	19.0
max	126.0	96.0	96.0	238.0

6.1.2 Bus travel time

With the help of the scheduled arrival time and pattern code, the correct rows can be coupled to calculate the bus travel time. West-bound means that the start-stop is 1637, and the ending stop is 1559. East-bound means that the start-stop is 839, and the ending stop is 1636.

Kernel density estimation plots were then constructed from the route duration data. Kernel density plots are similar to regular histograms, but they use algorithms to smooth the curve in case there are missing data points within the x-axis (Węglarczyk, 2018).

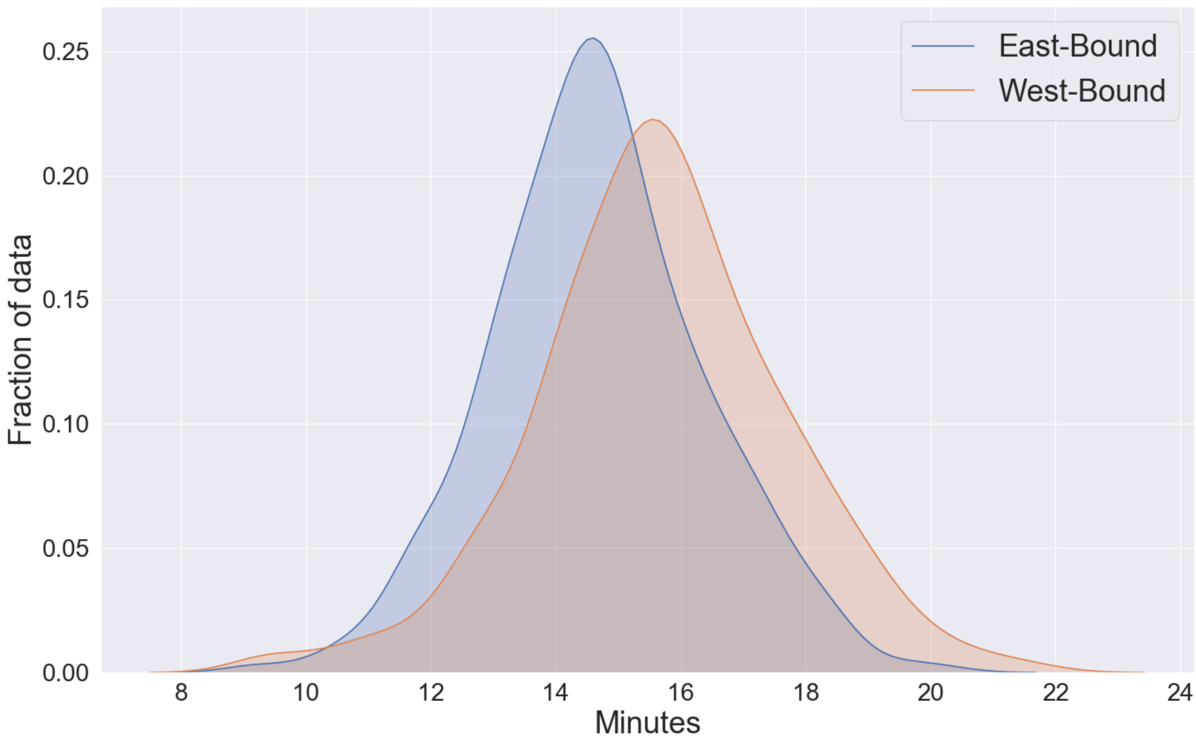

Figure 10: Kernel density plot of bus route times

Figure 10 shows that both directions have their peak at roughly the same time. The peak shows the most common duration of the bus route, but the median duration does not have to be on the peak.

In the figure, the west-bound is leaning more to the right compared to east-bound, which means that the East-bound buses are on average faster compared to the West-bound buses. West-Bound buses are also more spread out and do not peak as high as East-Bound buses, which means that there is more variance in travel times for the West-bound buses. One explanation for this could be that the bus stop durations are longer for the West-bound traffic, but according to Table 4, the difference in the mean times does not make up for the difference in route duration. Therefore, the morning and evening rush hours were divided into two separate figures to understand the west bus is slower than the east bus.

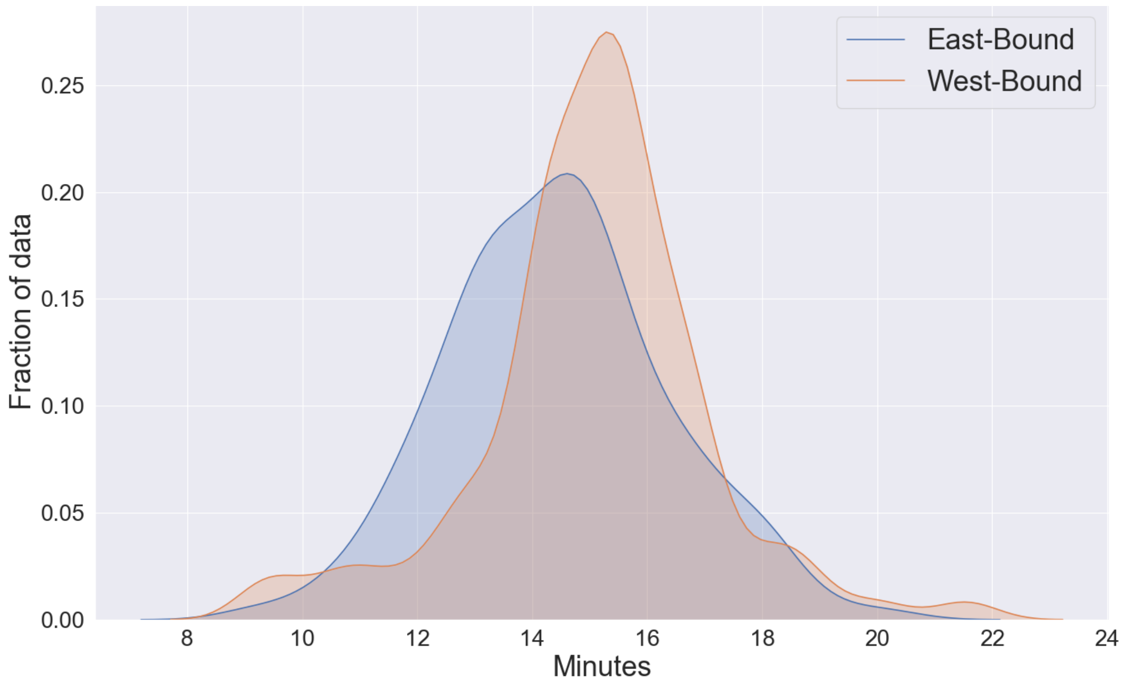

Figure 11: Kernel density plot of bus route times (morning)

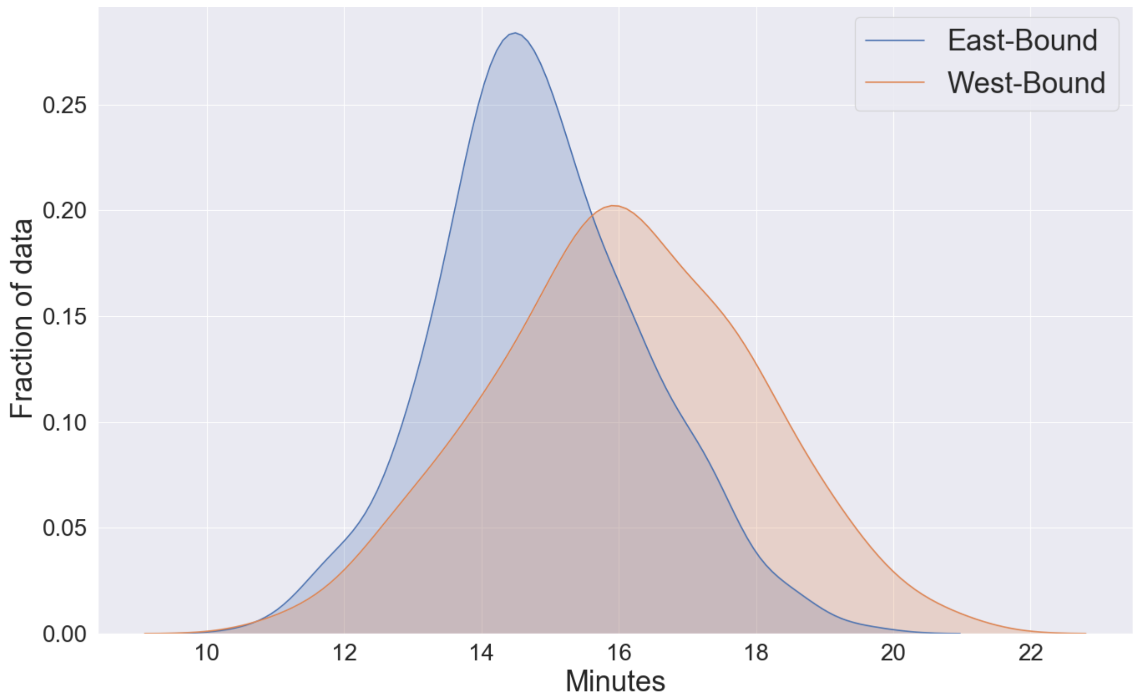

Figure 12: Kernel density plot of bus route times (evening)

In Figure 11 and Figure 12, the bus route time during rush hours is divided into two graphs showing both the morning and evening rush hours. Unexpectedly, the west-bound traffic is slower during both the morning and the evening. This is surprising because congestion often originates from people going to work and leaving work, which means that one direction is faster in the morning and slower in the evening. The peaks in the figures hint that in the morning, the east experiences more of the morning traffic, while in the evening, the west is more congested. This is based on how the peaks of each direction move and on the variance shown in the figures. The variance should be higher during rush hours because sometimes the bus is lucky and can pass within the first green within an intersection and other times it must wait until the next green. Furthermore, in the evening the peak for east moves to the left and for west it moves to the right.

There is also no correlation between the route-times and the bus-stops duration. According to Table 4, bus stops for buses going east are slower in the morning and almost identical in the evening. A longer bus-stop duration means that more passengers are going in that direction. The different route durations must therefore be because of congestion or because of how the traffic lights are configured.

Table 5: Bus route times statistics

	East		West	
	Morning	Evening	Morning	Evening
count	245	293	182	284
mean	14,43	14,87	15,06	16,04
std	1,92	1,48	2,01	1,90
min	9,05	10,83	9,22	11,05
25 %	13,18	13,92	14,20	14,75
50 %	14,45	14,72	15,21	16,07
75 %	15,55	15,83	16,02	17,35
max	20,28	19,53	21,72	20,85

In table 5, the statistics for both east and west bus route times are shown. The east direction is always faster than the west direction in every metric. Even the standard deviation is higher for the west bus. The route duration is mostly affected by the number of other vehicles. Therefore, the detector data will be analysed next to verify if that is the reason for the west direction being slower compared to the east direction.

6.2 Vehicle data

Traffic data were retrieved from induction loops, which are located on the incoming lanes in traffic light equipped intersections. Therefore, there is no data on how many cars will continue to a specific lane, but this can be guessed based on the data from the incoming lanes. The problem is that in many intersections the detector is located before the lane division or that cars can continue in multiple directions from a single lane. An example of this is shown in Figure 13. There the detectors 108L and 208L are both situated before the lane division, thus it is impossible to know how many cars will continue straight or turn in either direction. Therefore, some of the vehicle numbers had to be estimated for the simulation. The estimation is based on the number of cars in the next intersection and the location of the intersection. This is a satisfactory solution because the only roads between intersections without detectors are non-traffic light equipped intersections. These are ignored in the simulation because the bus will always have priority in them and because they cannot be equipped with a TSP system. Even if the turning rates are estimates, the vehicle number in both west and east direction is correct as they are based on the detector data.

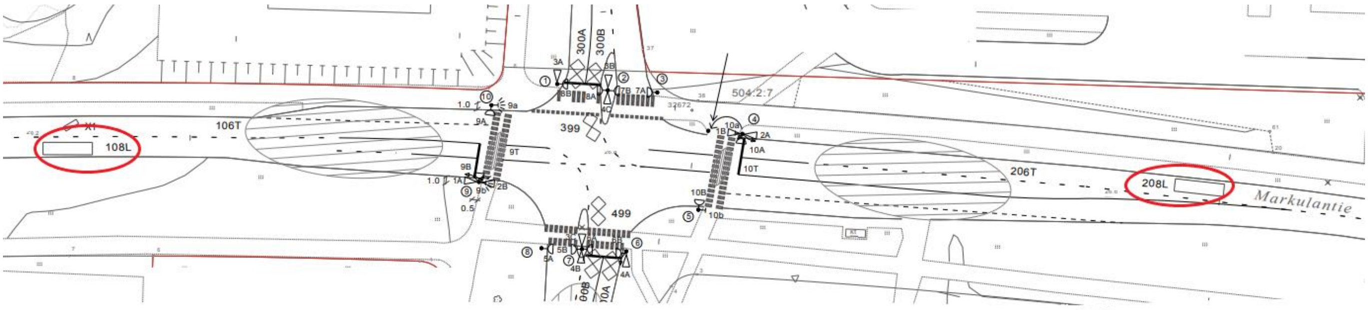

Figure 13: Detectors in intersection 336

The induction loop data retrieval is incredibly time-consuming. Therefore, the number of retrieved induction loops had to be limited. Only data from lanes that continue in the same direction as bus route 99 was retrieved. The exceptions were intersection 405 and 409, which are the most congested intersections in the simulation. For them, every detector's data were required to analyse the effect on the crossing traffic.

The induction loops work by having a magnet that triggers when a vehicle is above the detector. One problem is that these detectors are not always 100% reliable, which was recognised when the data were analysed. For example, in intersection 334, one of the lanes incoming from the west, only had 14 vehicles per hour, while the second lane had 228 vehicles per hour. These two were both going in the same direction, thus it is unreasonable that the left lane would have more than 16 times more vehicles. The probable explanation is that the detector is not calibrated correctly. Therefore, it only recognizes large vehicles such as buses and trucks, which would explain the low number of vehicles.

6.2.1 Vehicle data

Next, the detector data that was collected will be briefly discussed. Some intersections were better equipped with detectors than others. Therefore, some intersection had much more vehicle data compared to others.

Figure 14: Intersection numbering

In Figure 14, the numbering of the intersections can be seen. These intersection numbers will be used when referring to specific intersections below. Next, what data from each intersection was retrieved will be explained.

<u>333 – Markulantie/Viilarintie</u>

At intersection 333, data from all lanes incoming from the west was retrieved. The data showed how many cars are about to turn right, left and continue straight. On the east side, the detectors were located before the lane division. Therefore, only the data on how many cars in total are incoming from that direction was known.

<u>334 – Markulantie/Nuijamaankatu</u>

In this intersection, only data from Markulantie, which is the road bus 99 follows, was collected. The detectors are also placed before the lane division and one of the detectors on the west-side was defective. This was concluded based on the fact that it showed that the median number of cars driving on the right-side lane per hour was 12, while 228 cars were driving on the left side lane.

<u>336 – Markulantie/Eskonkatu</u>

Only data from the detectors following the bus route was retrieved. All detectors were placed before the lane division in this intersection.

<u>338 – Markulantie/Kekkurintie</u>

Retrieved data on how many cars are turning left and continuing straight from the west direction. From the east, only the number of vehicles was known.

405-Markulantie/Tampereen valtatie

This is the largest intersection on the route. Therefore, all available data from this intersection was retrieved. One problem was that according to the detector placed on the right-turning lane on Tampereen Valtatie incoming from the north, 498 cars turned right. This number could not be right because it would have meant that almost all vehicles driving west came from this direction. The reason this showed such a high number could be because the detector is overlapping with other lanes or because the vehicles are still in the wrong lane at that point. Therefore, this number was corrected to 339, which is a more realistic number.

While all data available were collected in this intersection, there were still unknown vehicle numbers. For vehicles coming from the west on Markulantie, only the total number of cars was known. For cars incoming from the east on Markulantie and from the south on Tampereen valtatie, only the total number of cars and number of cars turning left was known. From the south, the vehicle number was known in every direction.

409-Markulantie/Raunistulan puistotie

Intersection 409 is the second-largest intersection in the route. Therefore, data from all the detectors at this intersection were retrieved. As in the previous intersection, there was still some uncertainties, even if all the detectors were retrieved. On the incoming lanes from the west on Markulantie, there is data on the total number of cars and cars turning left, but the number turning right or continuing straight is unknown. The same is true for cars incoming from the east. From the north on Kärsämäentie, only the total number of cars and how many cars will turn left was known. All turning ratios were known in the south direction.

420-Gregorius IX:N Tie/Paavinkatu

Data were retrieved from all lanes on Gregorius IX:n tie, including the lanes that were turning left and right. Gregorius IX is the road that route 99 follows. Therefore, the number of vehicles turning left is known, but the exact number of cars turning right or continuing straight is unknown because they share the same lane.

506-Halistentie/Hämeentie

Intersection 506 is outside the simulation, but the data is needed to know how many cars should leave or join the simulation in the east. For cars going south, only one of the lanes was equipped with a detector. Therefore, the number of cars on the other lane was estimated according to the previous intersection for the simulation. The number of cars incoming from Tammitie was also unknown.

6.2.2 Analysing the vechile data

Figure 15 was constructed by summing all detectors in an intersection and then calculating the median amount of cars per hour throughout the month. One major flaw is that several lanes had no vehicle data. This meant that in every intersection, except for 409 and 405, only the number of vehicles following the route is known. Therefore, 409 and 405 are not equally compared to the other here because they contain the other directions data as well, but they are still the busiest intersections in the route. More exact data can be found in

Table 7.

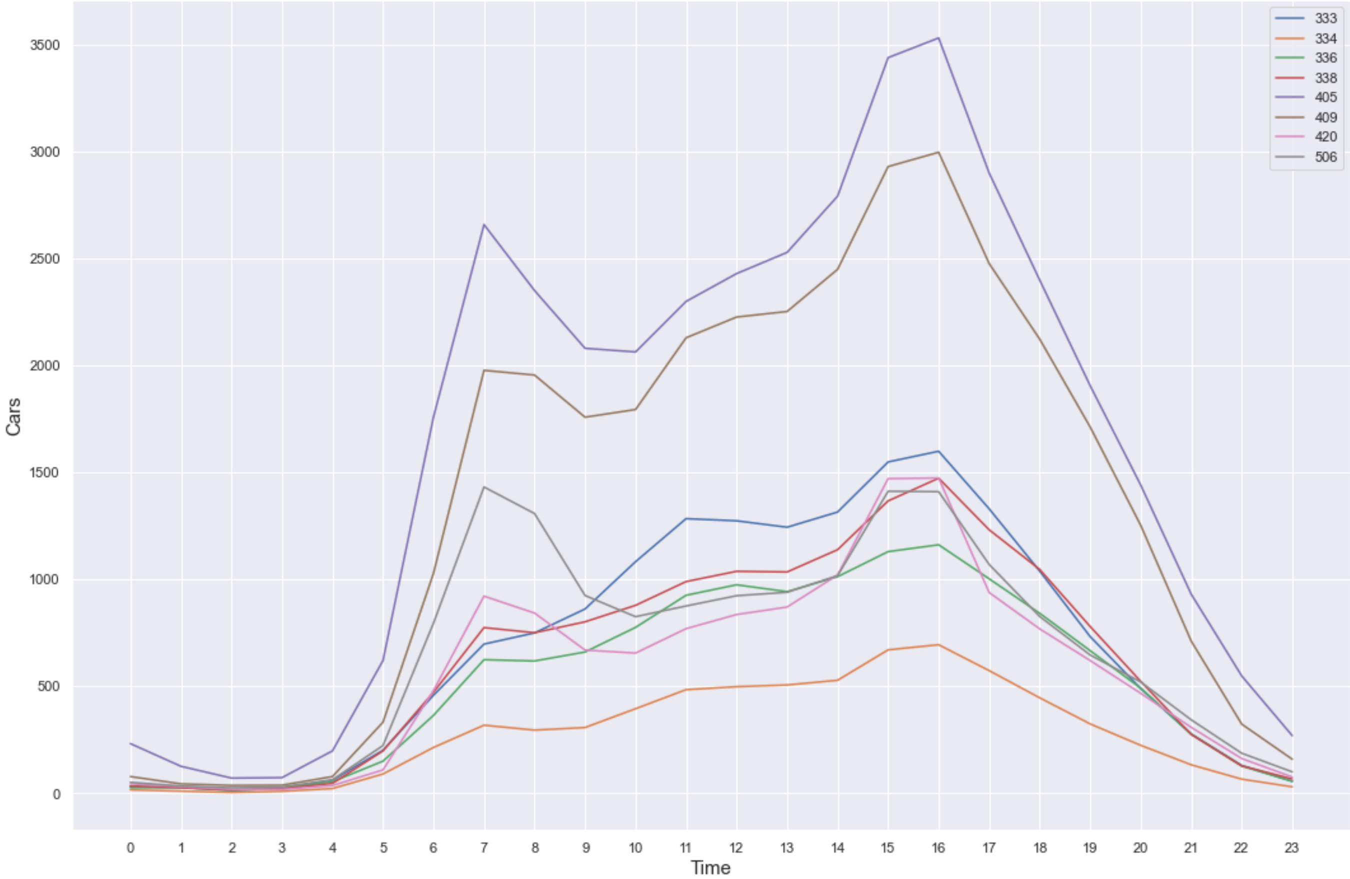

Figure 15: Median number of cars in intersection throughout the day

In Table 6, the number of vehicles going in each direction is shown. It should be noted that in 334, one of the detectors was not working correctly for the east direction, thus the value is lower than it should be. This table was made to understand why the west bus is slower than the east bus. According to the data, the traffic number is higher for the east direction in most intersections, except for intersection 409,410 and 420. The intersections with the most significant differences were 333,405 and 409. For intersection 333, the big difference is because cars that are going to Länsikeskus will use this route from Satakunnantie. By removing the turning vehicles from the original east value, the number of vehicles in 333 from east becomes 380, which is less than the total number of vehicles from the west.

