Analysing proposed Low Traffic Neighbourhood-like schemes using agent-based traffic simulation

Samuel Molyneux

BSc in Computer Science
The University of Bath
2022

Analysing proposed Low Traffic Neighbourhood-like schemes using agent-based traffic simulation

Submitted by: Samuel Molyneux

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Computer Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

Low Traffic Neighbourhoods are an innovative traffic policy tool seeing rising popularity in cities across the United Kingdom and Europe due to the potential benefits of safer roads, enhanced urban environment, higher active transport usage, and reduced emissions. Currently, the only way to analyse potential schemes is through costly physical trials. This project proposes a novel application of traffic simulation technologies for predictively modelling and assessing the effects of potential Low Traffic Neighbourhood schemes.

To facilitate the application of this method, several tools were developed to automate and simplify the process. The method was implemented to analyse and experiment with modelled outcomes to demonstrate how potential schemes could be evaluated in a real-world policy proposal, drawing actionable inferences and suggestions. The benefits of this technique are the vastly reduced time and cost needed to assess potential schemes, as well as the production of a non-disruptive environment allowing for experimentation and gathering of predictive data. This predictive data can be visualised and reinterpreted to supplement discussions about potential schemes and facilitate communication with the general public.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Analysing proposed Low Traffic Neighbourhood-like schemes using agent-based traffic simulation

1 INTRODUCTION

In London a policy tool termed 'Low Traffic Neighbourhoods' (LTNs) has been gaining momentum since 2014 (Jubb and Mulis). This policy, involving the restriction of through traffic to decrease traffic volume and to increase the use of active transport modes, has seen significant backing by Transport for London as part of its aims to meet climate targets and reduce emissions. In Spring of 2020, many councils took the opportunity provided by the reduced road usage during the COVID-19 pandemic to implement schemes using government funding. In London alone £20 million (Department of Transport, 2020a) was provided for long-term projects as part of the 'Active travel fund', this led to ninety-seven new LTN schemes emerging between 2020 and 2022 with many councils seeking to reap the benefits of safer roads and higher active transport usage (Donovan, 2022).

These LTNs are an evolution of the city planning concepts used in historical cities across Europe. They are a smaller scale equivalent of traffic-cell based policies that are used in many European cities, most notably Ghent where in 2017 such a plan was implemented over night (City of Ghent). As part of their aim to reach carbon neutrality by 2030, Bath and Northeast Somerset (B&NES) Council carried out a consultation into a similar plan for Bath city centre in 2021 and 2022 (Sumner, 2022).

Often the only way for planners to assess such schemes in advance of implementation is to carry out lengthy trials which can have high cost and can be difficult to gain support for (Ames, 2020). Additionally, failed trials often cause negative discourse within the community which can make implementing more effective measures in the future challenging.

This project investigates the use of traffic simulation to offer an alternative method to physical trials, allowing planners to easily and quickly experiment with and assess proposed schemes. By providing a projection of policy effects, this would decrease the number of failed trials and allow for free experimentation without the requirement to make changes to road infrastructure.

Even in places where trials were deemed successful or in areas where LTNs were implemented without public consultation, such as Ealing, there has been significant protest and resistance (Donovan, 2022). The proposed method helps avoid this by letting planners see detailed data about a scheme's effects in advance thus avoiding failed schemes. It also facilitates the dialogue with residents by providing clear visualisations of what the effects will be rather than nebulous projections about "attractive and safe" environments that councils often resort to.

In this project, following on from an investigation of the existing academic literature regarding LTNs and traffic analysis techniques, the problem is formally defined and the method is outlined in general terms. To develop the core concepts of the method it was first used to model and analyse a theoretical LTN scheme in an artificial grid-based road network. This simpler investigation allowed for a clear demonstration without the complexity of context-based details. It also increases confidence in conclusions drawn from emergent behaviour in more advanced applications by providing reliable results in a controlled environment.

As a demonstration of such an application and to further develop the intricacies of the method, an investigation was undertaken into the proposed traffic-cell based policy in the city of Bath. This involved generating an accurate representation of the existing traffic behaviour and road layout as well as constructing a road network that implemented the proposed scheme. A series of tools were developed to assist the conversion of traffic demand to the new network and generate a predictive simulation. Analysis was carried out for both of the studied simulations, showing how planners could use the method's results to assess, experiment with, and extract data about proposed traffic changes including the production of easy-to-understand visualisations. The results were used to form and justify a clear verdict on the proposed Bath city centre policy and suggestions for alterations and implementation considerations were given. The account of applying the method to the two scenarios and the data generated by them were then used to evaluate the method and discern its value for further academic usage. Suggestions for future development of and research utilising the method were also provided as well as final reflections on the success of this project.

2 LITERATURE AND TECHNOLOGY SURVEY

2.1 Low Traffic Neighbourhoods

This project aims to provide an objective method for analysing the effects of implementing LTNs and related traffic control measures on the surrounding traffic system. This would allow urban planners to better assess candidate areas for conversion to LTNs and to provide residents of prospective areas a better understanding of how measures may affect them. Considering literature regarding LTNs, related policies, and their predecessors provides a foundational understanding of the problem.

2.1.1 What are LTNs and why are they needed?

LTNs are a tool used increasingly by policy makers in cities like London (Aldred and Goodman, 2020) and Ghent . Their purpose is to reduce some of the negative impacts caused by excessive motor vehicle traffic through a city. The concept involves the removal of through motor vehicle traffic in an area such that only residents and limited delivery/service vehicles remain. The goal of these areas is an increased pedestrian usage meaning more community interaction, less pollution, and more foot traffic to local businesses and other societal improvements. More detailed investigation of methods and results can be found later in this section.

Before the invention and popularity of motor vehicles, pedestrianised residential and commercial areas were the norm. The increase in number of cars in Europe during the 1960s and 1970s lead to many problems with the existing city centre infrastructure: narrow roads lead to heavy congestion, noise and air pollution, and increased risk for pedestrians. The prevailing method of alleviating these problems was the 'Pedestrian Mall', a concept that involves the complete exclusion of motor vehicles in a commercial area. The Pedestrian Mall was primarily pioneered in the Unites States (US)(Judge, 2015) and Europe. By 1961 many cities had implemented these ideas and were seeing the benefits.

However, there were places where the Pedestrian Mall was unsuccessful, having been partially blamed for the mass movement to suburban areas in the US during the 1960s. They are criticised for placing too much focus on pedestrians and ignoring motor vehicles entirely.

These criticisms are focused in the US and, rather than there being an inherent issue with Pedestrian Malls, it is worth considering specific local circumstances. These might include poor planning by local authorities and issues with very high vehicle ownership, especially as Pedestrian Malls have seen success in European cities such as Amsterdam, Stockholm, and Gothenburg as far back as the 1960s (Robertson, 1991). There is also recent evidence for the benefits of excluding motor traffic, with cities like Seville (Castillo-Manzano, Lopez-Valpuesta and Asencio-Flores, 2014) expanding their pedestrianised areas.

LTNs are an evolution of this successful pedestrianisation, with increased consideration to motor vehicles to avoid failings like those seen in the US. By limiting traffic to residential access, rather than excluding vehicles entirely, many of the benefits of pedestrianisation can be realised without such a sacrifice of convenience for those living and using the area. This traffic limiting is done by the implementation of 'modal filters' that disallow or discourage certain forms of motor vehicles whilst allowing the passage of pedestrians, bikes, and local traffic. For example, a bus gate is a modal filter that is generally created using rising bollards, allowing public transport and cyclists through whilst stopping other motor vehicles.

Unlike the majority of previous pedestrianisation efforts which have been implemented in commercial districts, LTNs focus on residential areas with smaller adjacent commercial areas. By reducing traffic volume and speed, planners aim to increase community interaction and local business usage whilst decreasing pollution and road traffic injuries. Active transport methods like walking and cycling are provided with more direct routes to amenities, compared to the more circuitous path that cars must take to bypass modal filters. This means it is often more convenient for a resident to utilise these methods and active transport rates go up, further reducing the total traffic in an area and often leading to lower car ownership in general. Additionally, increased footfall through town centres leads to a boost in local business.

Modal filters often take the form of planter boxes, parklets, or width restrictions, the idea being that as well as fulfilling their role in traffic control, they improve the perceived 'quality' of a neighbourhood and act as clever road design to slow cars. This creates a neighbourhood that is safer for children to play in the streets and one that is more attractive to traverse by foot or bike.

Generally, groups of LTNs are placed in proximity and are connected by arterial roads that allow for service vehicles and through traffic to pass (Figure 2.1). These individual 'cells' are clustered around commercial and public transport hubs, ensuring that residents of all areas have reasonable walking time to important amenities. In order to avoid the high-capacity main roads forming a barrier between cells, care is taken to provide sufficient crossings for both pedestrians and cyclists. This helps maintain the shorter travel times for active methods compared to personal motor vehicle trip time (Jubb and Mulis).

Figure 2.1 – Use of arterial routes in LTN scheme (Jubb and Mulis)

LTN's have seen particular recent uptake as part of London's 'Mini-Holland' scheme which began in 2014 (Holland, 2014). This scheme, funded by the Mayor of London and inspired by Dutch cycling and pedestrian infrastructure sought to make various improvements to London boroughs through the implementation of LTNs. These are the most representative examples of the LTNs that are the focus of this paper, thus the results are particularly relevant for informing the investigation. The Outer London boroughs of Kingston, Enfield and Waltham Forest were selected due to their low levels of cycling and walking compared to Inner London areas (Aldred, Croft and Goodman, 2019) and extensive research has been done on the results of this scheme, leading to expanded funding as recently as 2020.