In 405, the number of vehicles in the east direction is more than double the number of vehicles in the west direction. However, this was expected because the trunk road E63 continues to the north in intersection 405. E63, which ends in 405, is the trunk road between Turku port, Tampere, Jyväskylä, Kuopio and Joensuu and therefore has a high number of vehicles.

In 409, the west direction has over double the amount of traffic compared to the east direction. The map must be studied to understand why this is the case. First, out of the 698 vehicles, 244 vehicles are turning towards the city centre. Secondly, for the east direction, the vehicles turning towards the city centre is very low because they will have turned in the previous intersection (405).

These values were analysed to understand why there is a difference in the route times between the two directions. While the values differ a great deal between the intersection, it is still not possible to say with certainty if the duration difference is only due to congestion differences.

Table 6: Traffic congestion separated by direction

Intersection	East	West
333	834	491
334	242	305
336	504	503
338	565	573
405	621	309
409	302	698
410	No Data	No Data
420	492	539

6.3 Map data

Map data were downloaded from OpenStreetMap. OpenStreetMap is a project that offers free geographic data. It was developed because most map services have some restrictions on how the data can be used (OpenStreetMap, n.d.). Users can easily update service data in the software such as restaurants. The editing of road data is more restricted, which makes it a reliable data source for our simulation. The tool used to download the map data was JOSM, which is an OpenStreetMap editor.

6.4 Traffic light data

Traffic light data were collected from the traffic planning department in Turku. It is the same department that configures the traffic lights in Turku. Additionally, some interviews were held to gain further understanding of how the traffic lights should be configured in the simulation because it was not possible to replicate SYVARI in SUMO.

7. Simulation

In this chapter, the steps for building the simulation in SUMO will be explained. There will be some more information on how the data were collected. Then the functions that were used in SUMO will be explained. Other similar functions in SUMO will also be reviewed for comparison and to ensure that the most suitable methods are used for this paper. During the building of the simulation, many exceptions and simplifications had to be made. Why these were done and why they are acceptable will be explained in detail. This chapter will give an understanding of how time-consuming building this simulation in SUMO was. The amount of workload is the main reason it was decided to combine both the morning and evening rush hours to a single simulation, instead of building them separately.

Once the baseline simulation is ready, the model will be validated. Student t-tests and some face validity were used to ensure that the simulation reflects the real world correctly. The experiments will be implemented once the model has been validated and calibrated. The building of the experiments will then be explained.

7.1 Simulation settings

All the runs in the simulation were 20 hours, and each test was done five times. Hence the total simulation duration was 100 hours per test. The simulation seeds used were 1111, 5222, 9333, 13444, 17555. These are important for the random configurations in the simulation if the simulations are to be replicated.

7.1.1 Map data

OpenStreetMap was used to construct the simulation's road network (OpenStreetMap, n.d.). OpenStreetMap is a community-driven map editor, thus the coordinates for each object can easily be extracted from the website. One disadvantage of being community-driven is that the maps are not always correct. JOSM was used to filter the map before using it with SUMO. JOSM is a map editor, which is used for OpenStreetMap. In JOSM, the map was modified so that the base map only included the roads for this simulation and nearby roads.

After this, the map was converted to a NetEdit (SUMO) file with Netconvert. During the conversion, functions such as removing too tight corners and guessing the junction directions were used. Netconvert has several functions that can be used during the converting process, which are described on the SUMO website (Simulation of Urban MObility, n.d.). The automatic functions were kept to a minimum to minimise the risk that the map was incorrectly altered.

The map could now be opened in NetEdit, which is the map editor designed for SUMO simulator. The imported map can be seen in Figure 16. The figure shows that there are many roads that bring no use in this simulation. For the simulation, only the first road away from route 99 needs to be included. Therefore all other roads were removed. This process had to be done by hand, but it could quickly be done with the removal tool in NetEdit. The final version of the map can be found in Figure 17.

Another problem with the map was that there were a lot of extra edges throughout route 99. An edge is a snip of the road. In SUMO, it is possible to use a single edge until the next intersection. Having a single edge is beneficial because it makes the future work less complicated. For example, every edge must be mentioned when creating a route. Therefore, the creation of routes becomes more time consuming with unnecessary edges.

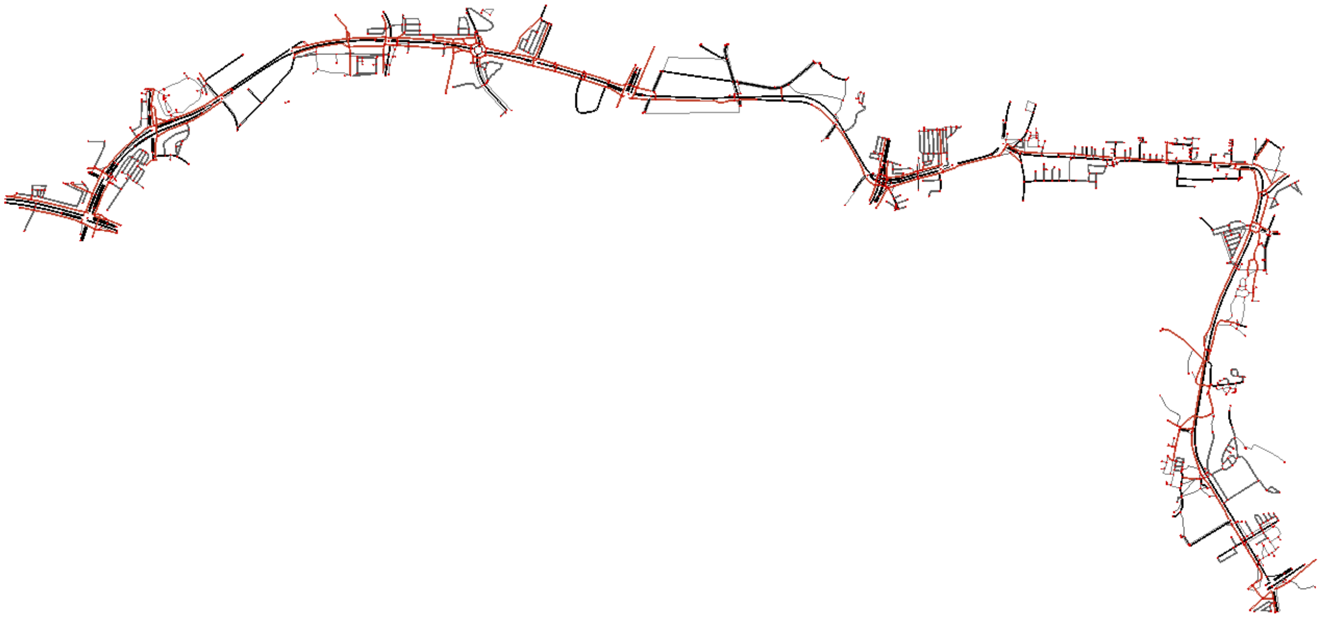

Figure 16: Uncleaned NetEdit map

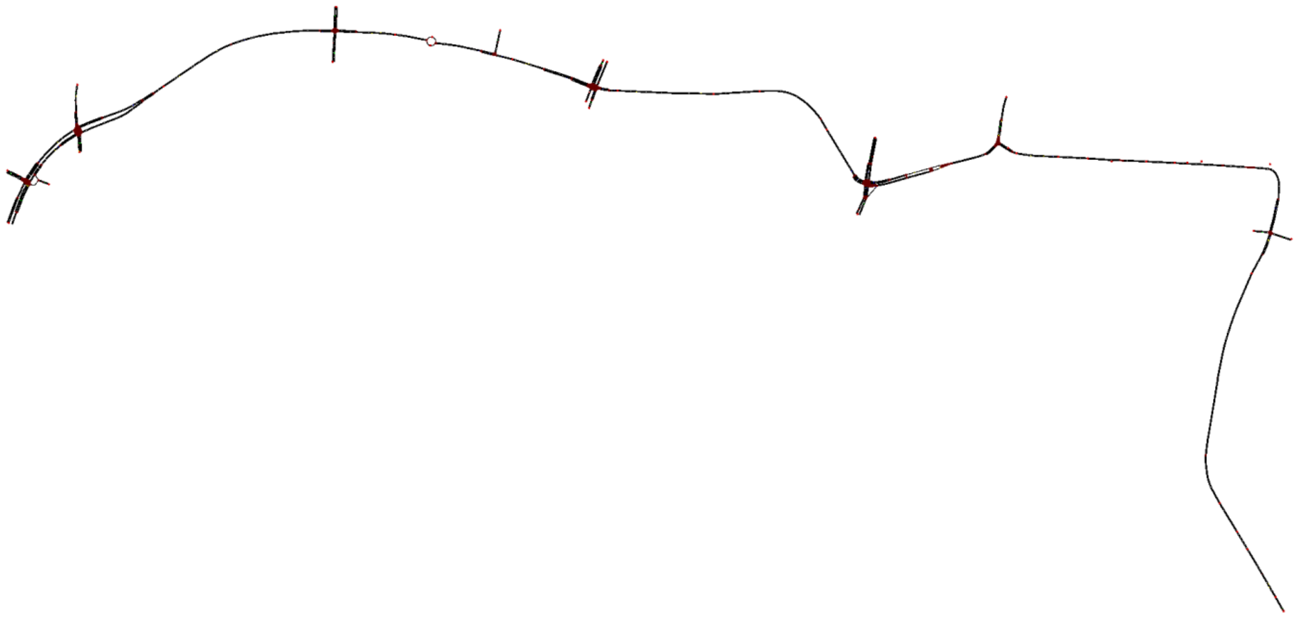

Figure 17: Final NetEdit map

When converting OSM maps, there is a function that guesses the direction for each lane in an intersection. This function was able to guess some intersections correctly, but most of them still needed to be re-coded. For example, NetEdit always allowed U-turns in intersections, which is something that is generally not allowed in Finnish intersections. These were all removed because the car doing the U-turn must wait for the oncoming traffic to pass before it can turn around, which causes congestion within the simulation. While U-turns happen every day, they are still uncommon. Therefore, it would be unwise to put this as an option in the simulation because it could cause problems that are non-existent in the real world.

Netconvert also got the lane directions often wrong in intersections. The lane directions were corrected by using google maps. The directions were then verified by driving through the route to check that no changes had been made since the Google Streetview pictures were taken.

In Turku, many bus stops are directly behind an intersection. These NetConvert often had a problem in recognising. On converting, the bus stops were often mistaken for an extra lane, which had to be changed.

7.1.2 Vehicle attributes

In SUMO, there are a lot of different attribute settings that can be applied to the vehicles (Simulation of Urban MObility, n.d.). Most of them are about direct vehicle attributes such as speed, acceleration and shape, but some of them are much more complicated. The more complicated ones were left untouched in the simulation because they often lacked proper documentation, except for Sigma.

Sigma is a variable that controls the driving imperfection. Having perfect driving means, for example, that the car can predict the other cars movement better and avoid bringing themselves to a complete stop when there is a red light. This value was lowered to 0.4 from 0.5 for buses. The value was lowered to make the buses better at merging after a bus stop. Furthermore, this is realistic because bus chauffeurs are professional drivers.

The maximum speed was set to 50 m/s (180 km/h) for both the cars and the buses. This value does not matter for the simulation as the drivers are still bound to follow the speed limits set in the simulation. In SUMO there are two variables called speedFactor and speedDev. These variables decide randomly if the car will drive slower or faster than the speed limit. At default, this option is enabled. However, it was disabled because the documentation on how it works was lacking. Having the bus drive at random speeds would create more variance in the duration, but it is impossible to know how much of the variance was from the bus driving slower or faster. Therefore, it would be more challenging to analyse the results.

The acceleration in the simulation was set to 1.314 m/s^2 for the bus and 1.818 m/s^2 for cars. These values are based on a study done in Japan, where bus acceleration and vehicle acceleration in intersections was studied (Sim & Lee, 2009). A corolla from 2010, which is one of the most common cars in Finland today, has an acceleration of 2.12 m/s² from 0-100 km/h. Therefore, an acceleration of 1.818 m/s² is realistic (Lukkari, 2019; Sortter, 2019).

7.1.3 Bus traffic

Bus schedules were not used in the simulation because implementing a schedule in the simulation does not affect the results when examining bus route duration. Additionally, because the vehicles are continuously spawning in a constant flow and the traffic lights are static. It is better to add some randomness to the simulation to minimize the risk that the bus is always approaching the same traffic light phase in the simulation. Using the bus schedule in SUMO would mean that every bus has to be defined separately, which makes it time-consuming to alter the code between experiments. Therefore, the bus spawning was instead implemented as a random function. In the simulation, there is a 0.1111111111111 % chance that a bus will spawn each second. This equates to 4 buses per hour. One problem with using the random method is that there is a risk that two buses will spawn right after each other. However, the chance for this happening is very slim. Additionally, multiple buses are using the same route in the real world, which means that this can happen there as well.

7.1.4 Bus stops

When converting the OSM map, most of the bus stops were included except for the ones directly after an intersection. The problem was that after correcting the intersections and edges, the original bus stops were either removed or misplaced. Therefore, all the bus stops were removed and added manually. Adding new bus stops in NetEdit is quickly done by placing it on the map and then modifying the size of the bus stop.

In NetEdit, it is not possible to add a bus stop in the middle of an intersection, which exists in the simulated route. An example of this is bus stop 1409, which is situated on the opposite side of a T-intersection. There are at least two workarounds for this. One is to teleport the bus to the side when it reaches bus stop 1409. This would probably be the best and most realistic solution, but it requires implementing extra functions to the simulation, which could potentially cause problems. Instead, the bus stop was placed before the intersection. This does not have any effects on the simulation results because it does not affect the traffic flow in the intersection.

In SUMO, the bus stops are defined in a separate file. The bus stops are then added to the bus route as scheduled stops. The duration of the stop is defined in the route file (route99.rou.xml). However, the bus stops duration can only be defined with a static value. Therefore, the variance in bus stop duration that exists in the real world cannot be implemented in SUMO.

For the simulation, the mean stop time will be used for each specific bus stop to make the simulations consistent. The mean was chosen instead of the median because when the no stop bus logs are included, the median becomes zero for many stops. Not including the no stop, bus stops would be wrong because then the bus stop duration would be longer in the simulation than in the real world.

In Turku, Mattersoft uses GPS signal for the bus logs. This GPS signal has a small delay between 0.5-1 second on average, but because this delay happens both when the bus arrives at the stop and merges again, it does not affect the data.

7.1.5 Traffic data

In the simulation, the median is used for the traffic data because then the extreme values have less influence on the simulation. These vehicle numbers on detector points were then converted into vehicles in the simulation. It is possible to do this manually, but it would be very time-consuming as every possible route would have to be defined separately. Therefore, SUMO offers several tools for traffic generation from detector data, which will be briefly explained below (Simulation of Urban MObility, n.d.).

DFROUTER

DFROUTER is the oldest traffic generator within SUMO. It generates traffic based on how many vehicles should cross a lane in a given time frame. It works by placing the real-world detectors in the simulation and feeding the detectors real-world data. DFROUTER will then generate the traffic until it matches with the detector data.

Flowrouter

Flowrouter is an improved version of DFROUTER and uses many of the functions that exist in DFROUTER. It is superior because it offers the ability to put more restrictions on what routes can be generated. Additionally, it is better at working with missing data and generating routes that maximise the vehicle flow according to the input data.

Jtcrouter

Jtcrouter is one of the newest tools with routeSampler. It works by inputting turn-count data into the simulation and building the traffic based on that data. Turn-count data means the number of vehicles turning in each direction at an intersection.

routeSampler

RouteSampler builds the traffic based on turn-count data, but instead of generating the routes, the user must define them themselves. RouteSampler then assigns vehicles to the set of routes that were given until the simulation matches the count data. For this to work correctly, the simulator must ensure that enough routes are defined.

Flowrouter was chosen for this study because it is most suitable for the available data. Jtcrouter and routeSampler could not be chosen because the turning counts were missing. Flowrouter was chosen over DFROUTER because it is newer and handles missing data better. Furthermore, DFROUTER is built to simulate highways according to documentation, while flowrouter is built for both cities and highways.

Even if flowrouter was designed to deal with missing data, the simulations quickly revealed that it was not able to generate the correct flows to different lanes. This is because flowrouter uses source detectors to generate the correct number of cars. The detectors that are placed as sinks or in-between detectors define the distribution (Simulation of Urban MObility, n.d.). Source detectors are the detectors which are placed at the beginning of the road, and they work as spawn points for new vehicles. Sinks are the detectors which are at the end of the route. The in-between detectors are placed in the middle of the route.

In this simulation, most intersections had no data on vehicles incoming from the north and the south, which would have been sources and sinks. Therefore, flowrouter was not able to generate the vehicles automatically. In flowrouter, the source is the essential value because the tool will assume that the source is correct and then use the middle and sink detectors to build routes for the vehicles. This meant that the simulation was not able to fill itself with enough vehicles. Therefore, estimates had to be assigned to roads to the sources.

The arbitrary numbers were estimated by calculating the number of missing vehicles in the next intersection. This was possible because the number of vehicles continuing straight and arriving at the next intersection was always known. With this vehicle number, we knew how many cars should turn to route 99 from the north and south directions. However, the number of vehicles turning from each direction was impossible to know. Therefore, the placement of the intersection was used to estimate the number of cars turning from the north and south direction. For example, in some intersections, it is less likely that the car would continue left because they could have taken a shorter route earlier.

There is still a considerable risk that these numbers are entirely incorrect. However, this is not a problem because the bus route is being studied and not the crossing traffic. This is not relevant in 405 and 409 because the vehicle number was collected from all directions in them. The arbitrary numbers in the other intersections do not invalidate the results because the traffic lights in the simulation are static. This means that even if the vehicle number is incorrect, the time the traffic light stays green is the same.

At intersection 409, the lane from the east direction and turning left (south) kept getting overflown in the simulation. Therefore, the original value was changed from 117 to 70. The missing vehicles were then redirected north instead because there was no risk of overflow. This way, the congestion stays the same on the bus route. In Figure 18, the lane in question is shown. The lane kept getting overflown because the traffic lights were static while in the real world, they are adaptive.

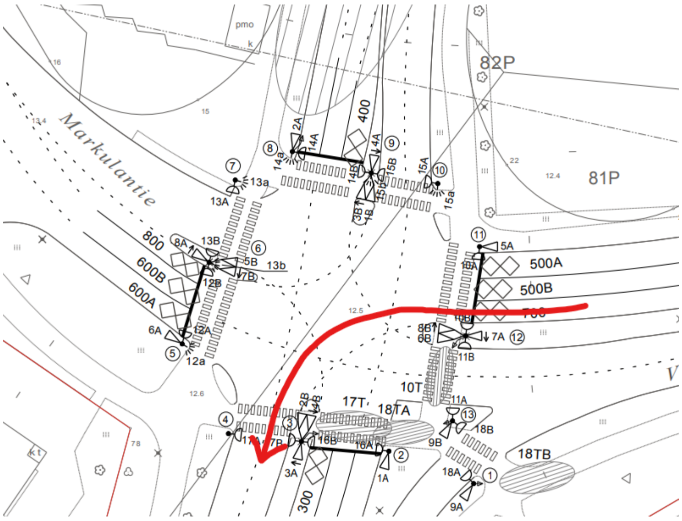

Figure 18: Intersection 409

7.1.6 Traffic lights

In SUMO, the traffic lights must be defined according to different phases. This means that different phases must be constructed and within each phase, the states of all the traffic lights must be defined. Figure 19 shows how traffic phases are defined in NetEdit.

Phases

dur	state	nxt
25.00	rrrrGGgrrrrgGg	
3.00	rrrryyyrrrryyy	
3.00	rrrrrrrrrrrrrr	
39.00	GGGrrrrGGGrrrr	
3.00	yyyrrrryyyrrrr	
1.00	rrrrrrrrrrrrrr	
15.00	rrrGrrrrrrGrrr	
3.00	rrryrrrrrryrrr	
4.00	rrrrrrrrrrrrrr	

Figure 19: Defining traffic light phases in NetEdit (intersection 334)

The states define whether the light should be green (G), yellow (y) or red (r). The small "g" stands for green light, but it must yield to conflicting directions. The small "g" is used in turning lanes where vehicles from the opposite direction simultaneously have green. The traffic light states are defined for each direction, even if there are multiple directions in a single lane. This is because the simulation must know which directions should have the yielding green (g).