The various studies and research have found the benefits to include:

- A reduction in car ownership by 8% (Goodman, Urban and Aldred, 2020)
- A greatly reduced risk from road traffic accidents for walking and cycling journeys (Laverty, Aldred and Goodman, 2021)
- Planners claim a 15% reduction in total traffic (Jubb and Mulis)

Whilst the terms 'traffic-cell plans', used for large area schemes like the City of Ghent, and LTNs, which often designates smaller scale plans, are distinct in that they often reflect the scope of such a scheme, the terms will be used interchangeably throughout this project. This choice was made as the relevant underlying method and goals are identical: to reduce traffic and insight active transport usage by reducing through traffic and diverting other journeys.

2.2 Traffic Analysis

To predict the impact of LTN schemes on surrounding traffic ecosystems, a traffic analysis method will be selected. From the historical literature in the area, context will be provided for review into more modern techniques. A technique will be selected and its previous usage will be assessed and explored to ensure it is suitable and inform the development of the predictive system.

2.2.1 Analytical Methods

Traffic modelling and analysis has been a consideration for almost as long as cars and traffic have been a feature of our society. As early as the 1950s (Pipes, 1953) traffic flow was being considered from a theoretical stand point. This work focussed particularly on the fundamental concept of following distance which although simplistic when compared to more recent works on the topic like that carried out by Hoogendoorn et al (2011), did lay the foundation for many future works.

By 1972 (Leutzbach, 1972), text books were being written that provided a comprehensive mathematical break down of traffic network design and control. This method sought to combine various techniques in order to provide a better understanding to the engineers dealing with such issues. However, this approach is limited in that it focuses entirely on the links between nodes in traffic networks. The exclusion of the nodes themselves is significant as these are often where the primary bottlenecks in traffic flow occur.

Extensive mathematical work has been carried out since that time, building and expanding on the core concepts. Many different frameworks for modelling traffic behaviour have been investigated, including cellular automata (Kerner, Boris S., Klenov and Wolf, 2002), Physics-based (Kerner, Boris S, 1999), and elastic demand modelling (Ben-Akiva, De Palma and Kanaroglou, 1986).

The above-mentioned techniques can all be defined as ‘analytical’ modelling, in that they use theoretical concepts based on maths, physics and geometry to find elegant solutions. However, they do tend to rely on assumptions and abstractions that can be seen to oversimplify complex systems. For this reason, it is often difficult to extract concrete predictive results when applying them to traffic networks.

The most relevant research into these techniques, (Fournier, 2021), explored a very similar policy to that of LTNs – implementation of Hybrid pedestrian and transit priority zoning. It is thus a good example to demonstrate the application of analytical methods for traffic modelling. The research aimed to determine optimal sizing of traffic-controlled areas for reduced travel time. The method works by parameterising the characteristics of an urban area such as dimensions and spacing of streets and pairing this with information about pedestrian areas within the city. Potential traffic demand is calculated, then through a series of mathematical equations the relations between various factors like zone sizing, traffic flow, and travel time can be derived. The paper concludes by presenting a graph demonstrating the calculated optimal pedestrian and transit zone sizing (Figure 2.2). This shows the usefulness of analytical methods in terms of their width of applicability and ability to provide backing for qualitative conclusions such as “Slightly over sized (pedestrian or transit priority) zones would not result in any appreciable loss in performance”. As noted by the author, the technique presented is limited in its evaluation of factors such as displaced traffic demand (a key factor in this paper’s investigation) and significantly simplifies the behaviours of motor traffic.

Overall Fournier's research (2021) is relevant to the project in its consideration of diverting/restricting traffic in lieu of pedestrian prioritisation, it also seeks to extract values that this research will consider such as traffic density and capacity. It does differ significantly from the aims of this project in precision of outcome, largely due to the inherent restrictions of analytical methods. Due to these differences, such analytical methods are deemed more useful in the research phase of traffic policy rather than the investigation of proposed changes that this project is based on.

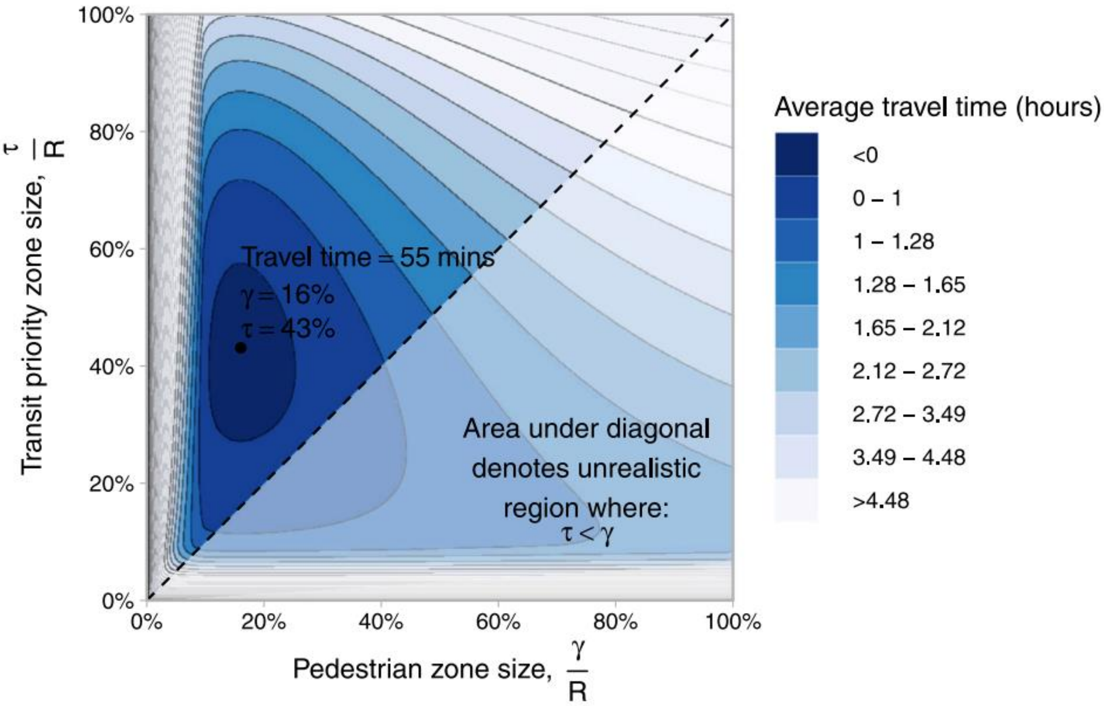


Figure 2.2 – Optimal zoning policy produced from analytical methods from Nicholas Fournier, Institute of Transportation Studies, University of California, Berkeley, 2021

2.2.2 Simulation Methods

Another approach to investigating traffic behaviour is simulation, which aims to provide more discrete answers to traffic problems, generally by modelling the behaviour of individual vehicles. This technique was first investigated in Germany by Larsson (Larsson, 1972) who created a rudimentary traffic simulation using sets of matrices. Even at this point, whilst computing power was far less available than it is today, consideration was given to making the problem amenable to solution by computation.

As reviewed by Chen and Cheng (2010), there have been many more recent applications of simulation techniques in traffic modelling, particularly in the field of agent based simulation. In their study Chen and Cheng categorised agent-based traffic simulation into five groups:

- agent-based traffic control and management system architecture and platforms
- agent-based systems for roadway transportation
- agent-based systems for air-traffic control and management
- agent-based systems for railway transportation
- multi-agent traffic modelling and simulation

The category most relevant to this project is that of multi-agent traffic simulations. In this form of simulation, each entity (vehicle, signal light, variable sign etc.) is modelled as an agent, interactions are then simulated over a period of time and deductions can be made by altering the setup parameters of the system. Using this form of simulation will allow for the necessary granularity of control, it also reduces the number of necessary assumptions and abstractions thus keeping the result more reflective of reality.

Such simulation techniques tend to be implemented through an established framework, examples include PTV Vissim (PTV Group), MATSim (MATSim Community, 2022), and Simulation of Urban Mobility (SUMO). These frameworks provide a researcher with the building blocks for modelling a traffic system and are amenable to configuration and alteration to suit a specific use. These tools provide fundamental behavioural models like choice(Flötteröd and Kickhöfer, 2016) and car-following models. By supplying these natively, a researcher is able to focus on novel ideas, building on the established work of experts in fields that have been under development for many years(Gipps, 1981). It also allows for new developments and applications upon new technologies such as inter-vehicle communication(Tang et al., 2014) to be quickly supplied to wider users.

As a result of these frameworks, much of the work that is done in the application of traffic simulation models is the parameterisation of the key features in an environment:

- mapping of road networks,
- demand modelling, and
- defining behavioural elements like traffic rules.

Frameworks usually provide a suite of tools to assist in this stage, which can be quite varied due to the different sources of data available and forms of results sought. In the final stage of implementing a traffic model, the researcher must extract the desired information from a simulation that has been parametrised to test their research question. Frameworks provide a vast amount of data to be analysed, along with a further set of tools to filter and visualise results. It is often necessary for more complex data to be analysed through bespoke scripts written by a researcher. Most frameworks include support for such methods to be easily integrated into the information pipeline.

For the purpose of this project, simulation techniques were assessed to be the most applicable due to the precision of results and the availability of relevant data sources for the area of concern. Specifically, the simulation framework SUMO was selected, due to its history of use for academic research, its suitability for alterations, and its availability as an open-source tool.