As discussed earlier, within SYVARI, the lights are defined according to each lane, and there are no direct phases. The most resemblance to phases that exist in SYVARI is in which order the lights should become green, which is shown in Figure 20. However, these only define the lights that must be green during that phase. The passive greens are not defined here because they differ depending on if there is a need for a green light or not. Therefore, the traffic light settings had to be modified to fit the SUMO format.

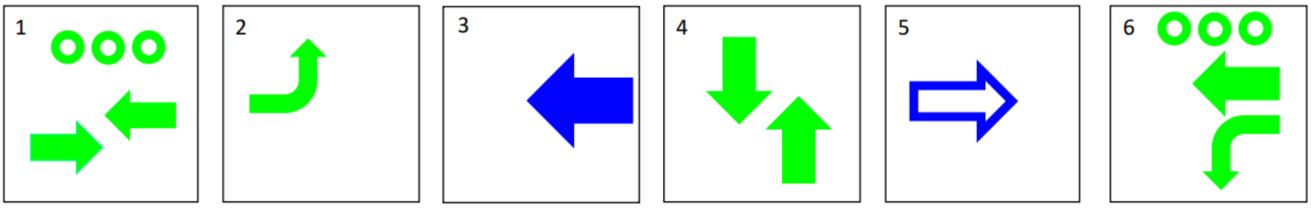

Figure 20: Traffic phases for intersection 334

Another problem was that in SYVARI, all directions have different min and max durations. In SUMO, it is possible to implement min and max durations, but only for phases and not individually for lanes In SYVARI, there is a background process that keeps track of each lane, which do not exist in SUMO. Therefore, it is not possible to implement adaptive traffic lights the same way as in SYVARI.

Instead, the phase length was defined by adding the min and max together and then dividing that by two. If the minimum time is 18, and the maximum time is 60, then the phase length in the simulation would be 39 ((18+60)/2). The adaptive function was not implemented because it would have been incorrect compared to SYVARI.

The phases were modified such that most intersections follow the format in Figure 21. The directions going straight will always have green in the same phase as the opposite direction, and this is also true for turning directions. However, the format was modified to fit the original format better for some intersections. Placing the bus directions in the same phase makes it much simpler to implement the bus priority functions. Otherwise, the priority function has to be defined separately for both directions. The fact that lane-based traffic configuration does not exist in SUMO meant that these exceptions had to be made.

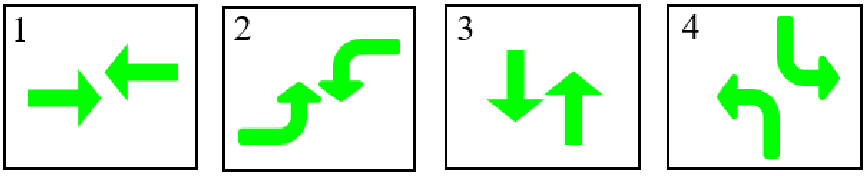

Figure 21: Most commonly used traffic phase format in the simulation

In SYVARI, the traffic lights are defined depending on what time is it. For the simulation, the light setting number one was chosen, which is used between 8:30-15:00 and 17:30-20:00. The other settings are either outside the rush hours or when the traffic flow is significantly higher in one direction. In general, the difference between the settings is still relatively small since the traffic flow distribution does not differ that much except for morning and evening rush hours. However, because the traffic data were generated based on both rush hours, the difference in the number of vehicles between the directions are rather slim.

7.1.6.1 Offset

SUMO does offer an offset tool. However, this cannot be implemented with priority functions. The problem is that once the bus priority functions are implemented, the offsets will stop working because in some intersections the lights might be shortened and in others extended. Therefore, the whole offset functions would be off sync after the first buses have driven through the route. In the real world, the lights will automatically sync themselves back to the offset value within SYVARI, but this is not possible to implement in SUMO.

7.1.6.2 Safety time

The safety time in SYVARI is defined in a matrix where all directions are presented on both the x- and y-axis and then give the correct safety time where the desired column and row intercept. An example of this can be seen in Figure 22.

In SYVARI, the traffic lights need to know the safety time between each direction because the phases can be changed independently. This means that one light might change after three seconds another after four and then a third after six seconds, depending on what the safety time for each direction is. In SUMO, every direction is defined together within one phase, which means that the safety time had to be implemented differently.

In the simulation, the safety times were implemented by checking which directions have the longest safety time between the previous and the upcoming directions. The longest safety time is then used for every direction by adding a red phase where every direction has a red light to clear all vehicles. If the safety time is 5 seconds, then the red phase would be 2 seconds. The yellow phase is always 3 seconds in this simulation, which is included in the safety time.

	Suoja-ajat ja aloitusviiveet																				Punakeltainen	Minimivihreä	Vilkkuvihreä	Kiinteä keltainen	Muuttuva kelt.
	Ohjelmoinnin alkava ohjaussuunta																								
Päättyvä tai määräävä ohjaussuunta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18							
1				4	9	5	5	4						5	8	4	4				1	10			
2			5		5	9	5	5						4	4	5	8				1	14			
3		6			5	5	4	6				5	8			4	4				1	5			
4	6				5	5	6	4	8	6	9			4	4						1	5			
5	3	9	4	5				4		4	4	5	8								1	14			
6	6	5	5	4			5		8	5	8	4	4								1	10			
7	4	5	5	4		6				4	4					5	8				1	5			
8	5	5	4	6	6							4	4	6	9						1	5			
9				2		2												4			1	6			
10				5	10	5	10															14	3		
11				5	5	5	5															11	2		
12		(1)	-		5	10		10														15	3		
13		(1)	5		5	10		10														15	2		
14	4	9		9	(1)			-														15	2		
15	4	9		9	(1)			4														15	2		
16	9	-	9			(1)	-															18	2		
17	9	5	9			(1)	5															11	2		
18									5													7	-		

Figure 22: Safety times for intersection 409

7.1.7 Pedestrians

Pedestrians will not be considered in this simulation because they are few in this route. Furthermore, pedestrians are more random compared to vehicles, and therefore more challenging to simulate accurately. Additionally, there is no data available about pedestrian traffic in Turku.

7.1.8 Road speed

Google maps and remix was used to check vehicle speeds. Remix is an urban planning tool, which has almost identical functions to google street view (Simulation of Urban MObility, n.d.). The tool is used internally within Turku city. Additionally, the whole route was checked in the real world to make sure that everything was correct. Throughout the route, the speed limit is 50 km/h.

7.2 Validation of simulation

Next, validation tests will be done to ensure that the simulation represents the real world correctly. Already during the building of the simulation, some validation was done by discussing results with people working at Turku city and then modifying the simulation. These have already been mentioned in the previous chapters. Now, the baseline simulation (no priority) will be compared to the actual data to test if the simulation accurately describes the real-world system.

7.2.1 Route validation

This validation was mostly done by examining the route in NetEdit. As said earlier, the route was first generated automatically, but the automatic generation had several mistakes in it that had to be fixed. For example, all speed limits were implemented manually. Furthermore, the number of lanes had to be changed in many intersections. The bus route length is 6.4 km, which is correct according to google maps.

7.2.2 Traffic lights

Traffic lights were not implemented in the simulation identically as in the real world. This can quickly be noticed in the simulation. In the simulation, directions have a green light, even if there is no car currently at the intersection. However, this can be true in the real world too because most directions have a minimum time of green per phase, meaning that the light can be green even if there is no car going in that direction. Nonetheless, having an adaptive green light length makes the traffic lights more efficient than having a static one. Split-phasing is also not implemented in the simulation, which further decreases the effectiveness of the traffic lights in the simulation.

Intersection 409 also had to be modified a bit from the original settings because the cars incoming from the east and turning south kept getting overflown. Therefore, the traffic phase length was increased to 13 seconds from 10 seconds. This was necessary because adaptive traffic lights were not implemented in the simulation. However, it is such a small change that it should not have any effect on the simulation results.

7.2.3 Traffic data

The traffic data were validated by placing induction loops in-between the intersections. While flowrouter was supposed to handle missing data, the validation showed that many roads were missing a significant number of vehicles. The reason it did not match was that flowrouter assigns vehicles to the simulation based on the source detectors and those were an estimate because the vehicle numbers from the south and north directions were unknown. Flowrouter estimates how many cars will turn in which direction, but there are always many different solutions for this. This means that even if the data is 100% correct, it does not mean that the turning ratios in the simulation are the same as in the real world.

Another problem was that vehicles in the simulation did not necessarily use the same lane choices as they would have in the real world. This was noticed when analysing the traffic data in
Table 7 because the total number of vehicles matched the real data, but when looking at the lanes separately, there was often a significant mismatch between the data. This could potentially have some effect on the results because the driving speeds could be slower in the more congested lanes. This effect is believed to be small in the simulation because the lanes were never fully congested, meaning that the vehicles were still able to drive according to the speed limits. In SUMO the vehicles are also able to change lane if they notice that one lane is really congested. Therefore, the vehicles in the simulation must drive on the preferable lane for their route, which causes the mismatch.

Table 7: Validation of detector data

Detector	Actual detector data	Simulation detector data	Simulation average speed
333_106	130	130	46.24
333_108A	227	455	33.09
333_108B	228	4	19.74
333_207A	251	146	47.14
333_207B	241	357	44.91
333_306	250	250	29.94
334_107	228	517	44.68
334_107L	352	45	47.52
334_207	183	239	46.02
334_207L	122	70	47.51
336_108	504	508	29.75
336_208L	503	507	43.80
338_100	37	37	15.99
338_108	565	569	44.28
338_208	574	548	29.51
405_110A	270	360	35.32
405_110B	299	185	31.44
405_210A	557	285	47.09
405_210B	264	536	44.85
405_305	68	68	20.69
405_403	39	39	47.14
405_503	285	285	45.64
405_607	309	312	42.00
405_707	329	304	45.96
405_800	10	6	18.40
405_907	292	292	46.05
409_103	118	82	22.80
409_107	102	111	47.15
409_107L	383	374	42.35
409_207L	179	235	47.58
409_300	23	25	17.95
409_302	25	25	42.46
409_400	113	113	22.29
409_405	227	171	47.65
409_507	188	188	22.32
409_507L	266	270	28.24
409_607L	302	306	44.34
409_707L	245	244	24.16
409_800	30	80	26.65
409_903	282	282	32.28
420_102A	504	508	32.75
420_102B	35	35	41.25
420_202A	468	472	38.64
420_202B	28	28	47.34

The induction loop data in

Table 7, shows that the mean speed is between 42-47 km/h for almost all detectors that were placed over 150 m from the intersection. The results show the significant mismatch between the simulation data and the real world between lanes. However, the total number of vehicles is equal if the detectors are paired with the second lane. Next, the intersections will be reviewed in detail.

Intersection 333

At intersection 333, the lane division can instantly be seen. For the detectors that have an A and B variant, there is a significant mismatch. Especially in 108A and B, where 455 vehicles drive on one of the lanes while only four vehicles on the other lane. Together, the sum is 459 in the simulation and 455 in the real world, which is only a four-vehicle difference. When looking at the average speed between the detectors, the less congested lane is slightly slower. What happens here is most likely that only buses are driving on that lane, because they will have bus stop on the right-side lane after the intersection. Therefore, the right-side lane is the most optimal for the buses, but not for the other vehicles. A similar mismatch can be found in detector 333_207A and 333_207B, but the sum is almost equal again.

Intersection 334

Here is a significant mismatch between the lane data, but the sum is very similar. For this intersection, the more congested lane is a bit slower. However, the difference is still only 3 km/h, which could be because the more congested lane is already slowing down for the upcoming intersection.

Intersection 336

At intersection 336, there are no mismatches, but the average vehicle speed for 336_108 is only 29,94 km/h. That is because the induction loop is placed less than 30 metres from the intersection. Thus, the vehicles have already reduced their speed before the intersection.

Intersection 338

In 338, there are no surprises. The speeds are relatively low, but that is because the detectors are placed close to the intersection.

Intersection 405

Intersection 405 is the most congested intersection on this route, which can also be seen from the vehicle data. Here, the number of vehicles in both 110A,110B,210A and 210B are wrong. However, the sums are similar, with 110 having 545 vehicles per hour in the simulation and 569 in reality. Detectors 210A and 210B have 821 vehicles per hour in the simulation and 821 vehicles per hour in the real world.

Intersection 409

In 409, detector 400 had zero vehicles per hour in the original simulation. This is the detector for vehicle incoming from north and turning left towards intersection 410. For some reason, Flowrouter generated zero vehicles to this route even if it was fed data that there should be vehicles. This was fixed by manually adding 120 vehicles per hour to this route that then continued north in the following intersection (410). These 120 vehicles were then removed from another route to keep the vehicle count correct, while still matching the induction loop data.

Intersection 410

For intersection 410, there was no real data. The data in

Table 7 is an estimate done by how many cars there were in the previous intersections. The detector 410_E is placed between intersection 409 and 410.

Intersection 420

At intersection 420, the vehicle numbers match, while the vehicle speeds are a bit slower. However, that is because the detectors are close to the intersection.

7.2.4 Vehicles

Some observations were collected from intersections to validate the acceleration in intersections. In the observations, the number of cars able to drive into the intersection from the time the light turned green was calculated. In total, 100 observations were collected, which are shown in Figure 23. These results were then compared to the simulation by observing the behaviour there. From this face validation, it was concluded that the acceleration in the simulation is practically identical to the real world.

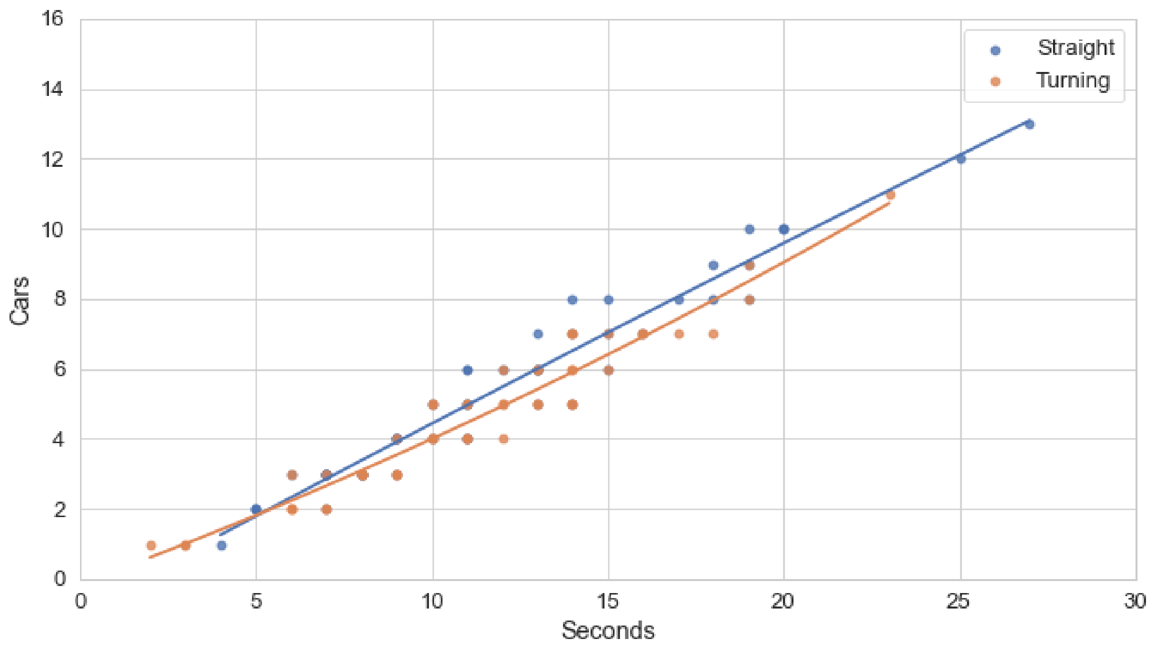

Figure 23: Cars driving through green light

From the observations, it was noticed that the drivers in the simulation are more careful compared to real drivers. This was obvious when studying the window when the traffic light is about to turn red. In the simulation, cars always stop during the yellow phase if it is possible to do so without abruptly stopping. However, in the real-world cars often drove through the yellow light even if it had already been yellow for 2-2.5 seconds, which means that it is about to turn red. Some drivers even accelerated to reach the light before it turned red. In the simulation, the drivers instead had faster reaction time when the light turns green. Therefore, the total number of cars became the same.

7.2.5 Bus data

Next, the bus data from the simulation was validated. Here the bus stops and route duration were compared to the actual data to check if they are similar enough to produce reliable results in the experiments.

5.2.3.1 Bus stops

In the beginning, the median bus route times were planned to be used in the simulation. This was because extreme values have little effect on the median value. The problem was that using the median meant that most of the bus stops would be zero seconds. This means that the bus will not stop at the bus stop most of the time. Therefore, the mean value was chosen for the simulation because then the bus will stop at every bus stop. The values used in the simulation are in Table 8.

Table 8: Bus stop times used in the simulation

Direction	Outside rush hours	Rush hours
East	71	141
West	86	117

In the original simulation, the bus stops were defined according to the data from the bus logs. However, once the values from the simulation were analysed, it became apparent that something was wrong because the bus stop duration was significantly longer in the simulation. This is because, in SUMO, the defined bus stop duration is only for how long the bus is going to pick-up passengers. Afterwards, the bus has to merge with the traffic again and how long this takes depends on the traffic. In the real world, this is accounted for in the bus stop duration. Therefore, the bus stop durations had to be changed accordingly in the simulation.

The bus stops were shortened evenly at first, but this meant that some bus stop durations became zero seconds, which was not acceptable. Therefore, the longer bus stops were shortened instead of to the shortest ones. This will not affect the result because the whole bus route is analysed. Therefore, it is only essential that the sum of the bus stops durations are the same as in the real world. The result from the base simulation can be found in Table 9: *Bus stop time statistics*Table 9.

Table 9: Bus stop time statistics

	Real World		Simulation	
	East	**West**	**East**	**West**
count	641	614	400	400
mean	141.1035	116.61	140.76	115.91
std	89.25	87.11	11.97	14.35
min	0	0	113	84
25 %	87	17	132.75	106
50 %	146	108	139	115
75 %	198	177	148.25	125
max	421	396	178	179

According to Table 9, the bus stop durations are far more consistent in the simulation compared to the real world. This was expected because the duration was a static value in the simulation. After changing the bus stop times the stop duration, there is less than half a second difference between the stop times in the real-world data and the simulation data. If the standard error in the simulation is compared to the standard error in the real world, then it can quickly be concluded that the simulation has much less difference in durations between the stops compared to the real world. Sadly, there is no function available in SUMO to implement adaptive bus stop durations. For the result, this means that the standard error will be higher in the real world compared to the simulation. The minimum and maximum values will also be lower and higher for the real world compared to the simulation in the results.

One problem is that while the bus stop times are now the same in the baseline, they do not have to be that in the actual experiments. In the experiments, the buses might merge with the traffic faster or slower, which will change how long the bus stop times are, which must be considered.

5.2.3.2 Bus route duration

Next, the bus routes will be validated. It is expected that the route times will be longer in the simulation compared to the real-world data. This is because there is no offset implemented in the simulation. The directional green phase length is also adaptive in the real world, which is not possible to implement in the simulation and the traffic lights in the real world allow for passive greens, which do not exist in SUMO. Below some terminology used in SUMO outputs files will be explained.

<u>Duration</u>

Duration simply means the duration it takes for the bus to drive the whole route.

<u>Waiting time</u>

Waiting time removes the time when the vehicle speed was below 0.1m/s from the duration. This removes all time that was spent in traffic lights from the duration. Scheduled stops are not included in the time removed, which means that bus stop durations are not removed.

Time loss

Timeloss calculates the time that was lost because of not driving at the speed limit. For example, if the vehicle is driving 45 km/h on a road where the speed limit is 50 km/h, then the time that was lost because of driving too slowly will be removed from the duration. Timeloss excludes time lost at intersections. However, bus stops are included.

An empty simulation was first compared to the real-world values from outside the rush hours (18:00-6:45). This simulation is to show how significant the effects of the static traffic lights are. Only buses were driving in the simulation, which means that there was zero congestion. Therefore, the duration should be lower in the simulation compared to the actual data.