2.2.3 Simulation of Urban MObility

SUMO has been defined by its authors as “a microscopic, inter- and multi-modal, space-continuous and time-discrete traffic flow simulation platform” (Krajzewicz, 2010; Krajzewicz et al., 2012). The microscopic nature is important for this investigation as it will involve subtle changes to street architecture that require individual agent consideration to see representative results. Being space-continuous and time discrete is also essential as to gain a full picture of resultant traffic impact, the varied demand throughout the day is important to model.

Rather than solely being a traffic simulation mechanism, SUMO is a suit of features that allow for the manipulation of input and output data before, during, and after processing (German Aerospace Center (DLR), 2022). These include:

- Road network generators – these create the network used for simulation
- Routing tools – various ways to establish a vehicles path through the network
- *sumo-gui* – provides a graphical user interface, allowing a simulation to be monitored during execution
- Traffic Control Interface (TraCI) – an API that allows external applications to interact with SUMO simulations

To create a road network in SUMO that reflects real-world streets, the service Open Street Maps (OSM) is often used (Krajzewicz et al., 2012) in combination with SUMO's native "net-convert" road network generator. OSM is an Open-Source project to create a freely available global road map using the same crowd sourcing structure that built Wikipedia (Haklay and Weber, 2008). Much of its benefit is in the transparency of its data and the ease by which it can be accessed and manipulated for use in various forms. It has seen extensive use in academic works and is well maintained and updated.

SUMO is a tool with a significant history of academic usage. In a paper aimed to assess the suitability for accurate traffic simulation, Dias et al.(2013) found it to be capable of "simulating urban traffic" by using the Portuguese city of Coimbra as an example. They found that it identified correctly the areas of congestion that exist in the actual traffic network when set up using recorded traffic data. This research did also identify that the tools provided in SUMO are not entirely sufficient to simply input traffic data and output an accurate simulation, some manipulation was required of the generated network and signalling systems. SUMO has also been used in academia to investigate various traffic systems including traffic signal analysis (Krajzewicz et al., 2005) and Traffic Demand Evaluation(Codeca et al., 2017).

3 APPROACH DESCRIPTION

In this chapter, the reasoning and methodology of the investigation are described. The general problem is defined and the key steps of the proposed method are outlined. An explanation of the case studies chosen as example applications is provided to give context to the later accounts of method development and application.

3.1 The Problem

The overarching goal is to project results of implementing LTN schemes by modelling the existing traffic system in SUMO and subsequently implementing the proposed changes in the simulated environment, ensuring traffic demand is kept constant. The result will be an accurate representation of traffic behaviour within the new road system that can be used in a variety of ways to analyse, assess, and experiment with candidate schemes.

This is beneficial as, when implementing LTN schemes to reduce motor traffic through key urban areas, policy makers must disallow through traffic. However, this inherently results in some amount of displaced traffic as deliveries and longer journeys cannot necessarily be replaced by public transport and pedestrian modes. This aspect is very difficult for planners to deal with pre-emptively using current methods. Often the displaced traffic causes excessive congestion or is only moved to other residential areas, causing discontent in the community (Donovan, 2022).

By modelling the implementation of a LTN policy in simulation software, it is possible to extract data that is not available through real-world trials. This data is more complete in that the entire policy and surrounding area can be modelled, so that bottlenecks can be identified wherever they may appear. Additionally, as all aspects of the system are computed, it is possible to extract information in forms that would not be possible with simple observation, for example emissions, stationary times, and average journey times. Once the simulation is fully parameterised, experimentation can also be carried out to test the effects that different approaches may have on the resultant traffic behaviour. These benefits are all at significantly lower impact to the public and lower cost compared to pilot schemes.

To produce specific results, it was important to give a clear specification of the problem that is being investigated. The method developed relates to the modelling and assessment of road networks and traffic situations that have the following characteristics:

- Is a pre-defined street network with little capacity for major structural changes.
- Is a high density urban environment.
- Has left hand drive traffic rules.
- Significant level of through traffic present.
- Has limited count data available (approximately five locations).

These are the characteristics chosen as they represent a proposal for implementation of LTN concepts in the city of Bath, as described in Section 3.4. Consideration is taken throughout to mention alterations that may be made for differently characterised problem cases. By following the characteristics throughout, consistency is maintained between the simple and Bath models, which ensures incorrect behaviours that may occur when applied to Bath are identified in the simple model.

3.2 Proposed Method

The proposed method is outlined here in general terms to provide a guide for understanding the more in-depth explanation and account of development contained in Chapters 4 and 4.3.1. This also serves to demonstrate the wider applicability of the method should it be reproduced using other technology or in other contexts.

1. Constructing a Network

   To allow for accurate simulation of a road system, the structure of the network must be modelled. This involves the abstraction of key elements such as road lengths, junctions, and traffic light behaviour.

2. Demand Generation

   To provide a baseline for comparative analysis and experimentation, the existing traffic volume and behaviour must first be modelled accurately. This is done by building a demand definition from source and destination pairs that can be used to generate agents during simulation.

3. Validation of traffic system

Ensuring the modelled baseline is valid constitutes an important step to provide confidence in conclusions made from the altered model. This can be done through use of various count-based and heuristic tools.

4. LTN conversion

   The changes proposed in the policy being investigated are applied to the baseline network by restricting traffic movement across specified boundaries. These alterations correspond directly to the 'modal filters' discussed in Section 2.1.1.

5. Rerouting of demand

   The demand definition constructed in Step 2 is then reapplied to the updated network and behaviour is simulated. Agents are forced to alter their routes according to the new network, finding the shortest path available to their destination.

6. Analysis of changes in traffic behaviour and identification of potential bottlenecks in the network

   The resulting simulation can then be analysed to understand the impact of applying an LTN policy. The simulation software provides extensive output that includes detail for the entire network and can be directly extracted for visualisation or used for further computation.

   Comparisons can be made to the pre-policy simulation to learn about changes in traffic volume, journey time, congestion, emissions, and other variables. The simulation can also be used to predict issues that may require changes to the policy or road infrastructure. This includes identification of bottlenecks caused by displaced traffic.

3.3 Simple Study

The first step of developing the method and demonstrating its use was applying it to a simplified artificial traffic system, this had the following benefits for the development process:

- Allowed an incremental establishment of the method as we can consider factors like network generation and demand modelling independent from complexities caused by converting from real world data which can be cumbersome to work with. This also allows for development of the core ideas involved in the techniques without a requirement to accurately represent a specific context which would distract from the general problem.
- Provided a controlled environment for testing, where the expected behaviour was known, so that any bugs/odd behaviours caused by issues with the method were clear when they appeared. This means that when applied to the more expansive (in terms of size, intricacy of road network, and complexity of demand generation) application in Bath, conclusions can more confidently be drawn from emergent behaviour.

This artificial application also has benefits in clarifying some complex concepts relating to LTNs and the method's specifics, it does this by:

- Providing a supplemental example of applying the technique in addition to the real use case described in the Bath example. This helps to make clear the key concepts of the method, especially as the simple example is decoupled from many implementation complexities.
- Providing a demonstration of the traffic patterns described in the Literature and Technology review section and a clear example of the benefits and outcomes of implementing LTNs (due to the lack of confounding elements caused by the complexity required to accurately model a complex urban environment).

3.4 Bath Study

In January 2022, B&NES Council carried out a public consultation into a proposed plan to implement a “cell based” traffic scheme in Bath city centre (Sumner, 2022). This candidate proposal was selected as an example for the development of this investigation method due to its embodiment of LTN concepts and goal of reducing through traffic.

The proposal, as part of the council’s plan to reach carbon neutrality by 2030, is inspired by the 2017 Ghent Circulation Plan that used a cell-based model to stop vehicles from crossing the city centre (City of Ghent). In line with principles set out in, SomersetLive (Sumner, 2022)quoted Councillor Warren of B&NES council saying, “Reductions in car use can only come if we start providing more cycle facilities, better public transport facilities and a safer transport network”. This demonstrates that the purpose and methodology of this plan match those that are to be investigated.

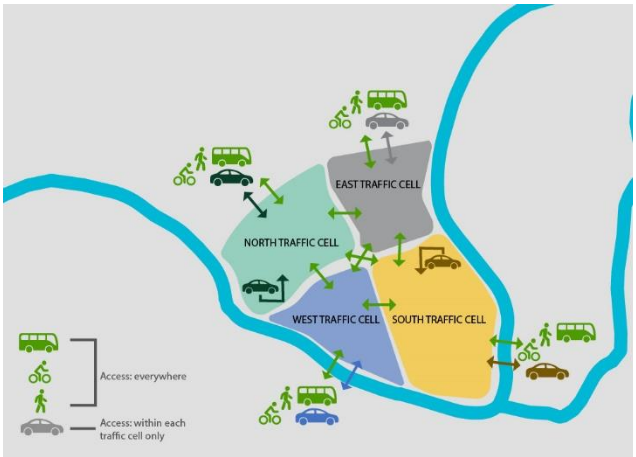


Figure 3.1 - Map showing example traffic cell implementation in Bath, from Bath and Northeast Somerset Council (Bath and North East Somerset Council, 2022).

The plan itself, shown in Figure 3.1, would split the city centre into four cells with no motor vehicles allowed to move between cells and one or two entrance/exit points for vehicles to exit each cell (Bath and North East Somerset Council, 2022). Public transport, pedestrians, and cyclists will be allowed to move freely, with a planned increase in infrastructure elevating the benefits of choosing these forms of travel.

This proposal was deemed to be suitable for investigation as it constitutes a truly feasible use case for the proposed method. SomersetLive reported that critics have questioned the validity of such a scheme as unlike Ghent, Bath has no ring road or tram network. By modelling the resultant traffic, the impact of these criticisms can be investigated, and experiments can be done to see how the plan can be altered to alleviate them. Additionally, data can be extracted to demonstrate to the public the potential benefits and to further inform the council’s projections for emission reduction.

4 SIMPLE MODEL

To give a clear example of applying the proposed method, it will be implemented on a small scale and simple grid-based road network. This artificial scenario provides a demonstration without the need to consider the complexities involved with context specific traffic modelling. It is also an important step of the scientific process for agent-based system design as it allows for easier identification of errors so that when applied to more advanced use cases, emergent behaviour can be more confidently considered as valid.

See Section 4.1.4 and Appendix D for information on how to access all files discussed.

4.1 Implementation

This section describes the initial development and application of the method, at the end a summary is provided along with information on how to access all requisite files.

4.1.1 Network Construction

To meet the goal of this section, the created network should accurately represent the road system-related characteristics of the problem (as described in Section 3.1) with minimal additional complexities. It was constructed from scratch using *netedit,* a graphical network editor included in SUMO (German Aerospace Center (DLR), 2022). This tool provides an interface for creating and editing models of road networks represented by XML (a standard format for structured data) files that are compatible with SUMO's full suite of features. The process of generating the grid network (Figure 4.1) and the reasoning behind it are discussed in this section.

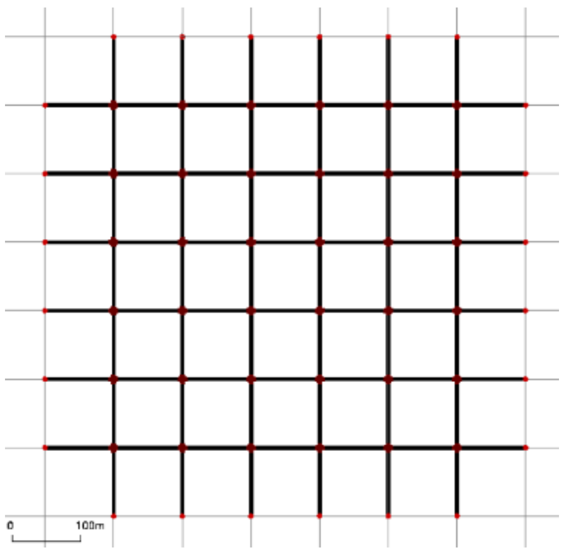


Figure 4.1 - Basic grid network, shown with *netedit* grid overlay. Red areas represent junctions, blacklines are two-way roads.

To represent a suitable candidate road system, the network structure must incentivise through traffic. A grid matches this criterium as when travelling from one outer edge to another, the shortest path generally involves entering the interior of the network, rather than using the exterior road. The grid structure was chosen for its simplicity as the traffic movement will be uniform and predictable, allowing unintended behaviour to be identified easily. Dead end roads were used along the edge of the grid, this allows for easier representation of through traffic volume when modelling demand (further discussed in Section 4.1.2).

Defining the size of this network is a balance between maintaining simplicity and adequately representing the high-density urban environments that are being modelled. A smaller network allows for easier identification of undesired behaviours. However, if the network is made too small, for example a single grid square, when traffic behaviour is modelled it would not match the larger urban environment. The size of the chosen grid is 700m x 700m with roads every 100m. As this was considered to be the smallest plausible structure for an independent traffic system in an urban environment.

By default, *netedit* produces roads with right-hand traffic, to be consistent with the specified characteristics the network should implement left-hand traffic behaviour. To achieve this, the option was selected in the *netedit* Processing Options > Processing > lefthand dialog.

4.1.2 Demand Modelling

To model the effects of implementing an LTN strategy on the road network, we first need to accurately represent a feasible system of traffic demand. We will utilise this demand when constructing the LTN simulation to ensure the variable is kept constant.

In SUMO, the network demand is defined by a route file that contains a time of departure and route for each vehicle that appears during the course of a simulation. SUMO provides a variety of methods for producing this, including manual, statistics based, and count based approaches. SUMO provides four Python scripts for generating demand from observation point data, each suited for different forms of input (German Aerospace Center (DLR), 2022). Due to the limited amount of road count data available the tool *routeSampler.py* was chosen as it requires edge (road) based counting data and does not require full coverage of entrances and exits to the network.

To produce accurate results from *routeSampler.py*, a file containing routes must be provided alongside the count information. The script then fulfils the count data by selecting from the route file. This produces a demand that closely matches the distribution of the original route file but is scaled to match the count data.

Forming demand distribution

The recommended method by SUMO for generating the base route file is to apply the script *randomTrips.py* on the grid network (German Aerospace Center (DLR), 2022). The random distribution of vehicle paths it produces can be tailored through optional parameters to the desired form. To generate a demand that fits the required characteristics, the 'fringe-factor' parameter was set to five. As seen in Figure 4.2, this results in routes that match the desired characteristics of high through traffic by increasing the frequency of routes that start and end at dead end junctions. As discussed in Section 4.1.1, the network was designed with many edge roads to support the use of this option. The output of *randomTrips.py* consists of a list of start and end locations with corresponding departure times that can be passed to SUMO's *duarouter* tool to generate a shortest path route for each. The flag 'junction-taz' was set with both functions so the trip start and end locations are junctions rather than network edges. This allows the demand elements to be reapplied to the converted network as described in Section 4.1.3.

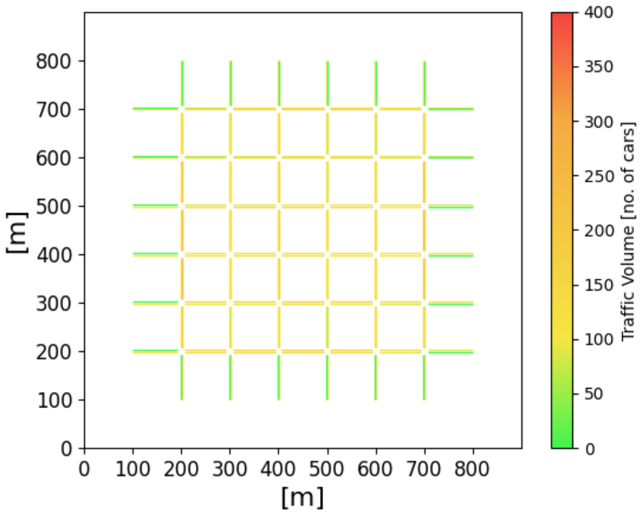


Figure 4.2 - Demand produced by randomTrips.py with fringe-factor value five

Scaling the demand volume

To accurately model the traffic system, this randomly generated distribution must be scaled up or down to the correct number of vehicles. To follow the problem definition in Section 3.1, this must be done with very limited count data. As this application uses a fictional road network, feasible count data was artificially generated for three edges chosen to characterise different parts of the network. By running the simulation with the demand generated by *randomtrips.py* the number of vehicles that use each edge can be extracted. For the selected edges, this base usage was then increased by 50% to manually form feasible count data.

The demand in the whole network must be scaled up to match this data. However, three count points is insufficient coverage for *routeSampler.py* when applied to a network of this size. Figure 4.3 shows the traffic volumes for a simulation run using the demand generated from *routeSampler.py* with this count data only. As can be seen, the demand is overfit to the limited data, only scaling up the number of routes that pass through the count points with no other routes being implemented.

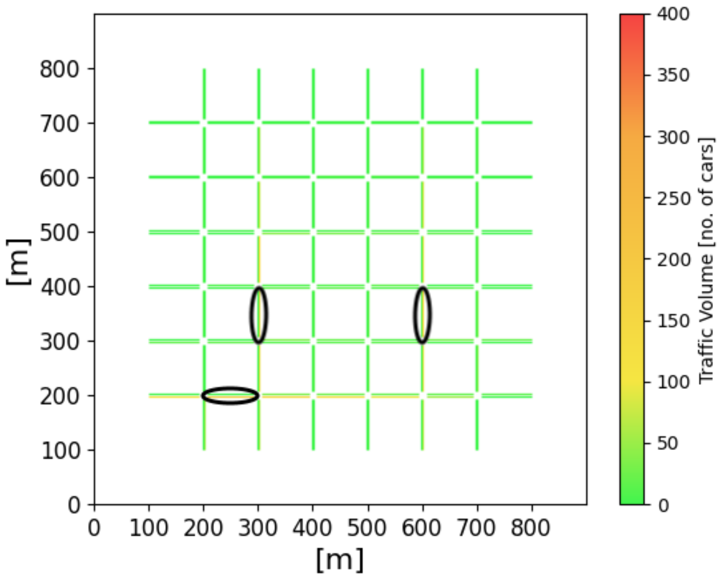


Figure 4.3 - Demand generated by *routeSampler.py* using count data only from the circled edges

To solves this issue, a simple Python script, *scaleDemand.py,* was developed to generate estimate count data covering an entire network using a representative route distribution and limited count data (code available via GitHub link in Appendix D). SUMO's *sumolib* library functions (German Aerospace Center (DLR), 2022) were used to extract the vehicle numbers from a simulation run utilising the distribution data (such as that visualised in Figure 4.2). For all edges with count data, the script then calculates the average ratio between the count based and simulated values. The vehicle counts for all edges in the distribution data without count data available is then scaled to produce a set of count data that has values for all edges in the network. To produce desirable results, it is important that the limited count data follows the same pattern as the distribution data. If one count has double the volume of the corresponding edge in the distribution data and another has half of its corresponding value, they would cancel out. As the purpose of the distribution step is to accurately reflect the demand pattern, this is not an issue when using accurate count data and sufficiently detailed distribution data.