Table 10: Outside rush hours results (minutes)

	East				West			
	Actual duration	**Simulation duration**	**No waiting time**	**No time loss**	**Actual duration**	**Simulation duration**	**No waiting time**	**No time loss**
count	357	399	399	399	211	399	399	399
mean	12.63	13.55	11.96	9.02	12.73	14.32	12.31	9.28
std	12.63	0.88	0.21	0.03	1.33	0.96	0.20	0.02
min	12.63	11.40	11.37	9.00	9.22	11.80	11.75	9.26
25 %	12.63	12.98	11.82	9.01	11.66	13.58	12.18	9.27
50 %	12.63	13.62	11.95	9.01	12.77	14.37	12.32	9.28
75 %	12.63	14.20	12.08	9.02	13.57	14.99	12.43	9.28
max	12.63	15.63	13.02	9.27	16.75	16.90	12.97	9.51

According to
Table 10, the buses in the simulation are slower compared to the real-world buses. This can be explained by the differences between the simulation and the real world. First, the bus stops in the simulation must be given a single duration value, while in the real world the bus might not stop at all. While the duration means are precisely the same in both cases, it still affects the routes durations because when a bus stops at a bus stop, it first has to slow down and then it has to speed up again when merging with the traffic after the bus stop. While this only takes a couple of extra seconds per bus stop, it accumulates quickly over the total 11 bus stops in each direction.

Secondly, the traffic lights in the simulation are static when they should be adaptive. This means that the difference in bus route times when there is the minimal and medium amount of traffic is small because the bus will still have to wait the same amount of time at the traffic light. In the real world, there is a significant difference in the time bus must wait at the traffic light because the traffic lights are adaptive. For example, in the simulation shown in

Table 10, there were only four buses on the road per hour in each direction. However, even then, the route time was much longer in the simulation than in the real world. Therefore, adaptive traffic lights are crucial when working with small amounts of traffic.

In Table 11, data from the rush hour simulation is compared to the actual rush hour data. These simulations are still for validation, thus no priority was implemented. For this simulation, it was expected that the durations in the simulations would be slightly longer than the actual data because there are no green waves or adaptive traffic lights implemented in the simulation.

Table 11: Rush hour simulation results (6:45-9:00 and 14:00-17:00)

	East				West			
	Actual duration	**Simulation duration**	**No waiting time**	**No time loss**	**Actual duration**	**Simulation duration**	**No waiting time**	**No time loss**
count	538	411	411	411	466	435	435	435
mean	14.67	15.34	13.48	10.18	15.66	15.78	13.18	9.80
std	1.71	0.90	0.36	0.21	2.00	1.14	0.40	0.24
min	9.05	12.85	12.60	9.77	9.22	12.53	12.18	9.26
25 %	13.6	14.65	13.25	10.04	14.40	15.03	12.92	9.64
50 %	14.62	15.25	13.45	10.16	15.62	15.75	13.17	9.77
75 %	15.7	15.92	13.68	10.31	16.89	16.50	13.43	9.95
max	20.28	17.57	14.72	11.20	21.72	19.38	14.85	10.45

In the rush hours simulation, the bus route durations are much closer to the actual data compared to the outside rush hour simulation. This is because the effects of not having adaptive traffic lights in the simulation decrease as the intersections become more congested.

When validating the data from the rush hour simulation, the difference between the data from the west direction is only 7.2 seconds. These seven seconds could be because the route in the simulation is slightly longer than the route calculated in the actual data. In the actual route, the route begins at the first bus stop and ends at the last bus stop, while in the simulation it is calculated from the source to the sink. The source and the sink are placed at the intersection before the first bus stop in both directions. Therefore, the route in the simulation is roughly 50-100 metres longer than the actual route. For a bus driving at 50 km/h, this would add between 3-8 seconds to the route time.

For the east direction, there is still a mismatch of roughly 59.4 seconds. This difference cannot be explained by just the different bus stop times or route length. In the simulation, the average bus stop time for the east was 140.99 seconds, while for west it was 117.4 seconds and the real bus stop values are 141 seconds and 117 seconds. Therefore, the difference must be from how the traffic lights are configured in SUMO.

In the data analysis chapter, the difference in route times between the east and west direction was discussed. According to the results, the difference in route times does not exist in the simulation. This is a bit surprising since the traffic numbers are the same in the simulation as in reality. Therefore, the delay from the congestion should be the same in both the real world and in the simulation. In the simulation, the difference in route time is only about 20 seconds, while in the real world, the difference is almost a whole minute. Therefore, the difference in the route times in the real world is probably from how the traffic light assigns the different phase, which could not be remodelled in SUMO

Next, a t-test will be performed to check if the simulation accurately describes the real-world situation. The real-world data will be compared to the no priority simulation results from the rush hours. The values used to do the t-test are shown below. The null hypothesis is that the two samples are equal, and the alternative hypothesis is that they are not equal. In a t-test, the null hypothesis is true if the p-value is larger than the alpha value, which in this calculation was chosen as 0.05. The alpha value describes what the chances are that the given result is random, and there is no significant difference between the results.

$$\text{Null hypothesis: } \mu 1 = \mu 2$$

$$\text{Alternative hypothesis: } \mu 1! = \mu 2$$

$$\alpha = 0.05$$

<u>East values</u>

$$\mu 1 = 880.189591\,, \qquad \sigma 1 = 102.687723, \qquad n1 = 538$$

$$\mu 2 = 920.267639902676\,, \qquad \sigma 2 = 54.1178154609922, \qquad n2 = 411$$

<u>West values</u>

$$\mu 1 = 939.424893\,, \qquad \sigma 1 = 120.024369, \qquad n1 = 466$$

$$\mu 2 = 946.806896551724\,, \qquad \sigma 2 = 68.1884718575175, \qquad n2 = 435$$

<u>P-value</u>

$$p - value\ east = 1.3502235941643193e^{-12}$$

$$p - value\ west = 0.26110728380728787$$

The t-test for the east direction failed, the chance of the result occurring at random is essentially zero. The t-test for the west direction passed. As stated earlier, the traffic lights in the simulation are not calibrated identically in simulation as in the real world because it is simply not possible. One possible solution is to change the traffic phases, but this is risky as lowering the durations could cause the intersections to overflow. It would be possible to reach better results if other parameters such as acceleration, speed and bus stops would have been changed, but then they would be wrong, and the experiments would have just become more unreliable. Therefore, it was decided that these results are acceptable and that the current parameters can be used in the experiments.

The results from the simulations might not always correlate with the effects in the real world because the simulation is not an exact representation of the real world. This was expected because of the static traffic lights and having to use traffic phases instead of split-phasing. The results are expected to be more positive in the simulation than they would be in the real world. This is because, in the baseline simulation, the bus will always have to wait for the whole traffic phase length, even if there are no other cars in the intersection. However, in the real world, the lights would change if there are no vehicles in that direction. When implementing the priority system, the lights will change, even if there are other cars in the intersection. Therefore, the improvements will be more significant in the simulation as there was extra unnecessary waiting in the baseline simulation. In the results chapter, the mismatches with the real world will be taken into consideration to create an objective viewpoint of how the results can be used to analyse the implementation of a TSP system in Turku.

7.3 Bus priority implementation

The transit signal priority system was implemented by placing detectors on every lane before and after the intersection. The detector before the intersection will send a priority request, and the detector after the intersection will terminate said priority request.

When a bus crosses one of the detectors, the bus id will be saved to a list that keeps track of which busses are in what intersections. A loop will then check which priority type should be given to the bus depending on what the current phase is. The after-intersection detector will then remove the bus id from the priority request list. Furthermore, the after-intersection detector ends the phase when there has been an extension or an extra green light.

Induction loops were used in the simulation to simulate the use of a GPS based transit signal priority system. Induction loops are a bit different than the GPS signal because they are 100% accurate and also because there is no delay between crossing the detector and the traffic light receiving the priority request. The delay was not implemented in the simulation because while there are inaccuracies in the real-world system, it mainly affects the extension. The GPS inaccuracies are also so small that it should not have a significant effect on the result, and can be countered by placing the detectors farther away.

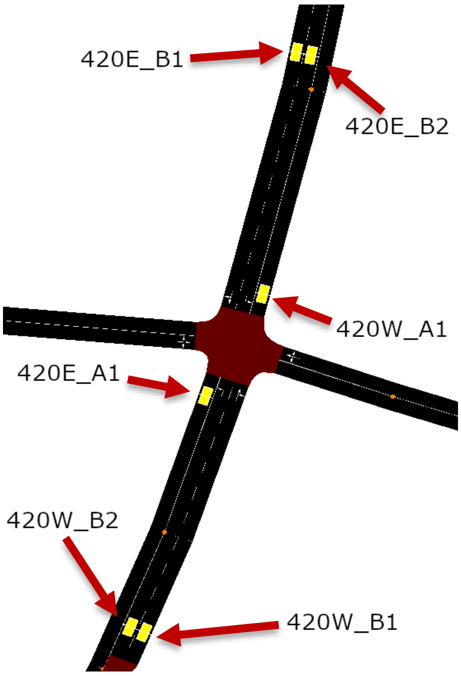

Figure 24: Bus detectors at intersection 420 in simulation

The induction loops scan every second if a bus drove past them. If there is a bus in the induction loop data, then the simulation will check what the current phase in that intersection is. There are two types of detectors in the simulation for the priority system. Some detectors are placed before the intersection, which are called B1 and B2 in Figure 24. These detectors are moved freely depending on what the tested detector distance is. The detectors after the intersection are named after the intersection number plus an "A" and a unique id number. These detectors are always placed 5 metres from the intersection.

7.3.1 Extensions

Because the traffic phase's length was calculated by adding the min and max value and dividing it by two, the extensions are not entirely accurate with how the system works in the real world. In the real world, the extension time is added to the max time. The extension time is usually 30 seconds. Thus, if the max time is 60 seconds, then the lane can be green for 90 seconds. However, in the simulation, the phase time was calculated by adding the minimum and maximum duration and then dividing them by two. Therefore, the extension could either be based on what the maximum time would be in the real world or based on the extension time defined. The difference was concluded to be insignificant, and therefore the original 30 seconds was used. The reason is that if the bus is driving according to the speed limit, which is 50 km/h, then the bus travels 417 metres in 30 seconds. Even if the bus would be driving only 25 km/h due to congestion, it still travels over 200m within 30 seconds. This is farther away from the intersection than any of the detectors were placed. Therefore, there is no risk of a bus missing the extension.

7.3.2 Red light shortening

The red light shortening was applied to the simulation by creating a list where every intersection minimum phase time was stored. When a bus entered the intersection, and the current phase was not the correct one for the bus, then the simulation checked whether the current phase could be shortened or not. If the phase could be shortened, then the function calculates by how much it could be shortened based on the minimum time and the duration left of the current phase.

Below the function that calculates by how much the current phase can be shortened is shown. The function is written in Python.

$$Phase\ length = min(max(MinTime - (phaseTotalDuration - phaseDurationLeft),0), MinTime)$$

7.3.3 Extra Green

Extra green only exists in intersection 334 and 409 because those are the only intersections where it is planned to be implemented in the real world. Because of this, the extra green was built as a separate function before the extension and red-light shortening functions.

The extra green had to be modified to fit the SUMO format. As discussed earlier, the phases in the simulation had to be constructed differently than in SYVARI due to the fact that SYVARI allows passive greens, which optimises the traffic flow. For example, phase 3 in Figure 25 means that if there is a bus approaching from the east, then an extra green phase will be inserted between phase two and four. In SYVARI, all the non-colluding directions could then have passive green during this time.

SUMO does not allow passive greens. Therefore, when an extra green is requested in SUMO, the traffic light will switch to the phase where the bus direction is green. In Figure 26, this is phase 1. For example, if extra green is requested in intersection 334, then the phases would be identical to Figure 26.

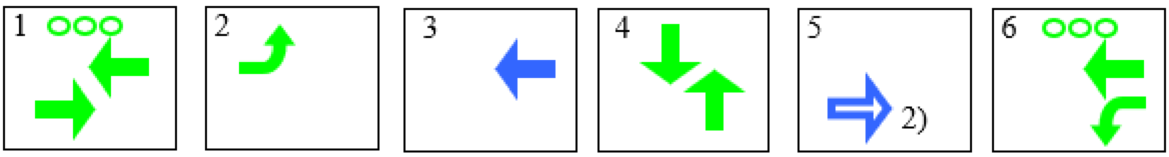

Figure 25: SYVARI traffic configuration for intersection 334

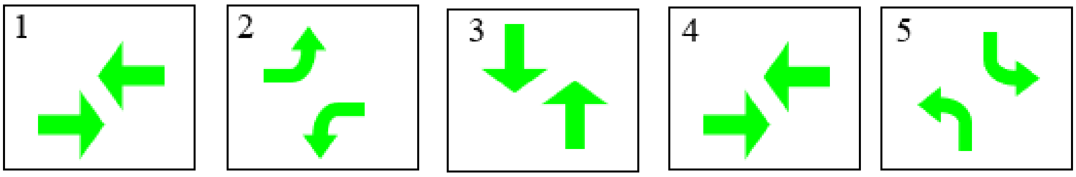

Figure 26: Intersection 334 in SUMO

In Figure 25, phase five, there is a hollow blue arrow. This means that upon a priority request, there can be a green rotation, which means that the phases rotate so that the bus will more quickly receive a green light. This was not considered in the simulation because there is no such function in SUMO. This was not seen as a significant change because of the extra green, in phase four, in Figure 26.

One thing that was discussed was whether the traffic lights should allow the extra green in both phase 3 and 5 seen in Figure 26 or only in one of them since it will give the green light to both directions simultaneously. Here, it was decided that the extra green would only be implemented once per traffic phase to limit the disruption in the traffic flow.

The other intersection with extra green was intersection 409. Intersection 409 followed the same logic as used in 334.

7.4 Analysing the data

From SUMO, it is possible to collect a wide variety of statistics with different inbuilt functions in the software, such as emission statistics, safety measures, traffic light statistics and also standard route statistics (Simulation of Urban MObility, n.d.). In this paper, the trip info statistic was mainly used. Trip info produces information about the vehicles in the simulation, such as route duration, bus stop duration and travel speed.

7.5 Experiments

This paper aims to discover whether it is worth implementing a TSP system in Turku, but the paper will also test different variables in the TSP system. The tested variable is how the detector distance affects the effectiveness of the TSP system.

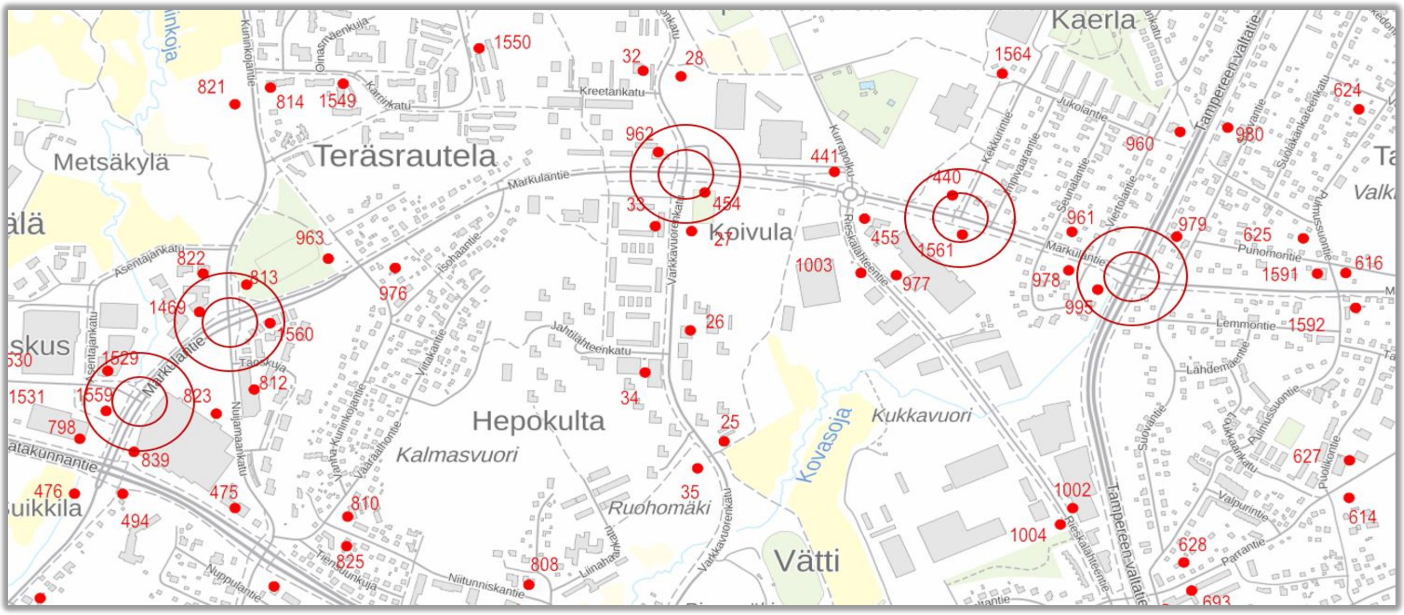

Figure 27: 100 and 200 metres circled around intersections part 1

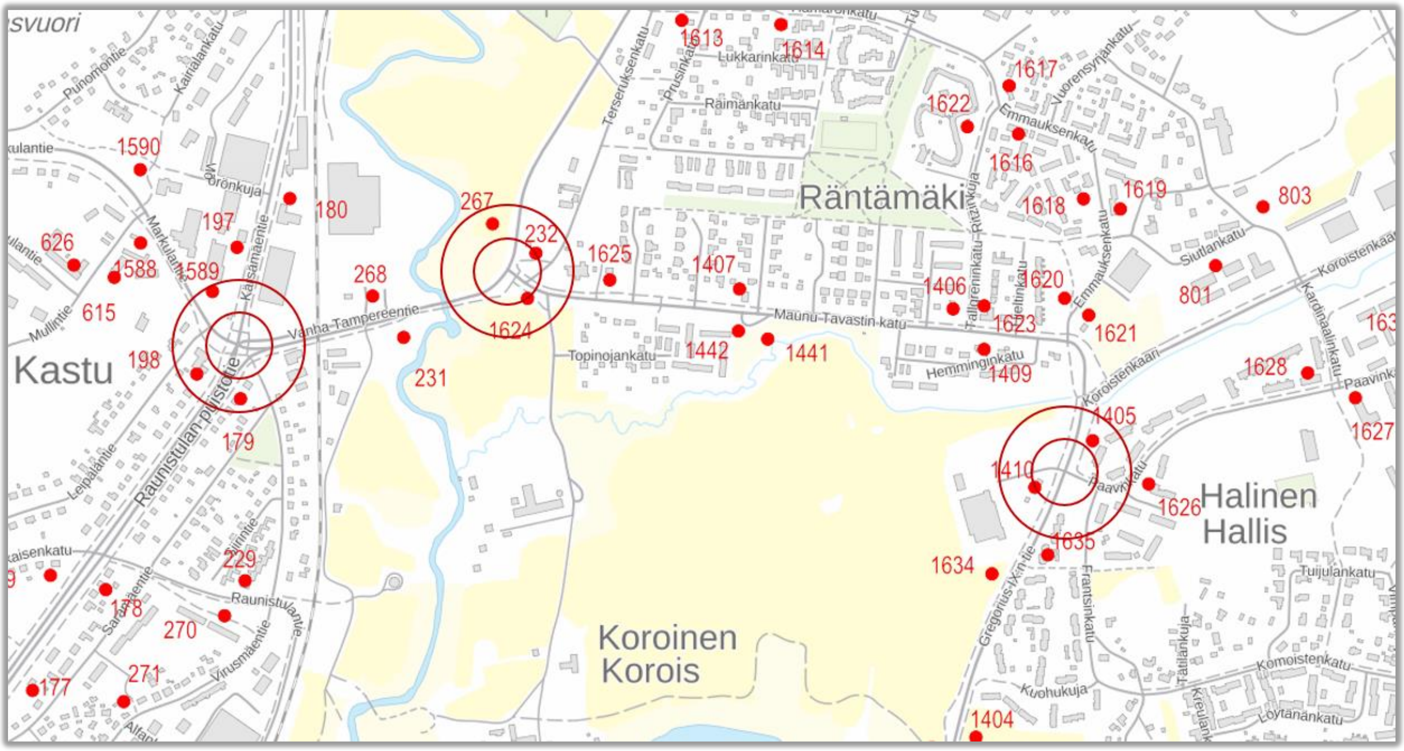

Figure 28: 100 and 200 metres circled around intersections part 2

For this experiment, the detectors will be placed at 60,90,120,150 and 180 metres from the intersection. The experiment will show whether it is more beneficial to place the detectors farther away from the intersection and how it affects what priority function will be used. Because of other intersections and bus stops in the simulation, it will not always be possible to move the detector farther away. These exceptional cases can be seen in Figure 27 and Figure 28, where the distances are marked with a circle around the intersection. One problem with the figures is that the bus stops are not always in the correct place, thus the moved detectors will be reviewed in detail below.

<u>60 metres</u>

Already at 60 metres, there is a detector that must be moved 5 meters closer. This detector is placed on the west side of intersection 405. The reason is that bus stop 978 in the east direction is located about 60m from the intersection.

<u>90 metres</u>

At 90 metres there are no new exceptions.

<u>120 metres</u>

At 120 metres the detector east of intersection 410 had to be moved a couple of metres because of a bus stop 1625. At intersection 333, the west side detectors had to be moved to 105 metres from the intersection because of bus stop 839. At intersection 420, the detector placed south of the intersection had to be moved because of bus stop 1635.