This expanded count data can then be used with *routeSampler.py* to create a route file that matches both the desired distribution and volume. The results of a simulation run with this route file can be seen in Figure 4.4, it is clear that the distribution matches that shown in Figure 4.2 and it is scaled up by approximately 50%. Further validation of these results is described in Section 4.1.3.

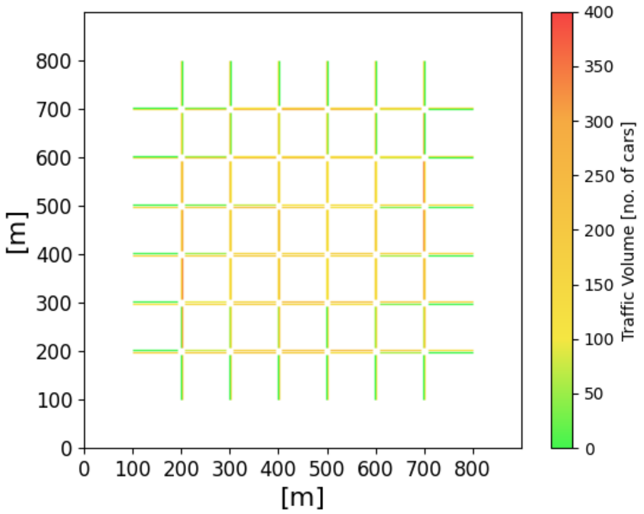


Figure 4.4 - Demand generated by using routeSample.py with expanded count data

4.1.3 Conversion to LTN

The network was then converted to a LTN based structure. The effect on traffic can be simulated by applying the core demand elements from the standard grid the effect on traffic can be simulated. by applying the core demand elements from the standard grid.

Adapting the network

To apply Low Traffic Neighbourhood policy, the grid network was split into four distinct 'traffic cells'. Each traffic cell has a single entrance/exit allowing for traffic to move onto the 'ring road' with no other paths in or out. Thus, traveling via the ring road is the only way for a vehicle to move between cells or from one edge of the network to another.

Network A and B, shown in Figure 4.5, were candidate structures considered due to the clarity with which they demonstrate the boundaries of the traffic cells. However, after consideration they were not suitable as the they both involve alterations to road geography and the problem definition specifies that the road network should not undergo major structural changes. This restriction also has technical benefits in the demand and analysis stages as it is important that the ids of edges and junctions in the new network match their corresponding value in the base model. By keeping the structure, the same the values can be directly copied en-masse and manual renaming is avoided.

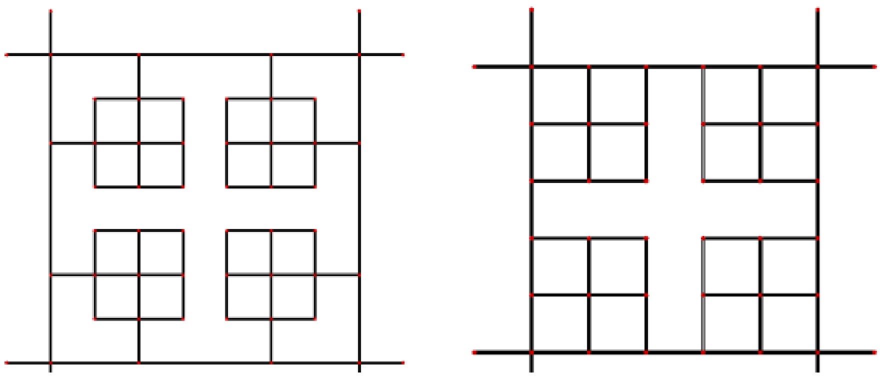


Figure 4.5 - Prototype networks A (left) and B (right)

The final converted network can be seen in Figure 4.6, with each traffic cell highlighted according to the colour scheme used in Figure 3.1 and the ring road and exterior edges in black. The traffic restriction were enforced in *netedit* by altering the rules at junctions: a cell is surrounded by dead end junctions (shown in Figure 4.7), with a single entrance/exit junction (shown in Figure 4.8) allowing access to and from the ring road, these are highlighted in blue in Figure 4.6. The dead-end junctions can be viewed as modal filters implemented onto an existing road network, disallowing motor traffic directly between cells.

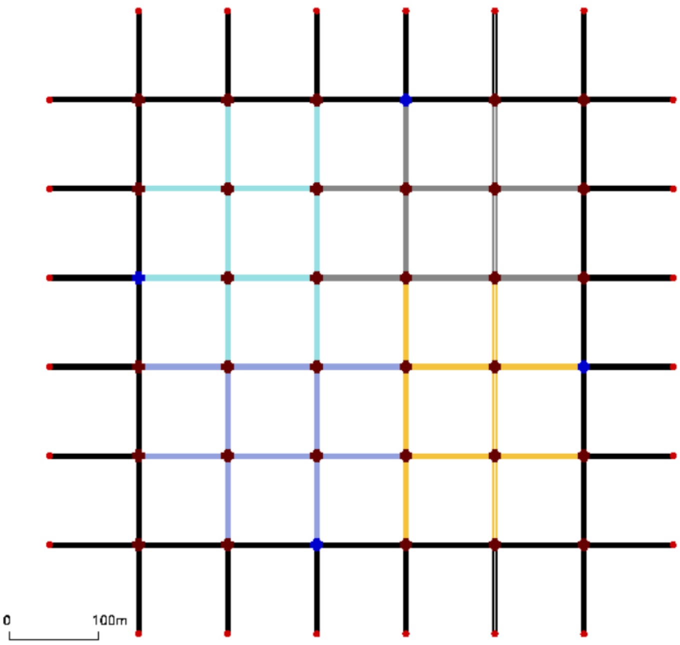


Figure 4.6 - The converted grid network with traffic cells highlighted.

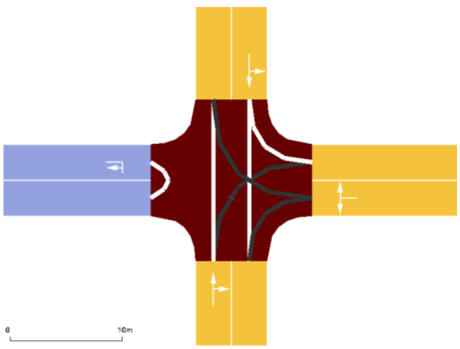


Figure 4.7 - A dead-end junction on the boundary of two traffic cells

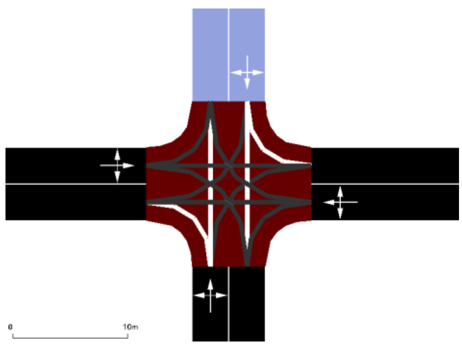


Figure 4.8 - An entrance/exit junction between a traffic cell and the ring road

Applying demand elements

To accurately model the impact of LTN implementation, we model how the pre-conversion traffic demand is altered when using the new network. To allow checking against count data and full simulation of the scenario, the pre-conversion demand data is stored in the form of a route file, which lays out the entire path for each vehicle. As the network has been altered significantly, many of these routes are no longer valid. To ensure the demand is kept consistent, the start and end location of each route must be extracted whilst maintaining the associated departure time. With this data in trip format, the same as generated by *randomTrips.py*, the shorted path can be recomputed for each vehicle.

To generate feasible routes from the pre-conversion demand data, the trip file must contain trips between junctions rather than individual edges. This allows the use of the option 'junction-taz' with the *duarouter* tool, by implementing this option, a route is calculated from one junction to another with the vehicle starting movement on whichever outgoing edge produces the shortest path. As new routes may be significantly different, using edge to edge trip data was deemed insufficient as it produces paths that require excessive turnarounds that do not reflect real traffic behaviour. This behaviour is demonstrated in Figure 4.9, in the standard grid network, the vehicle (represented by the yellow triangle near the top centre) takes a short path (highlighted in red) from its source edge to its destination. However, when the changed traffic rules restrict the vehicle from taking a left turn directly from its starting edge, it must take an exceedingly long path (highlighted in yellow) to turn around and terminate on the correct edge. The desired behaviour is for the vehicle to complete its route once the vehicle reaches the pre-conversion routes end point junction, as generated by *randomTrips.py*.

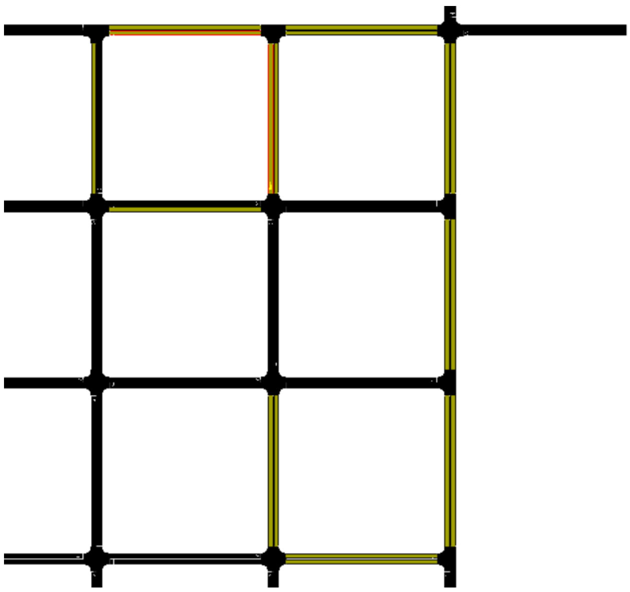


Figure 4.9 - The extended route a vehicle must take to turnaround when converting edge-to-edge demand data, shown in *sumo-gui* using an early prototype grid available in Appendix A.

To match the existing demand data to the original trips generated by *randomTrips.py* and thus allow for rerouting in the new network, the script *tripsFromRoutes.py* was written (code available via GitHub link in Appendix D). Using the *sumolib* functions to parse the XML data sets, the script iterates on the distribution route, scaled route, and trip files that were used to form the existing demand data. By matching vehicles in the two route files based on their path and matching vehicles in the distribution file to trips based on their ids, a relationship is created between a vehicle in the scaled file and the trip it was generated from. A new trip file is then written, containing the vehicle ids and departure times from the scaled route file with their routes mapped to the junction-to-junction trip that they were generated from.

Once the new trip file was generated, it was passed to *duarouter* with the option 'junction-taz'. The result was a new route file with the same traffic volume as implemented in the basic network, with each vehicle now taking a valid route from origin to destination according to the altered network. When simulated in *sumo,* the result is significantly decreased traffic within the cells and a significant increase of traffic on the ring road, as shown in Figure 4.10.

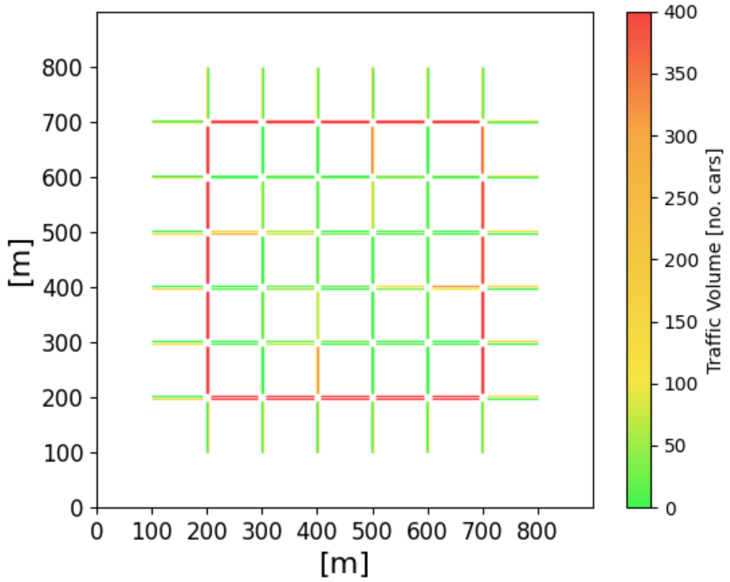


Figure 4.10 - Demand in the converted network

4.1.4 Implementation Summary

This summary serves as a review of the above sections regarding the method's implementation as well as a guide to accessing the scripts, configuration files, input data, and output files that were discussed. Files pertaining directly to this section can be accessed via the link in Appendix D in the folder 'simple'. With the correct SUMO installation, the Bash script 'commands.bash' can be used (with the option -f 2) to automatically execute the necessary steps once the networks have been designed.

1. Network construction
    - The artificial network was constructed using *netedit*
2. Demand generation
    - Generate a random set of trips for sampling using *randomTrips.py*
    - Route those trips using *duarouter* with option "junction—taz"
    - Run a *sumo* simulation with the random routes to get the distribution output data
    - Using *scaleDemand.py,* scale the distribution output data to match the count data
    - Generate a route file that fulfils this scaled demand using *routeSampler.py*
    - Run *sumo* to simulate the baseline traffic system
3. Network conversion
    - Use *netedit* to edit junctions to match the proposed scheme
    - Ensure edge and junction ids are kept the same
4. Demand conversion
    - Run *tripsFromRoutes.py* to get trip data with the volume matching the baseline demand element
    - Route the trips with *duarouter* with option "junction—taz"
    - Run *sumo* to simulate the projected model

A Bash file, 'vis_commands.bash' is also available that can be used to plot volume data for the grid.

Input count data is stored in the file 'countData.xml'.

4.2 Validation

To ensure the results of this controlled investigation were as expected and thus support the validity the method, a series of tests were undertaken on both the base demand element and the final LTN scenario.

4.2.1 Validating the base demand

To validate that the base demand modelling produced the desired traffic volume, a SUMO simulation was run with an 'edgeData' output listed as an additional file. This form of output produces an XML file with information regarding each edge of the network during the simulation. This includes average vehicle densities, road occupancies, and speed as well as information about the total numbers of vehicles. The desired value to compare to the count data is the total number of vehicles to have entered the simulation, as this most accurately reflects how vehicles would be recorded in a real-world study. The value for each counted edge was extracted from the output manually then compared to the artificially constructed count value to confirm they matched closely, see Table 4.1.

Edge ID	Count Value	Output Value
-gneE100	135	132
-gneE96	219	200
gneE89	125	109

Table 4.1 - Input vehicle count and simulated output

The slight difference in two of the cases was deemed to be caused by the limitations of using a finite set of routes to generate the demand. As routes pass through many edges and they must be selected in their entirety, increasing the number of vehicles that pass a particular edge by adding a route can have a knock-on effect to other edges. The *routeSampler.py* script attempts to avoid overflows, hence the values are slightly lower than the target value. The method was deemed to have sufficient precision for this investigation as the divergence by ~20 vehicles is relatively small when considering the network as a whole.

4.2.2 Validating demand conversion

Simulations using both the base and LTN configurations were run with the '—verbose' flag. This provides additional output for that was then compared to validate that the conversion worked as desired (output available in Appendix C).

The value for vehicles inserted was 4925 for simulations which confirmed that the total volume of traffic is consistent before and after conversion.

To check that the route conversion was working as intended, a selection of four routes from the baseline demand file were compared to the matching (by ID) routes in the LTN file. It was checked that the same start and end locations were the same, thus validating that the trips present in the two simulations were the same. It was also confirmed that the trips in the LTN file took a circuitous path to their destination, showing that vehicles were being correctly re-routed according to the network changes made.

4.3 Analysis

Analysis of the baseline and LTN simulations was carried out to gain an increased understanding of the underlying datasets and to further verify the method has the desired results. This also has the benefit of demonstrating how LTN policies achieve their goal and shows how governing bodies may analyse simulated data to improve proposals.

4.3.1 Journey analysis

To assess the effect of the LTN policy on vehicle journeys, both simulations were run with the option '--duration-log.statistics' (German Aerospace Center (DLR), 2022). This provides averaged statistics for trips across the whole simulation duration, as shown in Table 4.2 (exact output available in Appendix C).

Value (avg.)	Baseline	LTN
Route Length	460.45 m	569.43 m
Speed	9.76 m/s	8.25 m/s
Duration	46.88 s	81.58 s
Waiting Time	1.22 s	15.66 s
Time Loss	12.41 s	38.45 s

Table 4.2 - Statistical output from baseline and LTN simulations

There is an increase in average route length of 108.98 meters from the baseline to the LTN model, an increase of 23.67%. Such a significant increase shows that the LTN policy has forced traffic to use longer routes. This fits the stated aims of LTN policy as by increasing car journey length active transport modes are made more desirable by comparison.

The wait time value represents the average time vehicles spend stationary waiting at junctions, it therefore represents the congestion present in a simulation. The almost 13 times increase seen in the LTN simulation is caused by vehicles attempting to join the heavily populated ring road and the roads that lead to it within each LTN segment. The time loss value, representing the "average loss of time due to not travelling at the desired speed", is a more general congestion metric that takes into account waiting time. Although this congestion causes car journeys to be longer, thus fitting the goal of LTN policy, councils do not tend to want high congestion on their roads as it increases pollution and inconveniences residents. A policy maker seeing this projection may use it to inform decisions to reduce potential congestion, for example implementing roundabouts or adding turning lanes to the ring road in advance of policy changes.

As congestion is not an intended consequence and may therefore be reduced or removed, it is useful to know the increase in travel time without waiting time, i.e., only caused by the longer route. This could be used to decide whether the LTN policy has sufficiently large impact by comparing to estimated travel times for walking or cycling. By subtracting

One of the benefits of simulations is the comprehensive data generated, which allows interpretation of the impact of potential changes on the state of the network. In this case, the average journey duration without congestion can be calculated by subtracting the average time loss from the average duration (baseline – 34.47 seconds, LTN – 43.13 seconds). The resultant increase is 8.66 seconds. This is a relatively small increase in travel time so in this case a policy maker may choose to implement further changes to the network such as decreasing speed limits within traffic cells to further disincentivise car journeys. It is also important to note that the changes effect routes differently. For example, routes that start and end on the same edge of the ring road will see a very minimal change in travel time whilst routes that require access to a traffic cell will take longer. This variance has a big impact on the averaged values being used, it would be possible to categorise routes based on type and carry out further analysis, this concept is discussed further in Section 9.1.

4.3.2 Edge-based analysis

To gain detailed insight into how the LTN policy has affected traffic in different parts of the network, SUMO's output tools were used. The script *plot_net_dump.py* is provided to allow visualisation of a chosen attribute on each edge. This tool was used throughout the investigation to provide visualisations such as Figure 4.2 and Figure 4.3 which were used to confirm distribution layouts were as intended by plotting traffic volume.