<u>150 metres</u>

For 150 metres there are no new exceptional cases.

<u>180 metres</u>

At 180 metres the detector west of intersection 334 had to be moved to 175 metres so that it would not interfere with intersection 333. The detector west of 338 had to be placed 150 metres from the intersection because of bus stop 455. At intersection 409, the detector west of the intersection had to be placed 150 metres from the intersection because of bus stop 268.

In the second experiment, the crossing traffic of 405 and 409 will be studied. This will show what effects the priority system has on crossing traffic. These intersections were chosen because they were the only ones where every detectors data were collected. Intersection 409 is also unique because it is one of the intersections that have extra green implemented. Therefore, one of the experiments will be without extra green implemented. The bus spawn rate was set to 7.5 minutes in each direction for this experiment to increase disruption from the priority request.

8. Results

For every experiment, there are five different simulation runs, which are 20 hours each (72 000 seconds). In the five runs, the simulation seeds 1010, 1251, 9532, 13003, 17023 were used. There is also a warm-up period of one hour in the simulation to fill the roads. The last hours of the simulation are also not included in the results because no new vehicles are being generated.

The simulation results from rush hours without priority is used as a baseline in the results. These results were used instead of the real data because they better show what the difference between having a priority implemented or not is. Another factor is that the actual traffic lights have offsets implemented, which makes them more efficient than the traffic lights in the simulation. The real values are still included in the results as a reference point, and it is important to keep in mind that the results are most likely more positive in the simulation than in the real world.

Before discussing the results, it should be noted that the bus stop duration changed between the different experiments, which resulted in shorter or longer bus routes than it would have been with the same bus stop duration. Therefore, the results are adjusted according to the difference. This is to make sure that the benefits come from the traffic lights instead of having shorter bus stops. In the experiments, the east bus duration grew, while the west bus stop duration shrank compared to the baseline in Table 12. All duration values are changed to take this into account, except for the standard deviation, which is not affected. For example, in 60 metres, 2.29 seconds were removed from all values in the east column to neutralise the difference in bus stop durations.

Another thing to note before doing more in-depth analysis about the result is that no time loss, which is at the bottom in Table 12, is the same for every single experiment. This means that in perfect conditions, the bus route time would always be the same. This is a relief because it means that there are no variables that make the bus route times different between the experiments. Furthermore, the results from no waiting in the traffic light is very similar, which shows that the difference in the duration comes from the priority system reducing the dwell time in traffic lights.

Table 12: Results part 1

Detector distance	Actual traffic		No Priority		60 m		90m		120m		150m		180m	
Direction	East	West	East	West	East	West	East	West	East	West	East	West	East	West
Count	538	466	411	435	411	435	411	435	411	435	411	435	411	435
Duration														
mean	14.67	15.66	15.34	15.78	14.13	14.12	14.00	13.90	13.86	13.58	13.77	13.49	13.68	13.29
Improvement							0.97 %	1.59 %	0.97 %	2.31 %	0.61 %	0.64 %	0.69 %	1.46%
std	1.7	2.00	0.90	1.14	0.60	0.64	0.63	0.56	0.57	0.54	0.64	0.56	0.54	0.54
min	9.1	9.22	12.85	12.53	12.61	12.41	12.63	12.36	12.46	12.07	12.44	12.13	12.46	11.88
0.25	13.6	14.40	14.65	15.03	13.71	13.66	13.56	13.51	13.44	13.26	13.29	13.10	13.30	12.91
0.5	14.6	15.62	15.25	15.75	14.14	14.07	13.98	13.89	13.84	13.54	13.74	13.45	13.61	13.23
0.75	15.7	16.89	15.92	16.50	14.55	14.56	14.38	14.22	14.20	13.91	14.11	13.88	13.99	13.68
max	20.3	21.72	17.57	19.38	16.41	16.16	16.52	15.76	15.93	15.22	16.14	14.97	16.06	14.91
No waiting in traffic lights														
mean	-	-	13.48	13.18	13.39	13.08	13.34	13.04	13.30	12.92	13.25	12.90	13.19	12.81
std	-	-	0.36	0.40	0.37	0.35	0.41	0.33	0.37	0.33	0.39	0.35	0.33	0.35
min	-	-	12.60	12.18	12.46	12.22	12.50	12.18	12.39	12.06	12.43	12.05	12.43	11.88
0.25	-	-	13.25	12.92	13.12	12.82	13.07	12.79	13.06	12.71	12.98	12.62	12.96	12.55
0.5	-	-	13.45	13.17	13.39	13.07	13.33	13.01	13.28	12.92	13.23	12.92	13.14	12.78
0.75	-	-	13.68	13.43	13.64	13.29	13.58	13.26	13.51	13.12	13.49	13.13	13.38	13.04
max	-	-	14.72	14.85	14.66	14.39	15.53	14.04	14.64	13.96	14.53	13.75	14.38	14.00
Duration in traffic lights														
mean (s)	-	-	111.7	156.1	44.7	62.8	39.1	51.6	33.7	39.1	31.7	35.6	29.5	29.2
Bus stop time														
mean (s)	141.1	116.6	141.0	117.4	141.6	107.1	142.0	105.9	142.5	103.0	143.5	103.5	142.3	102.6
std (s)	-	10.28	12.35	14.64	12.01	11.75	15.40	11.60	13.38	11.36	14.05	11.18	12.76	12.19

Table 13: Results part 2

Detector distance	Actual traffic data		No Priority Baseline		60 m		90m		120m		150m		180m	
Direction	East	West	East	West	East	West	East	West	East	West	East	West	East	West
No bus stop														
mean	12.32	13.72	12.99	13.82	11.77	12.34	11.63	12.13	11.49	11.86	11.38	11.76	11.31	11.58
std	-	-	0.80	1.00	0.49	0.55	0.48	0.47	0.47	0.45	0.51	0.47	0.43	0.44
min	-	-	10.73	10.62	10.57	10.96	10.57	10.86	10.31	10.62	10.26	10.63	10.24	10.58
0.25	-	-	12.40	13.19	11.42	11.96	11.28	11.81	11.16	11.56	11.01	11.42	11.01	11.28
0.5	-	-	12.90	13.78	11.76	12.31	11.60	12.11	11.46	11.82	11.34	11.73	11.28	11.56
0.75	-	-	13.48	14.48	12.11	12.71	11.96	12.43	11.79	12.12	11.69	12.10	11.58	11.85
max	-	-	15.03	16.97	13.41	14.16	13.23	13.71	13.29	13.41	13.21	13.03	13.13	12.96
No Time loss														
mean	-	-	10.18	9.80	10.18	9.80	10.18	9.80	10.18	9.80	10.18	9.80	10.18	9.80
std	-	-	0.21	0.24	0.20	0.20	0.26	0.19	0.22	0.19	0.23	0.19	0.21	0.20
min	-	-	9.77	9.26	9.69	9.27	9.72	9.34	9.73	9.38	9.68	9.36	9.75	9.34
0.25	-	-	10.04	9.64	10.03	9.67	10.01	9.66	10.02	9.67	10.02	9.67	10.03	9.66
0.5	-	-	10.16	9.77	10.16	9.78	10.15	9.79	10.15	9.78	10.16	9.80	10.16	9.78
0.75	-	-	10.31	9.95	10.32	9.92	10.32	9.92	10.32	9.91	10.33	9.92	10.30	9.93
max	-	-	11.20	10.45	10.85	10.34	11.58	10.41	10.92	10.75	10.99	10.49	10.76	10.39

First is the experiment with the detectors at 60 metres from the intersection. Already at 60 metres, there is a significant difference between having the priority implemented or not. In the east direction, the route time shrank by about 1 minute and 13 seconds (1.21 minutes), and in the west direction, the duration shrank by 1 minute and 40 seconds (1.66 min). This is a 7.9% improvement for the east direction and a 10.5% improvement for the west direction. When comparing the data to the actual data, then the improvements in duration becomes 32 seconds (3.7%) for the east and 1 minute and 32 seconds (9.8%) for the west. Compared to the actual data, the east direction's improvement shrank by a great deal, while the west's improvement is almost the same. The west bus route was always slower in the analysis chapter 6.1.2. Therefore, one explanation could be that the traffic lights favours the east direction when there is no priority function. Hence, the benefits in the west direction are more significant.

The results show that the total duration reduces as the detector is moved farther away. In Table 14, the improvement is shown by calculating the mean duration minus the previous detectors mean duration. From this, we can see that there is no logic to how the improvement develops as the detector is moved farther away. Most surprising is the result from 150 metres, where the west direction only improved by five seconds while every other distance improved by over 10 seconds. It was expected that the improvement would be more significant when the detectors are close to the intersection, but as the detector are moved farther away, the benefits would start to diminish.

Table 14: Duration reduction per 30 metres

Detector distance (m)	East (s)	West (s)
60->90	-8.20	-13.46
90->120	-8.18	-19.24
120->150	-5.11	-5.23
150->180	-5.72	-11.84

Why the results from 150 metres are not as good as the others is a mystery, but when studying the standard deviation, it can be seen that it has a higher standard deviation than both 120 metres and 180 metres. One explanation could be that a couple of buses was stuck in traffic in the 150 metres run, but the min and max values disprove that theory. The min and max values tell us that nothing unordinary happened during the 150 metres simulation because both values are similar to the other detector distances. This proves that even if the results improved as the detector was moved farther away, there is still a great deal of noise in the simulation runs and random variables that cannot be fully understood.

Interestingly the west duration continues to improve by much more than the east direction in Table 14, and at 120 and 180 metres, it improved by over double amount of the east duration. This indicates that the west direction is unfavourably configured in the traffic lights settings because there is much more room for improvement compared to the east direction. This can further be seen when studying the duration in traffic light rows in Table 12. From the baseline, the amount of waiting in traffic lights is 112 seconds for east and 156 seconds for the west. Already at 60 metres, the difference between east and west is much smaller but still significant.

Figure 29 shows how the duration at traffic lights develops as the detectors are moved farther away. Here, the diminishing effects are apparent for the east direction as the inclination of the line is shrinking as the detector is moved farther away. For the west direction, the diminishing effect is not as apparent. According to Figure 29, the west direction spent much more time at traffic lights when the detector was close to the intersection. At 180 metres, the durations at traffic lights are equal. Therefore, the improvements are probably negligible for both after that. No farther tests were done because at 180 metres the detector lengths were already impossible at many intersections of the route. Implementing the priority function (60m) reduced the waiting time at traffic lights by 60% in both directions. After that, the duration at traffic lights shrank by 13.5% on average for every 30 meters.

The reason the west direction spends more time at traffic lights is due to how the traffic light phases are defined. Hence, the traffic lights are more beneficial for the east direction in the simulation. One reason could be because at intersection 338 and 410; the east direction has two phases at which it is green while the west direction only has one. The reason for this is that when the east direction is turning left in 338, the west direction cannot have a green light. The phases were designed according to the traffic configuration in the real world. However, it is impossible to prove if the difference between west and east direction in the real world comes from this. The reason for that is that the traffic lights in the real world are adaptive. Therefore, the effects of configurations cannot be compared to the simulation. It likely has something to do with the traffic light configurations, but also other things such as congestion.

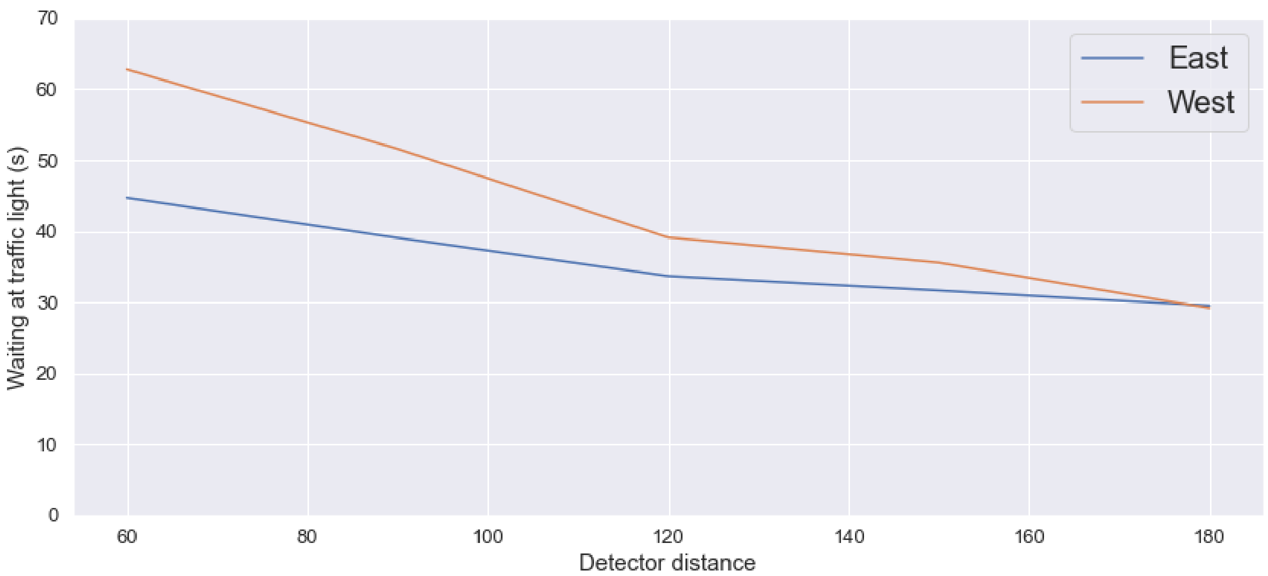

Figure 29: Total duration at traffic lights

According to Table 12, the standard deviation shrank by 33.3-40.0% for the east direction and by 43.9-52.6% for the west direction. Some of the standard deviation in the simulation comes from the bus stop duration because the stop time standard deviation is around 10-15 seconds for all the experiments. The big surprise is that the standard deviation is always very similar no matter where the detector is placed. The standard deviation was expected to decrease as the detector is moved farther away. This is because the current phase should matter less as the detector is moved farther away. The reason is that the priority request is sent earlier, which means that the bus should have a better chance to catch a green light.

Table 15: Standard deviation of duration

Detectors	East	West	Sum
Baseline	54.12	68.19	87.05
60	36.10	38.60	52.85
90	37.58	33.41	50.28
120	34.34	32.54	47.31
150	38.59	33.81	51.31
180	32.39	32.47	45.86

In Table 15, the sum of both directions standard deviation is shown. There it can be seen that the standard deviation shrank in all distances, except for 150 metres. However, the improvement was not significant. When studying the directions separately, the standard deviation goes both up and down with each experiment. This shows that many factors affect the standard deviation. Therefore, it is not possible to conclude that the standard deviation reduces as the detectors are moved farther away. However, there is a significant difference between having the TSP-system implemented or not.

Reaching a similar standard of deviation in the real world as in the simulations is not possible because of the varying lengths in bus stop durations. Furthermore, the simulation has a constant flow of traffic, while in the real world, the number of cars on the route might change significantly in just 15 minutes. For example, the roads close to schools are likely to be very congested before the school lessons are about to begin. This and many other things can cause variances in the route duration that do not exist in the simulation.

Table 16 shows how long it takes to travel either 60, 90,120,150 or 180 metres depending on the speed. In most cases, the bus should be travelling at around 40-50 km per hour if there is not heavy congestion. Table 7, in the validation chapter, showed that most of the traffic travels at around 45 km per hour in the simulation. At this speed, it only takes 2.4 seconds to travel 30 metres, but interestingly the 2.4 seconds have a significant effect on the bus duration times. Suppose there is a detector at 60 metres and 120 metres. The detector at 120 metres will know about the presence of a bus 4.8 seconds earlier than the detector at 60 metres. Now, if the current phase at the upcoming traffic light is red in the bus direction and the phase has four or more seconds left until it reaches it minimum phase time, then the detector length becomes insignificant. However, if the traffic light has already passed the minimum phase time, then the detector at 120 metres can start the next phase 4,8 seconds earlier, and the bus should gain green about five seconds earlier compared to having the detector at 60 metres. Additionally, these five seconds might mean that the queue can start moving before the bus reaches the intersection. Thus, the bus never has to stop. If the phase would have been green in the example, then the chance that there is an extension increases significantly.

In the real world, one cycle is 75 seconds in this route, but in the simulation, they are slightly longer because SYVARI could not be implemented. In 75 seconds, the four seconds can already make a big difference. Suppose the green phase is on average 25 seconds, then the chance to arrive at the green phase would be 33.33%, but adding four seconds to the green phase already increases the chance by 5.34 percentage units.

Table 16: Traveling speed

km/h	m/s	Time to travel (s)				
		60m	90m	120m	150m	180m
20	5.56	10.80	16.20	21.60	27.00	32.40
25	6.94	8.64	12.96	17.28	21.60	25.92
30	8.33	7.20	10.80	14.40	18.00	21.60
35	9.72	6.17	9.26	12.34	15.43	18.51
40	11.11	5.40	8.10	10.80	13.50	16.20
45	12.50	4.80	7.20	9.60	12.00	14.40
50	13.89	4.32	6.48	8.64	10.80	12.96
55	15.28	3.93	5.89	7.85	9.82	11.78

Next, some t-test will be performed to check whether the simulation results are just random chance or if there is a significant improvement. The first test will be if implementing a TSP system makes a significant difference. Here the no priority values are compared to the 60 metres values from Table 12.

$$\text{Null hypothesis: } \mu 1 = \mu 2$$
$$\text{Alternative hypothesis: } \mu 1 != \mu 2$$
$$\alpha = 0.05$$

<u>East values</u>

$$\mu 1 = 920.267639902676, \quad \sigma 1 = 54.1178154609922, \quad n1 = 411$$
$$\mu 2 = 848.5839416, \quad \sigma 2 = 36.09755934, \quad n2 = 411$$

West values

$$\mu 1 = 946.806896551724\,, \quad \sigma 1 = 68.1884718575175, \quad n1 = 435$$

$$\mu 2 = 836.983908\,, \quad \sigma 2 = 38.60436472, \quad n2 = 435$$

P-value

$$p - value\ east = 5.7437141667491475e^{-86}$$

$$p - value\ west = 2.565634501759854e^{-131}$$

In the first t-test, both directions scored a lower value than alpha. Therefore, there is a clear benefit in implementing a TSP system. The p-values from the tests indicate that there is almost a zero per cent chance of these results happening at random.

Next, a t-test will be done on the results from 60 metres and 90 metres to see if there is a significant difference when moving the detector 30 metres farther away from the intersection.

$$\text{Null hypothesis: } \mu 1 = \mu 2$$

$$\text{Alternative hypothesis: } \mu 1! = \mu 2$$

$$\alpha = 0.05$$

East values

$$\mu 1 = 848.5839416, \quad \sigma 1 = 36.09755934, \quad n1 = 411$$

$$\mu 2 = 840.8199513, \quad \sigma 2 = 37.57803836, \quad n2 = 411$$

West values

$$\mu 2 = 836.983908\,, \quad \sigma 2 = 38.60436472, \quad n2 = 435$$

$$\mu 1 = 822.3149425, \quad \sigma 1 = 33.40609829, \quad n1 = 435$$

P-value

$$p - value\ east = 0.002600091008874209$$

$$p - value\ west = 3.0196006183755145e^{-9}$$

Both p values are below 0.05, which means that moving the detectors from 60 metres to 90 metres has a significant effect on the results that cannot be explained by randomness. This shows that moving the detector farther away from the intersection is beneficial.

The results from earlier showed that the improvement reduced as the detectors were moved farther away. Therefore, a final t-test will be done on 150 to 180 metres to see if there is a significant improvement. This test is important because it will show whether the improvement has already diminished to the point that there is no real benefit in moving the detector farther away.

$$\text{Null hypothesis: } \mu 1 = \mu 2$$

$$\text{Alternative hypothesis: } \mu 1! = \mu 2$$

$$\alpha = 0.05$$

<u>East values</u>

$$\mu 1 = 828.978102189781, \quad \sigma 1 = 38.5904323704188, \quad n1 = 411$$

$$\mu 2 = 822.121654501216, \quad \sigma 2 = 32.3876639145418, \quad n2 = 411$$

<u>West values</u>

$$\mu 2 = 795.457471264367\,, \quad \sigma 2 = 33.8088951298933, \quad n2 = 435$$

$$\mu 1 = 782.75632183908, \quad \sigma 1 = 32.472965363938, \quad n1 = 435$$

<u>P-value</u>

$$p - value\ east = 0.005926244611521453$$

$$p - value\ west = 2.163920182677161e^{-8}$$

As it could be seen in the earlier results, the benefits were clearly shrinking as the detectors were moved farther away, but the t-test shows that there is still a significant benefit in moving the detector from 150 metres to 180 metres. Here it would have been interesting to move the detectors even farther away to see where the effect stops being significant. However, as stated earlier, that was not possible due to many detectors already being placed at max distance.