To generate a file that contains information on the differences between the two simulations, the SUMO provided Python script *netdumpdiff.py* was used, passing the edge data output files from both simulations. The resultant values for change in number of vehicles entering each edge were then plotted using *plot_net_dump.py* to view how road usage has changed, shown in Figure 4.11. As identified in Section 4.2.2, that the total number of vehicles is the same the two simulations therefore this data informs how LTN policy has changed vehicle concentrations. This depicts a large increase of at least 250 vehicles at each ring road edge reflecting the fact that more cars are being forced to use this path rather than travelling through the network. An increase of approximately 100 vehicles is also seen on the roads that lead to an entrance/exit junction between the ring road and each traffic cell. Depending on the nature of the road in question (for example, whether its residential), this may be undesired by policy makers. Analysis of this kind would help identify this issue and allow changes to be made to mitigate it.

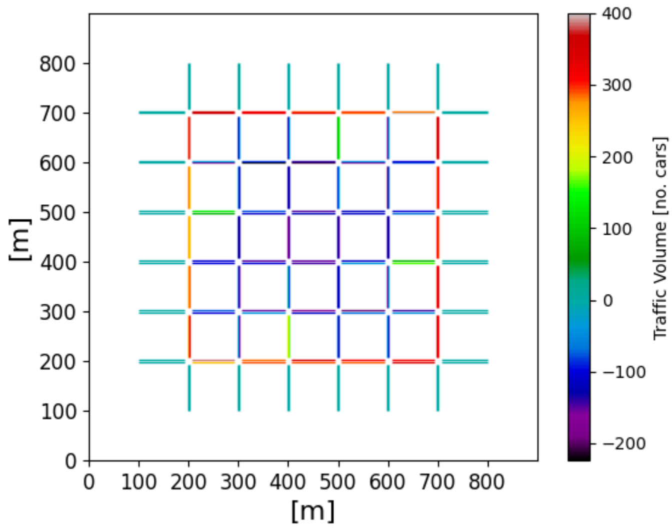


Figure 4.11 - Change in traffic volume after implementation of LTN changes

The results show a clear decrease in traffic within the cells, the greatest decrease was in one of the centre edges which saw a decrease in volume of cars by 224. This means that during the simulated hour, only 13 cars entered the edge. This would be a clear demonstration to an investigating policy maker that the goal of reducing traffic on and making interior roads safer has been achieved. It is clear to see the difference this would make to children playing on a residential street.

During this analysis process an issue was found with the compatibility of the *netdumpdiff.py* SUMO script relating to the permitted type of the simulation duration value. Alterations were made to the script by casting the value to a float rather than an integer, copy of the amended file is available via the GitHub link in Appendix D within the folder 'tools'.

From these results it was decided that the method is valid for replicating demand and predicting the impact of LTN policy. The same method was therefore used for the Bath investigation, with some alterations deemed necessary to suit the particular problem.

5 BATH MODEL

To develop the method further, and demonstrate its applicability to a realistic use case, it was used to investigate the traffic-cell based proposal for Bath city centre (detailed in Section 3.4). This exercise, following the description laid out in Section 3.1 using a limited set of real traffic counts, demonstrated the implementation of the method upon a real-life traffic system. Predictive traffic data was generated for implementation of LTN concepts in Bath and the system was parameterised for further experimentation and analysis. Traffic was modelled for a single hour of simulation as this was deemed to be sufficient to see any large congestion areas appear.

See Section 5.1.4 and Appendix D for information on how to access all files discussed.

5.1 Implementation

5.1.1 Network Construction

As a starting point for both base-line demand generation and creating a representation of the planned structural changes, the current Bath road network was parameterised for use in SUMO simulations. This process involved:

- defining the area to utilise,
- importing layout and road information from Open Street Maps data, and
- using *netedit* and other tools to refine the imported network for simulation usage.

The resultant SUMO xml file, as displayed in *netedit,* is shown in Figure 5.1.

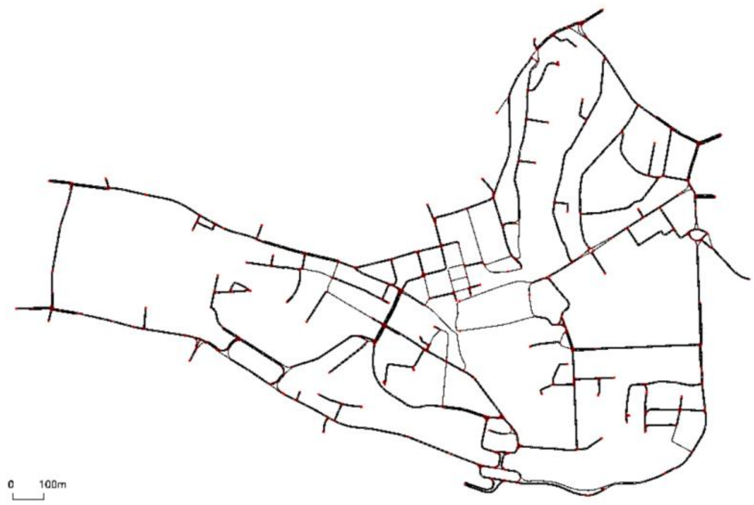


Figure 5.1 - The final baseline network for Bath

Selecting location and dimensions

Defining the boundaries of the simulated traffic system is an important step that impacted all future decisions. The structure and extent of the network chosen also defines the scale and available information in the resultant model. It is essential to model all traffic that may influence behaviour in the network, including through knock on effects caused by congestion at junctions and edges. However, modelling superfluous traffic causes additional processing time and involves unnecessary manual configuration.

The smallest possible network size that still represents all relevant traffic data was optimal because there is only limited traffic data available within Bath city centre. The utilisation of count data is maximised when using the smallest possible network that contains all relevant count points. This allows more accurate modelling of the baseline demand which will, in turn, lead to a more accurate final model.

The plans for traffic-cell policy in Bath, detailed in Section 3.4, were used to decide which roads were relevant to the investigation, the plans for traffic cell policy in Bath, shown in Figure 3.1 were referenced. As these preliminary plans do not contain specifics regarding the individual roads or cells, a feasible application was constructed following the principles discussed in Section 2.1.1. First, a 'ring road' system was defined. As Bath does not have a dedicated road that fits this description, a selection of major roads were chosen that together, closely encompass the area defined in the policy description. This chain of roads, seen as the outer path in Figure 5.1, consists of the following roads:

- Upper Bristol Road
- The Paragon
- Bathwick Street
- Claverton Street
- Lower Bristol Road
- Windsor Bridge

Being A roads, these were chosen as boundary edges because they have the requisite infrastructure (for example sufficient lanes, roundabouts) to handle the high traffic demand. There is a limited number of major roads in the area and any other roads would have been residential or commercial. Simply diverting traffic to these roads would only move the problem to a new area.

Also, it does not align with the aims of LTN policy to include A roads as part of a cell. When compared to the smaller interior roads, there are fewer store fronts and poorer pedestrian connectivity so there would be less to gain by diverting the traffic away.

A model that uses the 'ring road' as a boundary for simulation sufficiently captures all relevant traffic behaviour as any traffic outside of the network can be modelled as entering/exiting via dead-end edges along the exterior of the ring road. The primary feature that this model would lack is the ability to predict how vehicles would change their routes to avoid congestion on the ring road. This would lead to through traffic in the exterior residential areas which is an undesirable feature from the policy maker's perspective. However, this behaviour can be deduced from the network as, wherever there is excessive congestion on the ring road, it can be assumed that traffic will divert to avoid it. Thus, any congestion on the ring road can be seen as an action point for potential improvements to the policy.

Importing the network

As described in Section 2.2.3, SUMO provides features to import OSM map files and convert them to SUMO networks via the *netconvert* application.

To export the open street maps file, the online OSM 'export' function was utilised (Open Street Maps). The resultant map, as seen in Figure 5.2, consists of a square covering the boundaries of the area to be modelled as well as some extremities.

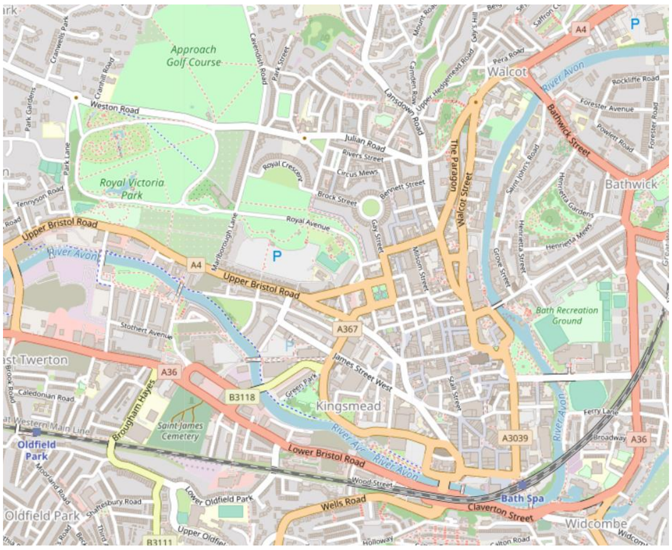


Figure 5.2 - map exported from OSM website

To produce a SUMO network file with accurate traffic rules and the desired road characteristics (shown in Figure 5.3), the SUMO application *netconvert* was then run via a Bash script with the following options:

- left-hand: As SUMO is designed to be a tool for use in any country where traffic rules may vary, left hand driving must be specified explicitly as it cannot be assumed by SUMO and is not included in the OSM map file.
- traffic light options: Rules defining the behaviour of traffic lights in the system are required as OSM maps do not contain data regarding their patterns etc. For this model, the standard SUMO-recommended options were used to maximize flow across junctions.
- junction.join: This option is used to merge any OSM junctions that are very close together into a single SUMO junction. This SUMO *netedit* documentation mentions that "In OpenStreetMap roads forming a single street and separated by, for example, a lawn or tram line, are represented by two edges that are parallel to each other. When crossing with another street, they form two junctions instead of one". If these junctions were not merged, vehicles would have issues waiting in the middle of junctions or on short roads in between many junctions. Following recommendations from the *netedit* documentation all merged junctions were manually checked as the automated option is not very precise and can cause implausible traffic flows.
- typemap – OSM maps do not include some road data that is important for modelling traffic behaviour (such as speed limits and lane priorities). A "typemap" file can be used to define these behaviours. The typemap used was adapted from the SUMO provided typemap "osmNetconvertUrbanDe.typ.xml" available from in the SUMO netconvert documentation (German Aerospace Center (DLR), 2022).