Table 17 shows the number of times the different priority methods were used. The extra green simply means the number of times the intersections gave an extra green. The intersections with extra green implemented were 334 and 409. Here, the per intersection value was divided by both two intersections and eight intersections. Red phases shortened counts how many times a phase was shortened. Because the traffic lights in the simulation have two to four phases implemented, it means that buses can shorten up to three phases in a single intersection. Red shortened intersections instead count how many buses had red shortenings at the intersection, hence even if several phases are shortened for a single bus, it still counts as one per intersection. The extensions rows calculate how many extensions were given. The values gathered in Table 17 did not contain a timestamp. Therefore, the first and last hours of the simulation had to be included in these results. This means the per bus row had to be divided by the total number of buses, which was 933. The total amount of intersections was eight.

Table 17: Priority requests statistics

	60m	90m	120m	150m	180m
Extra Green					
Total	413	449	450	457	432
Per bus	0.4427	0.4812	0.4823	0.4898	0.4630
Per bus and intersection (2)	0.2213	0.2406	0.2412	0.2449	0.2315
Per bus and intersection (8)	0.0553	0.0602	0.0603	0.0612	0.0579
Red phases shortened					
Total	4776	4954	4865	4949	4763
Per bus	5.1190	5.3098	5.2144	5.3044	5.1050
Per bus and intersection	0.6399	0.6637	0.6518	0.6630	0.6381
Red shortened intersections					
Total	3321	3482	3419	3436	3340
Per bus	3.5595	3.7320	3.6645	3.6827	3.5798
Per bus and intersection	0.4449	0.4665	0.4581	0.4603	0.4475
Extensions					
Total	963	1145	1397	1519	1866
Per bus	1.0322	1.2272	1.4973	1.6281	2.0000
Per bus and intersection	0.1290	0.1534	0.1872	0.2035	0.2500
Increase %		18.9%	22.0%	8.7%	22.9%
Statistics					
Number of red phases shortened per priority	1.4381	1.4227	1.4229	1.4403	1.4260

The results from the extra green and red shortened intersections show that the number of priorities given reached their peak value when the detectors were placed 150 metres away from the intersection. While the results from 150 metres did not follow the same pattern as the others in the previous results, it should not affect these results. That is because the bus should always have been able to drive through the intersection on the first green light. Otherwise, the maximum route duration value would have been significantly higher in Table 12 for 150 metres.

At 180 metres, both the extra green and red shortening values are almost identical to the results from 60 metres. This is because the extensions kept increasing as the detectors were moved farther away, and after 150 metres, the constant increase in extensions starts decreases the other priority types.

The increase of the extensions was expected, as it is only logical to happen as the detectors are moved farther away. However, the sudden decrease in the other priority types after 150 metres was not anticipated. This is highly informative because extensions are the type of priority that causes the least disturbance in the other traffic. Therefore, traffic planners always strive to maximise the number of extensions instead of red shortenings or extra greens.

The per bus and per intersection rows describe how likely it is for one bus to gain that type of priority in each intersection. Surprisingly every priority method seems to be quite common. For the extra green, there is a 22-24% chance that a bus will gain extra green when approaching the intersection. Overall, the difference is small, with just 2.3 % difference between the smallest and largest value.

The red shortened intersection has an even smaller difference when looking at the percentages compared to extra greens. Here the difference between the largest and smallest value is roughly 1.8 %, which occurs between 60 and 90 metres.

The extension results are very positive. Moving the detectors from 60 metres to 180 metres almost doubled the number of extensions given. Additionally, many of the detectors had to be placed closer than 180 metres away. Hence, the result would have been even better if they all were at 180 metres away from the intersection.

In Table 18, the crossing traffic statistics are shown from intersection 405 and 409. All values are in seconds, except for the count data. The duration simply describes how long the whole route took and the depart delay describes if there was any delay when spawning the vehicle. Depart delay was included because if the road in SUMO is congested, then the vehicle does not spawn. This is important to include in the analysis because it can happen if the queue at the intersection becomes too long.

Table 18: Disturbance on crossing traffic in intersection 405 and 409 at detector length 150m (s)

	No Priority		-	Extra green		-	No extra green	
	405	409		405	409		405	409
Duration								
count	108490	45390		-	45390		108490	45390
mean	44.46	49.09		-	50.58		44.72	49.97
std	24.08	23.32		-	24.16		23.64	23.26
min	9	15		-	14		9	15
25 %	21	26		-	27		23	27
50 %	44	47		-	49		44	48
75 %	66	69		-	71		65	70
max	94	103		-	129		114	116
Depart delay								
count	108490	45390		-	45390		108490	45390
mean	-0.45	-0.47		-	-0.47		-0.45	-0.47
std	0.38	0.31		-	0.31		0.38	0.31
min	-1	-1		-	-1		-1	-1
25 %	-0.74	-0.74		-	-0.74		-0.74	-0.74
50 %	-0.47	-0.47		-	-0.47		-0.47	-0.47
75 %	-0.21	-0.21		-	-0.21		-0.21	-0.21
max	6.79	2		-	2		7.64	2

First, the values from no priority and no extra green will be analysed in Table 18. These results show that in both intersections, the difference between having a priority system or not has little effect on the mean duration. For 405, the mean only increased by 0.26 seconds (0.6%), and for 409, it increased by 0.88 seconds (1.8%). The fourth quartile of the duration shows an increase of 1 second for both intersections.

The results from the extra green shows also show that there is no significant difference in having the extra green implemented or not. The duration for 409 increased by 0.61 seconds (1.2%) on average, when extra green is implemented. When compared to the no priority simulations, the total increase became 1.49 seconds (2.9%). The quantile numbers also show a minimal increase except for the max value, which increased to 129 seconds from 116 seconds.

The same analysis was also done by using the third quartile of the traffic data instead of the median. The results from those test showed that the difference between having a TSP system or not is insignificant. The results can be found in Table 21 in the appendices. Based on this, it can be concluded that the TSP system causes an insignificant disturbance on the crossing traffic.

9. Conclusion

The research question for this paper was whether a TSP system would be effective in Turku. According to the results, there is a significant improvement in the bus route duration from implementing a TSP system. However, the reliability of the results can be questioned. The duration improved by 7.9-10.8% for the east direction and 10.5-15.8% for the west direction. The difference comes from how far the detectors were placed from the intersection. The total route duration continued to shrink as the detector was moved farther away from the detector. According to Table 1, these results are within the average improvement seen in other cities after implementing a TSP system. Still, the results are unlikely to be that impressive in Turku because the traffic lights were not configured identically according to the real world. This was because it was not possible to add split phasing in SUMO, which means that the directions can be configured individually. Therefore, the bus route duration from the baseline simulation is longer than it would be in the real world. This could be seen in the baseline bus route duration as the east duration was almost a whole minute longer than the actual duration. However, the west duration was almost identical. Comparing the results to the actual data shows a 3.7-6.7% improvement to the east and 9.8-15.13% to the west, which is probably more realistic.

The next researched theme in this paper was whether moving the detectors farther away from the intersection improved the duration. Here, the results should be accurate since they were measured against simulations with TSP system implemented. The results showed that there is a significant improvement in moving the detectors farther from the intersection, even at 180 metres. The duration improved by 0.61-0.97% for the east direction throughout all distances and 0.64-2.31% for the west direction. In both cases, the lowest value came from detector distance 150 metres. This showed that even if the random variables were minimised in the simulation, the simulation still contained some randomness. The result from which priority type was used was even more impressive. The number of extensions rose by around 20 % for every thirty metres except at 150 metres. After 150 metres, the number of other priority types also started to shrink. This shows that when possible, it is always worth placing the detectors as far away from the intersection as possible.

The fourth research question was whether the crossing traffic is affected by the TSP system? According to the simulation, the TSP system should have almost no effect on the crossing traffic. This was consistent with what the other studies researched in this paper showed. Furthermore, the buses were set to arrive every 3.75 minutes at the intersection, which is higher than in the real world. Thus, the results from the simulation are worse than they would be in the real world

10. Practical conclusion

This conclusion will describe my experience of working with SUMO. The research questions that will be answered here is: Should simulation simulations be used more in traffic planning in Turku even if there is a lack of data? Because SUMO was used in this paper, the experience with SUMO will first be discussed. Secondly, the thoughts about using other simulation software within Turku city will be discussed.

As stated earlier, SUMO is open software, which means that it is free to use, while other simulation programs can be up to 50 000 euros for a single license. Because of this, SUMO could potentially save a substantial amount of expenses for the company or government. However, there are some downsides with it being open-source that must be discussed.

First, I will explain shortly what experiences I previously have. I have about one year of simulation experiences within process simulation and system dynamics. Previously I have built macroscale traffic simulations. I have three years of programming experiences with python, SQL, R and java and would rank myself as an intermediate programmer within python. Which means that I understood the functions in SUMO, but had trouble understanding where the problem was at times. This was extremely frustrating because the documentation for SUMO is often lacking, and the documentation was incorrect at times.

The greatest quality with open-source software is that one is able to modify almost everything within the simulation. When studying other simulations, the capabilities of SUMO was often shown. Many papers built systems within the simulation program that might have been impossible with other simulation software. However, I have not used other traffic simulation software, and can not commentate on what their capabilities are. Nevertheless, building the priority system was relatively simple in SUMO. It did require having a basic understanding of how TraCI (Traffic Control interface) works, but once the basics were understood, the tool becomes simple. TraCI is used to change parameters in the simulation when the simulation is running. In this simulation, TraCI was used to gather loop data (detector), then analyse the loop data for buses and apply the correct priority function to the upcoming intersection. The best thing with TraCI was that everything was written in python code. This means that there are a ton of functions within python that can be implemented to the simulation.

SUMO has applied functions such as map-generating into their system, which makes it effortless to create the simulation. However, the map was often inaccurate, and many things had to be corrected. The function did offer capabilities to modify how the maps were converted, but these lacked documentation.

Next, some of the negative sides of SUMO will be discussed. Compared to other simulation programs, SUMO requires the user to program almost everything to build a simulation. This can be great because it means that the simulator can modify everything in the simulation, but sadly the documentation was often lacking. For someone who had never used XML before, this was very time-consuming. However, having the option to change numerous parameters is a positive feature, even if it is time-consuming. Most of the parameters also have reasonable default values, which means that nothing has to be changed. It is recommended that all attributes are learned when building the simulation, in case some default settings should be turned off.

The largest disappointed with SUMO was the traffic light program. The problem was that in SUMO, the traffic lights must have phases. These phases can have an adaptive length, but the directions are still bound to a specific phase. This meant that the system used in Finland with split-phases could not be replicated in SUMO. While it is possible to code a similar system in SUMO, it would require a great deal of work and knowledge, considering the whole traffic light system would have to be programmed from the beginning within SUMO. In Vissim, this could be done relatively quickly because it has a ready-built function that allows the simulation to connect an outside program to change the traffic lights. However, SUMO does not offer such a function.

Not being able to implement split-phasing in the simulation meant that the simulation was never a valid representation of the real-world system. Therefore, the results can be questioned because the traffic lights were not working correctly. However, this was taken into consideration throughout the results.

Overall, the experience with SUMO was mostly positive. The simulation program offered most of the things that this paper required and implementing functions was time-consuming but straightforward. One of the problems with SUMO was that finding examples was difficult. Most scientific papers do not upload their simulation to the web, which means that the solutions can not be found. Another problem was that the SUMO documentation was lacking, which makes it hard to find a solution when a problem occurs.

In conclusion, I do not believe that SUMO is the correct software to use in a governmental traffic planning department because of its complexness. While the software is relatively simple, every part of building a simulation requires a great deal of work. Another reason is the number of programming skills the software requires. I was able to understand most things in the simulation, but it took hundreds of hours. Due to the lack of documentation and steps needed to build a simulation, it is not recommended that SUMO is used in a governmental traffic planning department. The software would require someone knowledgeable to work there. Furthermore, if the person who built the simulation were to leave, then it becomes difficult for the next one to continue with the same project. Additionally, the traffic light settings alone are a reason I can not recommend SUMO to Turku city because without having correctly configured traffic lights, the validity of the simulation can always be questioned.

The research question about whether simulations should be used more in traffic planning is complicated to answer. This is because other traffic simulation software were not tested. Therefore, it is impossible to evaluate how time-consuming they are. There are still many benefits with using simulations for testing a new system and variables because both are time-consuming and expensive to test in the real world. However, the results from a simulation can always be questioned. Therefore, the organisation must have accurate and enough data. In this study, there was a sufficient amount of data to create a realistic simulation. The missing data were vehicle traffic, which existed, but was too time-consuming to extract. Furthermore, the data were sometimes inaccurate because some detectors did not work correctly, and the bus GPS was not necessarily always accurate.

As stated earlier, simulations are already being used in Turku city for traffic planning, but these are mostly small-scale simulations with just a few intersections. Many of the problems that arose in this paper do not exist in small scale simulations. A small-scale simulation is less time consuming, and the validity process becomes much more straightforward as there are fewer factors that affect the results. Therefore, Turku city should continue to use small scale simulations to optimise traffic in the future.

For large scale simulations, the answer depends on what is being studied. One should strive to keep the simulation simple instead of extensive to make the most accurate simulation and to minimise the workload. The simulation in this paper was more extensive because the effect of implementing a TSP system was studied over many intersections instead of just one. However, there are many uses for large scale simulations. For example, if bus routes are studied, then a large-scale simulation is a must. Additionally, a large-scale simulation is great for designing the road network. These simulations can have enormous benefit in the planning phase. The problem is that a large-scale simulation takes a long time to build. Therefore, large scale simulation can be useful for large new projects to save potential expenses that could have been avoided. However, large scale simulations should be avoided as they are very time-consuming and does not necessarily reflect reality.

11. Swedish summary: Effektivitetsanalys genom simulation av bussprioritetssystem i Åbo stad

Avhandlingen "*Efficiency analysis of transit signal priority using microsimulations: A case study in Turku",* analyserar ifall det är gynnsamt att investera i ett TSP-system i Åbo. Ett TSP-system är ett bussprioritetssystem som ger prioritet åt bussar vid trafikljus. Genom att använda bussprioritetssystem kan man minska ruttens resetid. Detta gör bussar till ett lockande alternativ till privatfordon. Ifall flera människor använder buss betyder det även att trafikstockningen i staden minskar. I studien utforskas också ifall det är lönsamt att öka avståndet mellan bussdetektorerna och trafikljusen. Detektorerna utlöser en prioritetsbegäran åt trafikljusen och ifall detektorerna är placerade längre bort från korsningen så får trafikljuset tidigare en signal om bussen. Dessutom analyseras effekten bussprioritetssystemet har på den korsande trafiken och ifall bussprioritetssystemet inverkar negativt på den övriga trafiken.
Denna undersökning görs genom att bygga en simulation av busslinjen 99, som går ytterom Åbo centrum. Rutten som simuleras är ca 6,4 km lång och innefattar totalt 8 korsningar där ett bussprioritetssystem kan implementeras.

Åbo stad har nyligen investerat i ett nytt betalningssystem inom lokaltrafiken som innehåller ett GPS-system. Därför är de nu intresserade av att använda denna samma GPS-teknologi för att skapa ett bussprioritetssystem, men är tveksamma ifall GPS-signalen är tillräckligt pålitlig. Efter att ha analyserat andra bussprioritetssystem runtom i världen, kan det konstateras att GPS-precisionen inte är perfekt och signalen har en liten fördröjning, men detta är inget hinder för att skapa ett funktionellt bussprioritetssystem.

Studien utfördes genom att göra en kvantitativ dataanalys. Data för analysen kommer från Åbo stads interna system. Den första delen av data är från bussloggar som sparas då bussen anländer och avgår från en busshållplats. Denna data hämtades från bussavdelningen inom Åbo stad (Föli) databas genom att använda MySQL workbench. Ett problem med data är att databasen klarar endast av att spara de senaste fyra månaderna, men det är tillräckligt för att bygga en trovärdig simulation. Med hjälp av bussloggarna analyserades tiden som bussarna stannar på busshållplatserna och bussruttstiden.

Den andra data som användes i simulation samlades genom induktionsdetektorer som ligger längs vägen vid korsningar. Denna data berättar hur många fordon som kör över detektorn per timme. Detektordata var mera besvärlig att hämta då den har ett eget program där endast data från en detektor kan hämtas åt gången. På grund av att det var så tidskrävande att hämta denna data så hämtades endast trafikdata i bussens riktning. Dock är detta inte ett problem då endast bussruttstiden undersöks och för att den skall vara trovärdig krävs endast trafikdata i bussriktningarna.

För att bygga simulationen gjordes även expertintervjuer med trafikplaneringsavdelningen vid Åbo stad och relevanta personer som var anställda på bussavdelningen. Det gjordes även en intervju med Swarco, vilket är trafikljusleverantören i Åbo stad. Dessa intervjuer krävdes för att få mera informationen om hur simulationen skulle byggas.
Dataanalysen visade vissa intressanta egenheter som finns inom buss rutten. Till exempel framkom det att bussen som går mot öst alltid är snabbare än den som går mot väst. Detta är intressant eftersom vanligtvis går morgonrusningen i en riktning och kvällsrusningen i den motsatta riktningen. Denna skillnad i resetiden kunde inte replikeras i simulationen fastän trafikmängderna är identiska, vilket tyder på att skillnaden uppstår av hur trafikljusen är konfigurerade i verkligheten.

Analysen visar även att bussen inte stannar vid busshållplatserna i majoriteten av fallen. Då bussen stannar på busshållplatsen är mediantiden för stoppet mellan 15–20 sekunder för alla busshållplatser i denna rutt. Dock varierar variansen en stor del mellan busshållplatserna. För vissa busshållplatser så är tredje kvartalet nästan exakt samma som medianen medan för andra är den 50 % högre än medianen. Detta betyder att även om bussprioritetssystem minskar betydligt på tiden bussar står vid trafikljus, så kommer det att finnas en stor del varians mellan bussarna på grund av stoppen vid busshållplatserna.

I byggandet av simulationen har många förenklingar gjorts eftersom simulationen programmet SUMO inte klarade av att replikera hur trafiken i Finland fungerar. En av förenklingarna var hur trafikljusen fungerar i simulationen. I Finland används ett trafikljusschema som kallas för ”split-phasing”, vilket betyder att man kan definiera varje riktning skilt i trafikljuset. Detta betyder att en riktning kan ha grönt för till exempel fem sekunder medan en annan riktning har samtidigt grönt för 15 sekunder. I SUMO måste trafikljusen däremot definieras i så kallade skeden (phases), vilket betyder att riktningarna inte går att definieras individuellt. Detta betyder att trafikljusen i simulationen är definierade annorlunda i jämförelse med verkligheten. Detta var ett undantag som inte gick att undvika i simulation, dock klarade västra riktningen i simulationen av t-testet i valideringsprocessen fastän trafikljusen inte var identiska i jämförelse med verkligheten. Den östra riktningen klarade dock inte av t-testet på grund av hur trafikljusen var konfigurerade, men det beaktades i resultatet.

Ett annat undantag i SUMO var att det endast är möjligt att definiera ett statiskt värde för busshållplatstiden. Detta betyder att standardavvikelsen i simulation är betydligt mindre än den vore i verkligheten, vilket beaktades då man analyserade resultaten.

Enligt resultaten sjunker den totala resetiden med 7,8–10,8 % för bussen som går mot öst och med 10,5–15,8 % för bussen som går mot väst. Den lägre gränsen är då detektorn är placerad 60 meter från korsningen, medan den övre gränsen är när detektorn är 180 meter från korsningen. Resultaten visade att genom att flytta detektorn 30 meter bakåt så kan man redan se en signifikant förbättring i busruttens resetid.

Då detektorerna flyttades från 60 meter till 180 meter, så fördubblades även mängden förlängningar av det gröna ljuset. Detta är mycket positivt eftersom förlängningar skapar minst störning för både trafikljuscykeln och för andra fordon i jämförelse med de andra prioritetsfunktionerna. Experimenten visade även att de andra prioritetsfunktionerna blev mindre frekventa då detektorn placerades över 150 meter från korsningen. Det konstaterades även att TSP-systemet har knappt någon effekt på den korsande trafiken, då deras rutt tog endast 0,26–1,49 sekunder längre (0,6–2,9 %).

I avhandlingen undersöktes även frågan ifall simulationer borde användas mera inom Åbo stad även om det ibland finns brist på kvantitativa data. Därför beskriver avhandlingen i detalj alla steg som krävdes för att bygga denna simulation och vilka problem som uppstod. I slutsatsen rekommenderas det att staden fortsätter använda småskaliga simulationer för att testa nya projekt. Däremot rekommenderas det att man undviker storskaliga simulationer eftersom de är mycket tidskrävande och deras resultat kan ofta ifrågasättas.

Bibliography

Albright, E., & Figliozzi, M. (2012). Factors influencing effectiveness of transit signal priority and late-bus recovery at signalized-intersection level. *Transportation Research Record*, *2311*, 186–194. https://doi.org/10.3141/2311-18

Arellano, W., & Mahgoub, I. (2013). TrafficModeler extensions: A case for rapid VANET simulation using, OMNET++, SUMO, and VEINS. *2013 High Capacity Optical Networks and Emerging/Enabling Technologies, HONET-CNS 2013*, 109–115. https://doi.org/10.1109/HONET.2013.6729767

Barcelo, J. (2010). *Fundamentals of Traffic Simulation (International Series in Operations Research & Management Science)* (F. S. Hillier & C. C. Price, Eds.; 145th ed.).

Barnham, C. (2015). Quantitative and qualitative research: Perceptual foundations. *International Journal of Market Research*, *57*(6), 837–854. https://doi.org/10.2501/IJMR-2015-070

de Oña, J., de Oña, R., Eboli, L., & Mazzulla, G. (2013). Perceived service quality in bus transit service: A structural equation approach. *Transport Policy*, *29*, 219–226. https://doi.org/10.1016/j.tranpol.2013.07.001

Doucet, C. (2008). *Commuter Plans in CIVITAS MOBILIS cities Commuter Plans and Innovative Mobility Solutions*.

D'Souza, C., Gardner, K., Hounsell, N. B., & Shrestha, B. P. (2010). *New developments for bus priority at traffic signals in London using iBus*. 72–72. https://doi.org/10.1049/cp.2010.0386

Ejercito, P. M., Nebrija, K. G. E., Feria, R. P., & Lara-Figueroa, L. L. (2018). Traffic simulation software review. *2017 8th International Conference on Information, Intelligence, Systems and Applications, IISA 2017*, *2018-January*, 1–4. https://doi.org/10.1109/IISA.2017.8316415

European GNSS Agency. (n.d.). *GNSS raw measurements delivering greater accuracy | European GNSS Service Centre*. Retrieved September 2, 2020, from https://www.gsc-europa.eu/news/gnss-raw-measurements-delivering-greater-accuracy

Gardner, K., D´Souza, C., Hounsell, N., BShrestha, B., & Bretherton, D. (2009). *Review of Bus Priority at Traffic Signals around the World*.

Gardner, K., D´Souza, C., Hounsell, N., Shrestha, B., & Bretherton, D. (2009). *Review of Bus Priority at Traffic Signals around the World*. http://content.tfl.gov.uk/interaction-of-buses-and-signals-at-road-crossings.pdf

Gentile, G., Florian, M., Hamdouch, Y., Cats, O., & Nuzzolo, A. (2016). The Theory of Transit Assignment: Basic Modelling Frameworks. In *Modelling Public Transport Passenger Flows in the Era of Intelligent Transport Systems* (Issue 9783319250809, pp. 287–384). Springer International Publishing. https://doi.org/10.1007/978-3-319-25082-3_1

Goertzen, M. J. (2017). *Applying Quantitative Methods to E-book Collections: Vol. Vol 54, No 4*. https://journals.ala.org/index.php/ltr/issue/view/640

Goh, K. C. K., Currie, G., Sarvi, M., & Logan, D. (2014). Bus accident analysis of routes with/without bus priority. *Accident Analysis and Prevention*, *65*, 18–27. https://doi.org/10.1016/j.aap.2013.12.002

Goh, K., Currie, G., Sarvi, M., & Logan, D. (2013). Road safety benefits from bus priority. *Transportation Research Record*, *2352*, 41–49. https://doi.org/10.3141/2352-05

GPS.GOV. (2020). *GPS Accuracy*. https://www.gps.gov/systems/gps/performance/accuracy/

Helbing, D. (2008). From microscopic to macroscopic traffic models. In *A Perspective Look at Nonlinear Media* (pp. 122–139). Springer Berlin Heidelberg. https://doi.org/10.1007/bfb0104959

Horni, A., Nagel, K., & Axhausen, K. W. (2016). *The Multi-Agent Transport Simulation MATSim*. https://www.ubiquitypress.com/site/books/e/10.5334/baw/

Hounsell, N. B., Shrestha, B. P., Head, J. R., Palmer, S., & Bowen, T. (2008). The way ahead for London's bus priority at traffic signals. *IET Intelligent Transport Systems*, *2*(3), 193–200. https://doi.org/10.1049/iet-its:20070060

Hu, J., Park, B. B., & Lee, Y. J. (2016). Transit signal priority accommodating conflicting requests under Connected Vehicles technology. *Transportation Research Part C: Emerging Technologies*, *69*, 173–192. https://doi.org/10.1016/j.trc.2016.06.001

Inkiläinen, J. (2012). *Reaaliaikainen matkustajainformaatiojärjestelmä*. https://core.ac.uk/download/pdf/38053883.pdf

Korvenmaa, M. (2020). *Personal interview*.

Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker-Walz, L. (2012). Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, *3 & 4*.

Lee, Y.-J., Dadvar, S., Hu, J., & Park, B. B. (2017). Transit Signal Priority Experiment in a Connected Vehicle Technology Environment. *Journal of Transportation Engineering, Part A: Systems*, *143*(8), 05017005. https://doi.org/10.1061/JTEPBS.0000062

Li, L., Persaud, B., & Shalaby, A. (2017). Using micro-simulation to investigate the safety impacts of transit design alternatives at signalized intersections. *Accident Analysis and Prevention*, *100*, 123–132. https://doi.org/10.1016/j.aap.2016.12.019

Liikennevalot.info. (n.d.). *HELMI - älykäs liikennepalvelu*. Retrieved September 9, 2020, from http://www.liikennevalot.info/historia/helminalku.shtml#jalkiarviointia

Lim, K. G., Lee, C. H., Chin, R. K. Y., Beng Yeo, K., & Teo, K. T. K. (2017). SUMO enhancement for vehicular ad hoc network (VANET) simulation. *Proceedings - 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems, I2CACIS 2017*, *2017-Octob*, 86–91. https://doi.org/10.1109/I2CACIS.2017.8239038

Lin, Y., Yang, X., Zou, N., & Franz, M. (2015). Transit signal priority control at signalized intersections: A comprehensive review. *Transportation Letters*, *7*(3), 168–180. https://doi.org/10.1179/1942787514Y.0000000044

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flotterod, Y. P., Hilbrich, R., Lucken, L., Rummel, J., Wagner, P., & Wiebner, E. (2018). Microscopic Traffic Simulation using SUMO. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *2018-November*, 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938

Lukkari, J. (2019, March 25). *Suomen autot vanhenevat käsiin – keski-ikä jo yli 12 vuotta*. Tekniikka&Talous. https://www.tekniikkatalous.fi/uutiset/suomen-autot-vanhenevat-kasiin-keski-ika-jo-yli-12-vuotta/23026fa8-b43a-31e8-ba08-3c6b6245ac54

Mäkelä, J. (2020). *Personal interview*.

Miguel, R., Paulo, B., Lin, J., Rouphail, N. M., & Sacks, J. (2004). *Calibrating and Validating Deterministic Traffic Models: Application to the HCM Control Delay at Signalized Intersections Acknowledgment and Disclaimer*. www.niss.org

Monzón, A., Hernandez, S., Martínez, A. G., Kaparias, I., & Viti, F. (2016). Public transport in the era of ITS: Its technologies for public transport. In *Modelling Public Transport Passenger Flows in the Era of Intelligent Transport Systems* (Issue 9783319250809, pp. 85–128). Springer International Publishing. https://doi.org/10.1007/978-3-319-25082-3_3

Moore, F. (2016). *Qualitative vs Quantitative Research Completing the PhD-level Research Design course at NCU View project Completing the IT Data Communications Management course at NCU View project*. *November*, 3. https://doi.org/10.13140/RG.2.2.34861.49128

Naznin, F., Currie, G., Sarvi, M., & Logan, D. (2016). An empirical bayes safety evaluation of tram/streetcar signal and lane priority measures in Melbourne. *Traffic Injury Prevention*, *17*(1), 91–97. https://doi.org/10.1080/15389588.2015.1035369

OpenStreetMap. (n.d.). *OpenStreetMap*. Retrieved September 2, 2020, from https://www.openstreetmap.org/about

PTV GROUP. (n.d.). *Our Portfolio | Traffic & Logistics Software | PTV Group*. Retrieved September 2, 2020, from https://www.ptvgroup.com/en/solutions/products/

Salonen, M. (2010). *Joukkoliikenteen valoetuuksien toteuttaminen SYVARI-ohjauksella*.

Sim, J.-G., & Lee, S.-S. (2009). Estimation of Acceleration Rates of Bus and Passenger car at Signalized Intersections. *Journal of Korean Society of Transportation*, *27*(6), 69–77.

Simulation of Urban MObility. (n.d.). *SUMO*. https://sumo.dlr.de/docs/index.html

Singh, S. K., Komolkiti, P., & Aswakul, C. (2017). Impact Analysis of Start-Up Lost Time at Major Intersections on Sathorn Road Using a Synchro Optimization and a Microscopic SUMO Traffic Simulation. *IEEE Access*, *6*, 6327–6340. https://doi.org/10.1109/ACCESS.2017.2739240

Song, Y., & Noyce, D. (2019). Effects of transit signal priority on traffic safety: Interrupted time series analysis of Portland, Oregon, implementations. *Accident Analysis and Prevention*, *123*, 291–302. https://doi.org/10.1016/j.aap.2018.12.001

Sortter. (2019). *Suomen suosituimmat autot 2019 ja autoilun kasvavat trendit 2020-luvulla*. Sortter. https://sortter.fi/blogi/artikkeli/suomen-suosituimmat-autot-2019/

Stevanovic, A., & Martin, P. T. (2006). *Assessing the Ageing of Pretimed Traffic Signal Control Using Synchro and SimTraffic Objectives and Performance Based Management of Traffic Signal Operations View project Traffic Bottlenecks Identification, Diagnosis, and Innovative Solutions to Local/Sys*. https://www.researchgate.net/publication/274137225

Symonds, P. J. (2014). Traffic signal priority for buses. *IET Conference Publications*, *2014*(CP633). https://doi.org/10.1049/cp.2014.0796

Toronto Transit Commission. (2004). *Transit Signal Priority, The Toronto Experience*.

Turku. (2020). *Raitiotien suunnitelmat ja selvitykset*. http://www.turku.fi/asuminen-ja-ymparisto/projektit-ja-hankkeet/raitiotie/raitiotien-suunnitelmat-ja-selvitykset

Udomsilp, K., Arayakarnkul, T., Watarakitpaisarn, S., Komolkiti, P., Rudjanakanoknad, J., & Aswakul, C. (2017). Traffic data analysis on Sathorn road with synchro optimization and traffic simulation. *Engineering Journal*, *21*(6 Special Issue), 57–67. https://doi.org/10.4186/ej.2017.21.6.57

von Flavio, P. (2017). *Public Transit Mapping on Multi-Modal Networks in MATSim*. www.morf-ag.ch

Węglarczyk, S. (2018). Kernel density estimation and its application. *ITM Web of Conferences*, *23*, 00037. https://doi.org/10.1051/itmconf/20182300037

Yang, K., Menendez, M., & Guler, S. I. (2019). Implementing transit signal priority in a connected vehicle environment with and without bus stops. *Transportmetrica B: Transport Dynamics*, *7*(1), 423–445. https://doi.org/10.1080/21680566.2018.1434019

Yao, D., Su, Y., Zhang, Y., Li, L., Cheng, S., & Wei, Z. (2009). Control strategies for transit priority based on queue modeling and surrogate testing. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, *13*(3), 142–148. https://doi.org/10.1080/15472450903084287

Appendices

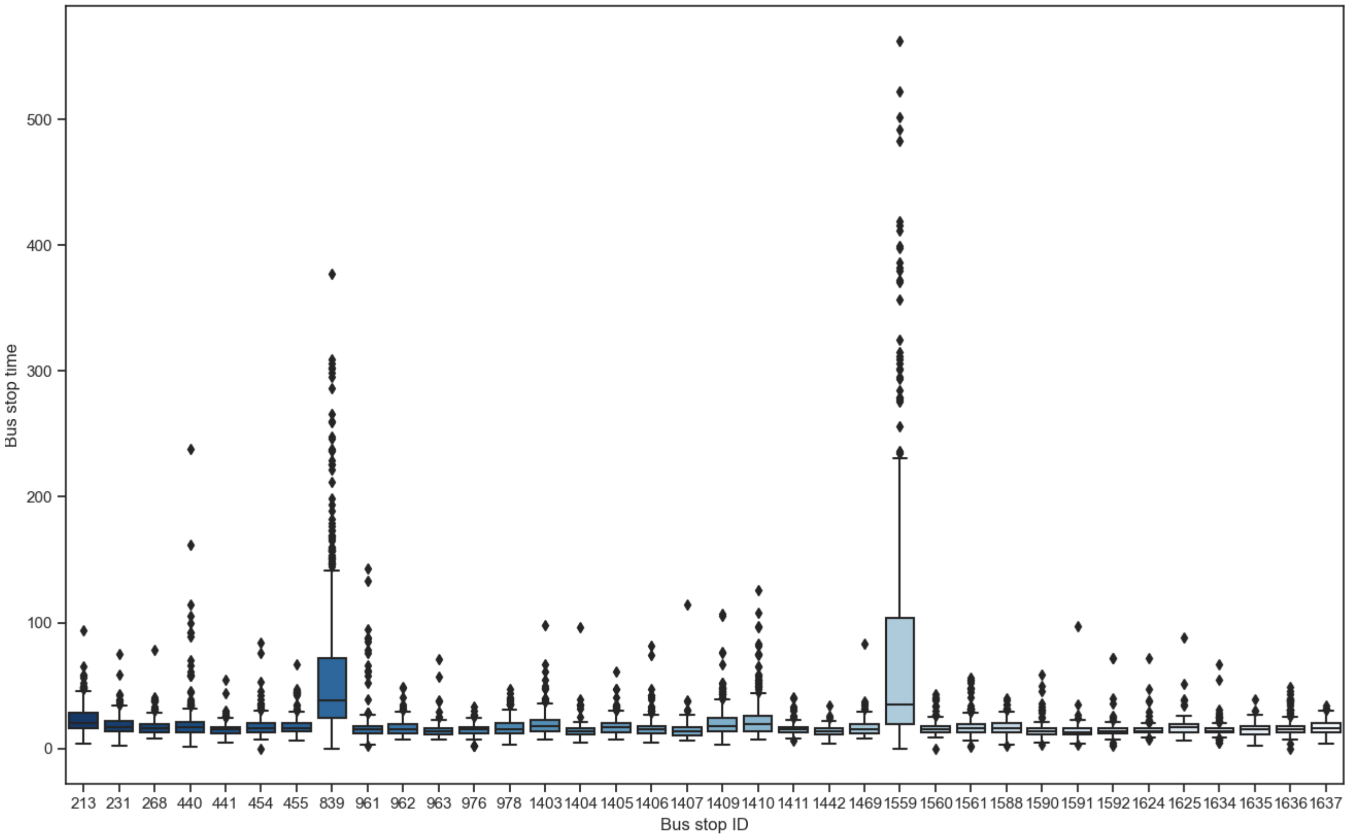

Figure 30: Bus stop times during rush hours (outliers included)

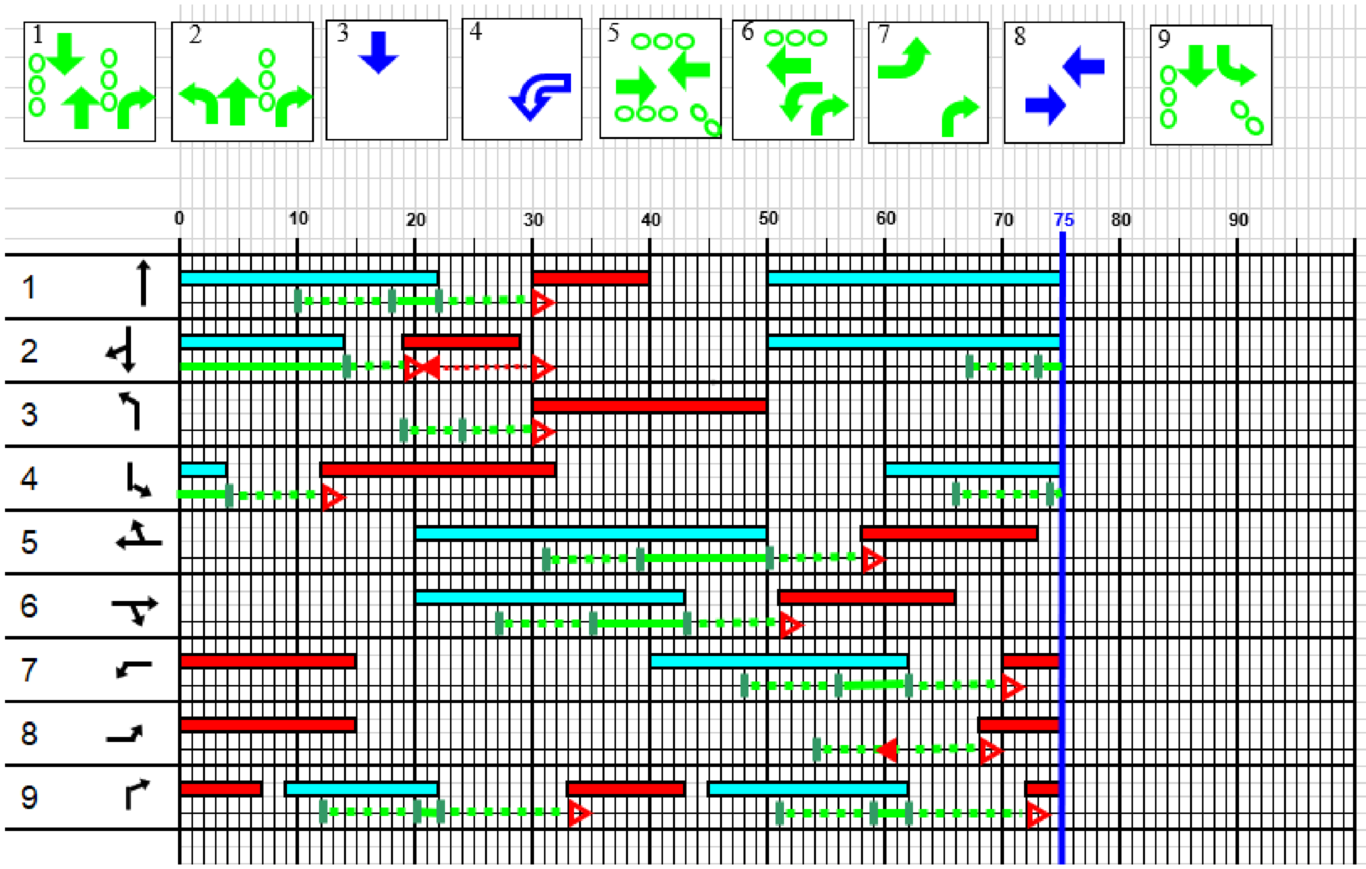

Figure 31: Traffic light configuration file for intersection 409 (part 1)

Ohjaussuunta	Perusvihreän vaiheet	Ylimääräisen vihreän vaihe	Takuumaksimiaika	Synkronointimaksimiaika	Etuusmaksimiaika	Synkronointipidennys		Synkronointinollaus		Synkronointiviivytys		Passiivivihreän esto	Pyyntötapa	Suunnat, joille joukkoliikennepyyntö antaa aiennusnollauksen	Takuuvihreä		Pitoaika	Aloituslupa-ajan rajoitus	Lepotila	configurations
						Alkaa	Päättyy	Alkaa	Päättyy	Alkaa	Päättyy				Alkava suojatie	Kesto				
1	1,2		10	40	+25	50	22	30	40				K				5		V	25
2	1,9	3	12	40	+25	50	14	19	29			22 - 30	K	3,5,6,7,8					V	26
3	2		6	10		-	-	30	50				O						P	8
4	4		8	30		60	4	12	32				O						P	19
5	5,6	8	10	40	+25	20	50	58	73				O + 15	1,2,3,4,8					P	25
6	5	8	10	40	+25	20	43	51	66				O + 17	1,2,3,4,7					P	25
7	6	4R	10	20	+25	40	62	70	15				O	1,2,3,4,6					P	15
8	7		6	10		-	-	68	15			61 - 68	O						P	8
9	1,6 7		10	20	+25	9 45	22 62	72 33	7 43				O						E	15

Figure 32: Traffic light configuration file for intersection 409 (part 2)

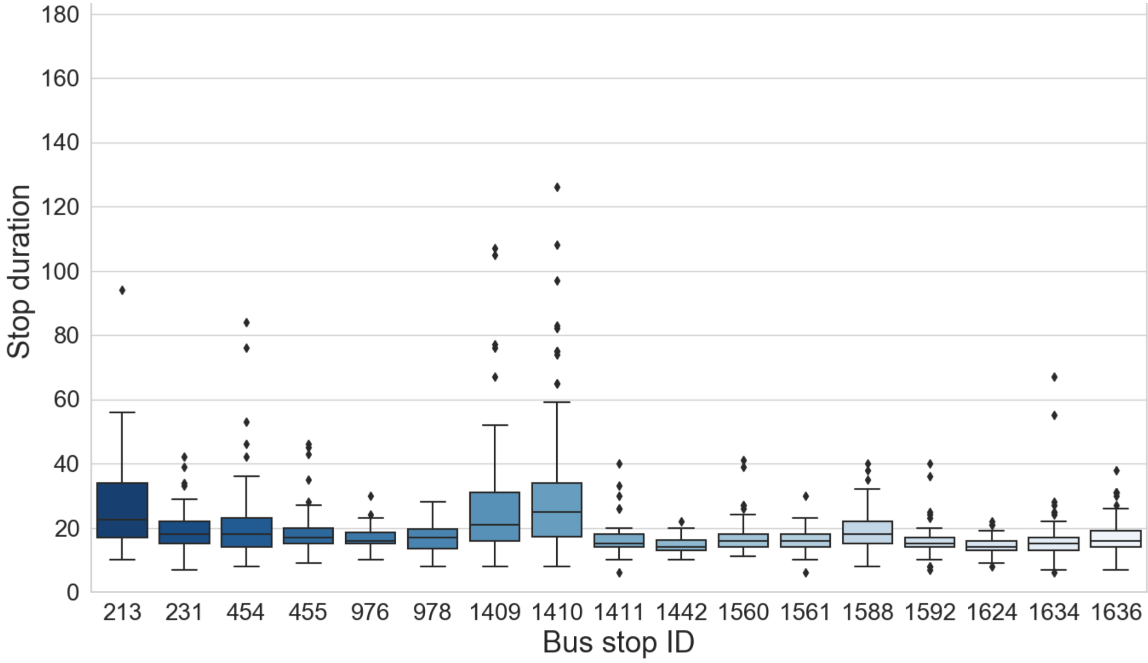

Figure 33: Bus stop duration during rush hours (east, evening)

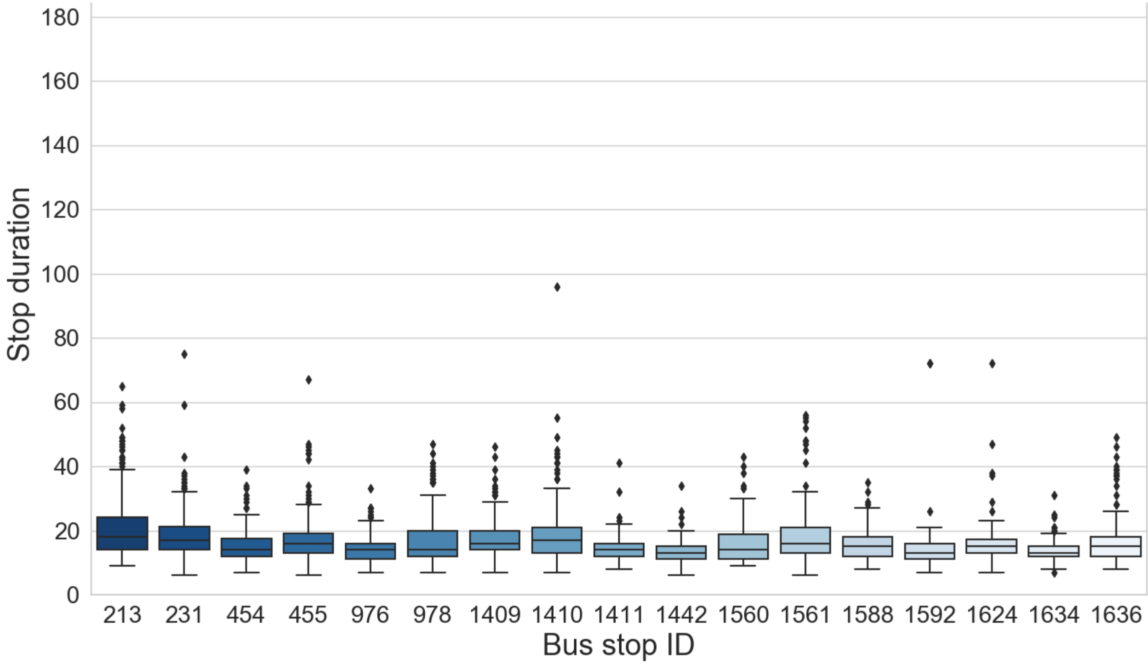

Figure 34: Bus stop duration during rush hours (east, morning)

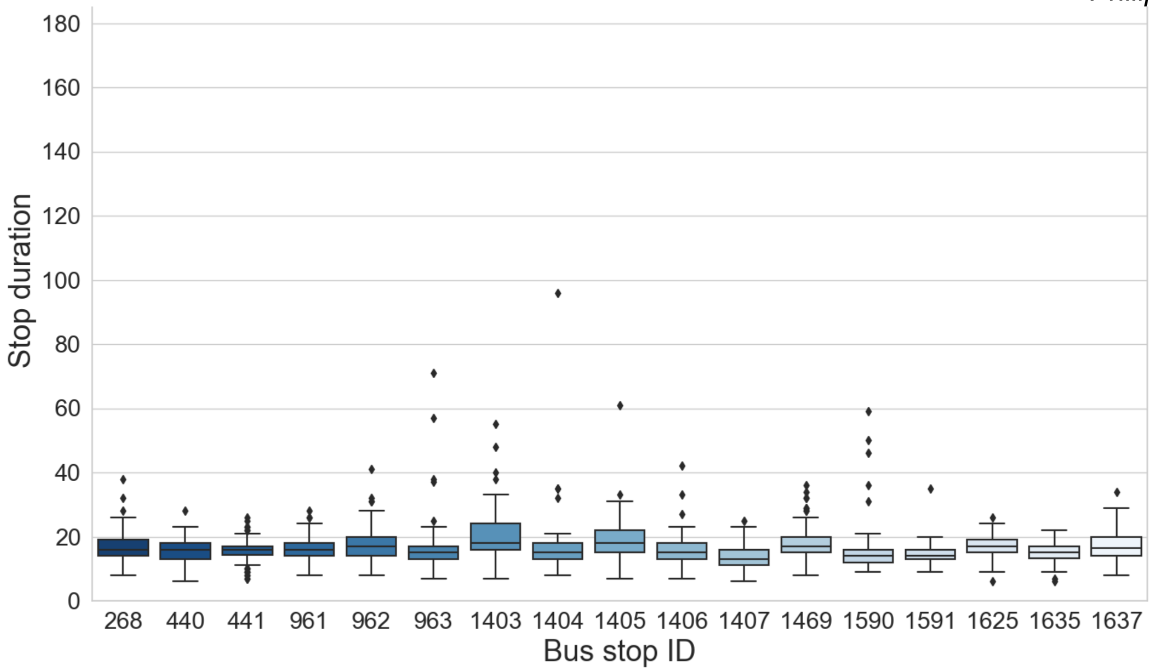

Figure 35: Bus stop duration during rush hours (west, morning)

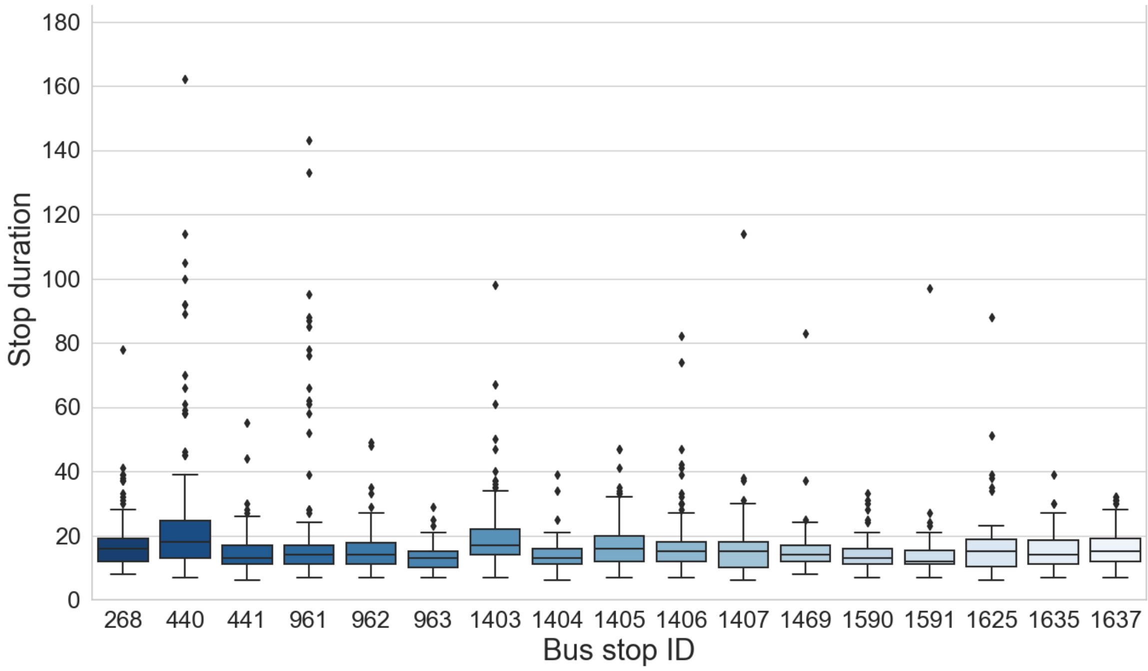

Figure 36: Bus stop duration during rush hours (west, evening)

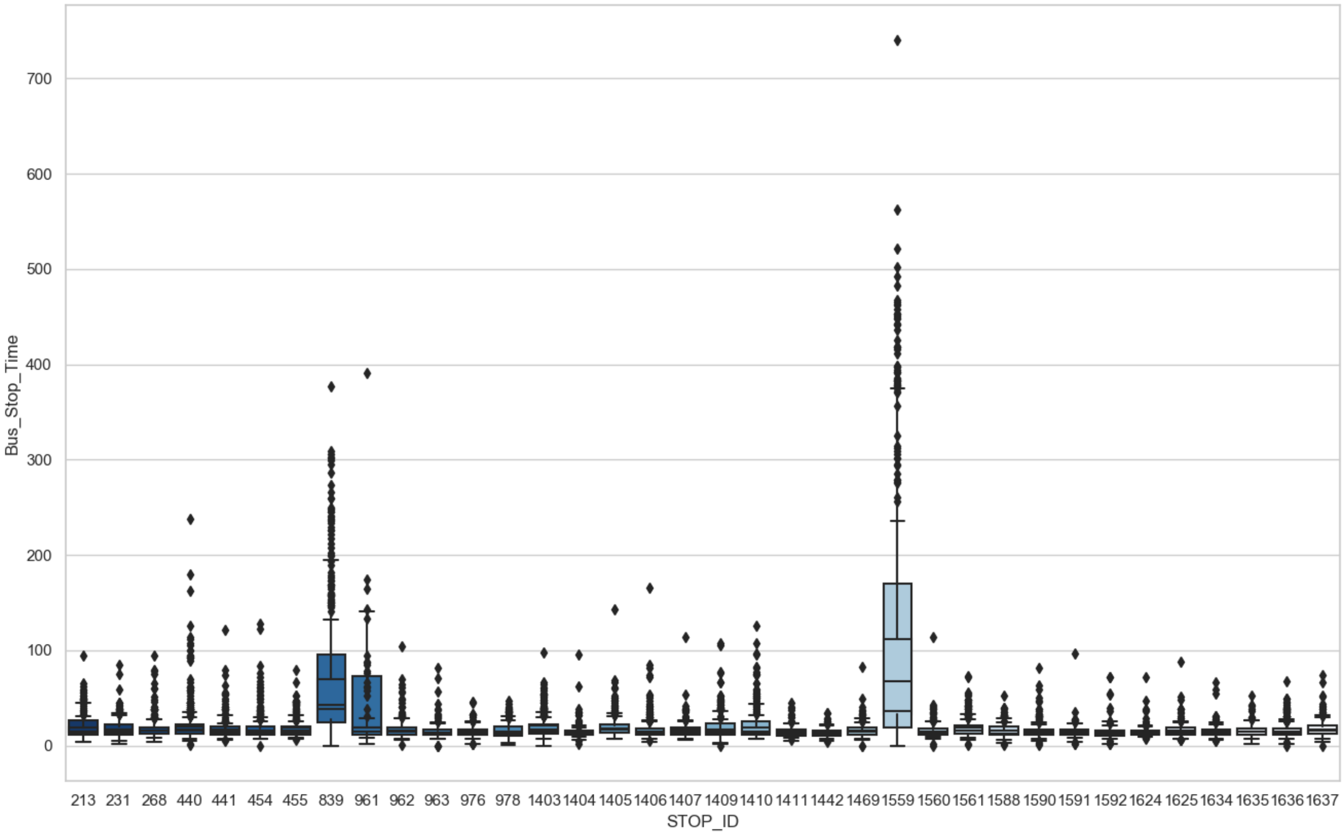

Figure 37: Bus stop duration rush hours (All values included)

Table 19: Bus stop durations

Bus stop	Bus stop time (mean)	Bus stop time (median)
213	22,90	20
231	18,66	17,5
268	16,77	16
440	21,54	17
441	15,27	15
454	17,36	16
455	17,71	16
839	62,82	39
961	19,56	15
962	16,07	15
963	14,96	14
976	15,71	15
978	17,07	15
1403	19,60	18
1404	15,29	14
1405	17,62	17
1406	16,23	15
1407	15,64	14
1409	20,72	18
1410	22,80	19
1411	15,61	15
1442	14,03	14
1469	16,22	15
1559	80,53	36
1560	15,99	15
1561	17,92	16
1588	17,14	16
1590	14,67	14
1591	14,73	13
1592	14,94	14
1624	15,32	14
1625	17,25	17
1634	15,16	14
1635	15,08	15
1636	16,52	15
1637	16,40	16

Table 20: Results with original values

Detector distance	No Priority Baseline		60 m		100m		150m	
Direction	East	West	East	West	East	West	East	West
Count	400	400	400	400	400	400	400	400
Duration (min)								
mean	15.35	15.71	14.28	13.99	14.13	13.65	13.89	13.24
std	0.94	1.16	0.49	0.66	0.58	0.53	0.60	0.49
min	13.20	12.68	12.67	12.08	12.57	11.97	12.52	11.83
25 %	14.70	15.03	14.02	13.53	13.83	13.28	13.55	12.95
50 %	15.28	15.73	14.28	13.95	14.12	13.58	13.85	13.16
75 %	15.95	16.40	14.53	14.42	14.37	13.95	14.10	13.53
max	18.18	19.62	16.48	16.90	16.12	16.07	16.38	15.20
No waiting in traffic lights (min)								
mean	13.47	13.16	13.46	12.91	13.43	12.82	13.34	12.67
std	0.35	0.40	0.33	0.37	0.37	0.34	0.38	0.34
min	12.62	12.18	12.53	11.83	12.32	11.78	12.42	11.77
25 %	13.23	12.90	13.23	12.67	13.17	12.60	13.10	12.43
50 %	13.45	13.15	13.45	12.90	13.42	12.79	13.33	12.63
75 %	13.68	13.40	13.67	13.17	13.64	13.05	13.53	12.88
max	14.73	14.75	14.52	14.25	14.52	13.87	14.83	13.88
Stop time (s)								
mean	140.96	116.02	143.25	108.19	144.27	106.04	144.78	103.20
No bus stop (min)								
mean	13.00	13.78	11.91	12.19	11.74	11.88	11.49	11.52
std	0.85	1.01	0.41	0.54	0.45	0.43	0.47	0.39
min	10.98	10.92	10.62	10.65	10.47	10.58	10.35	10.60
25 %	12.41	13.18	11.68	11.83	11.50	11.60	11.20	11.28
50 %	12.97	13.79	11.93	12.15	11.73	11.82	11.47	11.48
75 %	13.55	14.37	12.13	12.53	11.95	12.18	11.73	11.73
max	15.42	17.25	13.72	14.70	13.18	14.07	13.42	13.15
No Time loss (min)								
mean	10.18	9.78	10.22	9.65	10.24	9.61	10.24	9.56
std	0.19	0.25	0.19	0.20	0.23	0.19	0.25	0.19
min	9.73	9.24	9.81	9.08	9.76	9.08	9.74	9.08
25 %	10.05	9.61	10.08	9.50	10.07	9.48	10.07	9.44
50 %	10.16	9.76	10.21	9.62	10.20	9.60	10.21	9.54
75 %	10.30	9.91	10.34	9.79	10.38	9.74	10.37	9.68
max	10.76	10.64	10.82	10.23	11.00	10.09	11.30	10.15

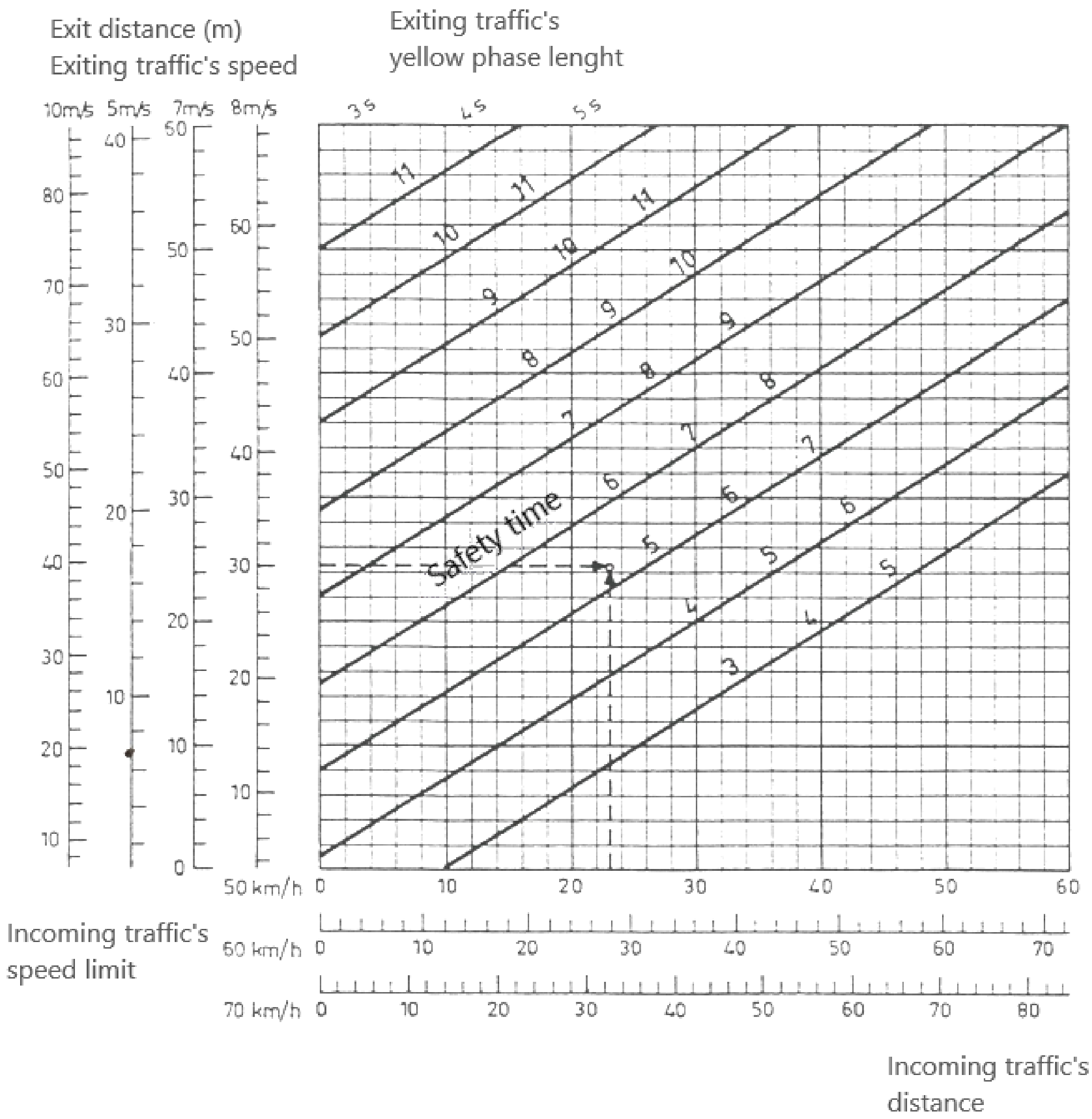

Figure 38: Safety time

Table 21: Disturbance on crossing traffic at detector length 150m (75% of traffic)

	No Priority		-	Extra green		-	No extra green	
	405	409		405	409		405	409
Duration								
count	129695	54390		-	54390		129695	54390
mean	45.48	49.76		-	50.99		46.40	50.30
std	23.78	23.18		-	23.99		23.54	23.08
min	9	15		-	15		9	15
25 %	24	27		-	28		25	28
50 %	45	49		-	50		46	49
75 %	66	71		-	71		67	70
max	94	100		-	126		116	117
Depart delay								
count	129695	54390		-	54390		129695	54390
mean	-0.41	-0.30		-	-0.29		-0.37	-0.30
std	0.54	0.36		-	0.37		0.86	0.37
min	-1	-1		-	-1		-1	-1
25 %	-0.71	-0.6		-	-0.6		-0.71	-0.6
50 %	-0.47	-0.2		-	-0.2		-0.47	-0.2
75 %	-0.19	0		-	0		-0.18	0
max	15.09	2.99		-	2.95		33.85	2.95