Alterations were made to filter out pedestrian paths as these were not being modelled in this case.

Figure 5.3 - OSM map converted using *netconvert* shown in *netedit*

Refining, cleaning, and validating the network

Since the OSM map extracted was a rectangle, the imported SUMO network has many roads exterior to the ‘ring road’ that, as discussed previously in this section, are not relevant. Edges outside of the boundary were deleted using *netedit* to ensure only the desired parts of the traffic system were modelled, using *netedit* edges outside of the boundary were deleted. Some edges that directly connected to the ring road were excluded from this process for use as ‘fringe edges’ in the demand modelling stage. These are directly equivalent to the edges discussed in Section 4.1.2. See Figure 5.1 for a depiction of these changes.

This import process is effective at bringing a large amount of geographical information into the SUMO network format and thus avoided a significant amount of manual labour in *netedit* or remapping the network and defining each road. However, the automatically generated traffic rules and junction controls are imperfect with many inaccuracies and fault, which would cause excessive congestion and unrealistic traffic behaviour. To account for this, the entire network was manually checked to ensure the simulated model accurately represents the city of Bath.

In addition to *netedit,* a series of tools were used to validate and refine the network. The SUMO supplied python script *netcheck.py* was used to identify any areas where unconnected ‘islands’ were formed. These were either caused automatically as part of the import process, where their original connections fell outside of the selected area, or by a mistake in the manual editing of the ring road. Once ‘island’ components were identified, the 'locate' function in *netedit* was used to find them and the issues were rectified manually. This tool only checks for connections between edges; this is a limitation in this application as junction-to-junction trips are used to generate demand. This means that in cases where two edges are connected but junction flow logic or one-way streets restrict vehicles from travelling between two junctions, no issue is identified. If not found manually, this can lead to errors in demand generation as trips are formed by selecting junctions at random without checking for a valid route between them.

It was also necessary to perform some manual inspection of the network, particularly in areas where net convert had provided warning messages and in junction logic. As the amount of data available in OSM is limited, junction and road characteristics must be guessed by *netconvert*. As SUMO is designed for global usage, many of the guesses do not align with United Kingdom traffic law in general or more specific Bath based rules. To identify and rectify these cases, Google Maps and StreetView were used to provide an understanding of what the existing traffic rules were. As traffic rules are subject to regular change, it was important to ensure that the imagery used was up to date with current layout. Google maps was used for this purpose as it is updated more regularly than OSM. Issues found and rectified using this method included:

- Misidentification of roads as two-lane rather than one-lane due to dedicated on-road parking along the entire length
- Bus-only lanes and junction routes being treated as normal roads. For the purposes of this investigation, public transport was not being modelled so for clarity these lanes and routes were simply removed.
- U-turns being allowed at all junctions.
- Turning lanes not being identified and handled properly. To fix this, the rules at junctions were changed to match road markings as seen on StreetView.
- Removing/altering roads that do not allow through traffic

Google maps and street view were also used when editing junctions that had been identified by *netconvert* warning messages for inspection. Due to the highly detailed road geometry data in OSM, junctions were often very complex with splitting one-way lanes and multiple stage traffic lights. Much of this detail can be removed, as shown in Appendix B, as it is not relevant to the behaviour of agents in the system.

As a final validation step, a simulation in *sumo-gui* was run with the network. This application provides a visual representation of a running simulation, with vehicles represented as yellow triangles (Figure 5.4). Vehicles were generated using *randomTrips.py* as described in Section 4.1.2, the traffic volume was then increased using the sumo-gui 'Scale Traffic' feature. As a stress test a large amount of traffic that was generated, bottlenecks and high congestion areas could be identified and investigated. For example, where two nearby junctions were not merged properly, vehicles would remain stationary in the centre of the junction, blocking all other traffic for extended periods of time, this is the scenario shown in Figure 5.4. To solve this problem, the junctions were merged, and incoming lanes were altered to match the traffic rules as pictured in StreetView. This technique identified other issues included incorrect lane priorities at roundabouts and bottlenecks caused by inaccurate one-way roads.

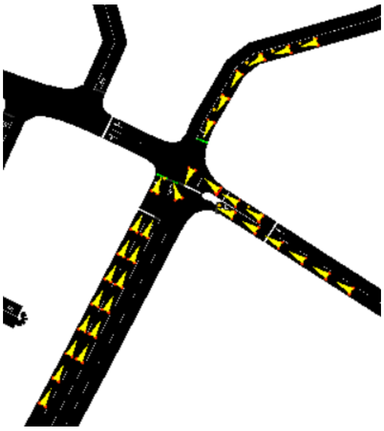


Figure 5.4 - Screen capture from *sumo-gui* showing congestion caused by a poorly converted junction.

5.1.2 Demand Modelling

Acquiring count data

To ensure that the modelled traffic accurately represents the existing volume and demand, count data was sourced from the Department of Transport. The relevant data had to be extracted from the dataset, which includes counts from across the B&NES council area over a period going as far back as 2000.

Although there was more recent data available from 2020, data from 2019 surveys was used because it is not affected by any temporary changes caused by COVID-19 lockdowns and is more representative of expected future traffic patterns.

Using Google's My Maps website, all locations in the dataset for the relevant time period were mapped by manually imputing their co-ordinates with their count id associated, as shown in Figure 5.5. The map was then used to isolate the count points that fall within the area of interest, shown in Figure 5.6, by comparing against the SUMO network described in Section 5.1.1.

The dataset contains redundant data regarding the types of vehicles counted, which is not relevant to the investigation so it was removed. Additionally, the original data contains counts for each point hourly over the duration of a day. For simplicity and to reduce the impact of random variance, the average over a day was used for demand generation. This excludes count point 38015 (Broad Street) for which the average between the hours of 10 00 and 17 00 were used due to changes in traffic rules existing for these times. This period was chosen as it includes peak time for total traffic in the network and therefore is more relevant to this investigation as it is where the proposal will have the largest effect. The condensed dataset used for demand generation can be seen in Table 5.1.

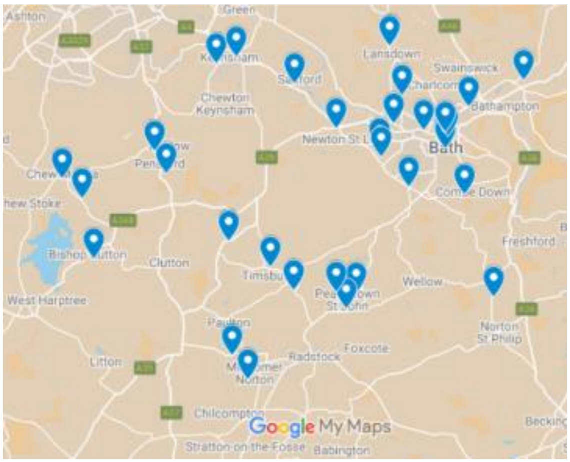


Figure 5.5 - All count points from B&NES traffic dataset mapped using Google My Maps

Figure 5.6 - Count points from B&NES traffic dataset that fall within area of interest mapped using Google My Maps

One drawback to this dataset is an inconsistency in days of the week and times of the year on which it was gathered. As traffic demand shifts over the time periods, it may be that when used together the data does not accurately reflect the traffic volume on a given day. For example, data gathered on a Friday may have higher volume than a Tuesday. To treat these values as if they were collected at the same time could lead to an incorrect perception of the traffic distribution. This would result in it being impossible to produce an accurate representation of traffic that also matches all count point data exactly.

Count point id	Direction	Day of week	Date	Latitude	Longitude	Vehicles
18277	N	Tuesday	19/03/2019	51.38466	-2.3604925	220
27156	E	Friday	26/04/2019	51.385091	-2.3751534	510
27156	W	Friday	26/04/2019	51.385091	-2.3751534	564
38015	N	Wednesday	26/06/2019	51.383	-2.359373	74
38015	S	Wednesday	26/06/2019	51.383	-2.359373	75
48359	N	Monday	03/06/2019	51.377647	-2.3605092	543
48359	S	Monday	03/06/2019	51.377647	-2.3605092	487
75360	N	Friday	15/03/2019	51.378544	-2.3613065	766

Table 5.1 – Condensed count data used for demand modelling (Department of Transport, 2020b)

Once the data was sufficiently processed, a mapping was created between each count point ID and the network edge that most closely matches its location, shown in Figure 5.7. Notably, as described previously, the Broad Street count point location is a special case that does not directly correspond to an edge in the network (as it was removed to reflect traffic restrictions during peak times). To accurately reflect the limited vehicles that are allowed to pass this point (buses and taxis), the two nearest edges in the correct directions were chosen as the mapping.

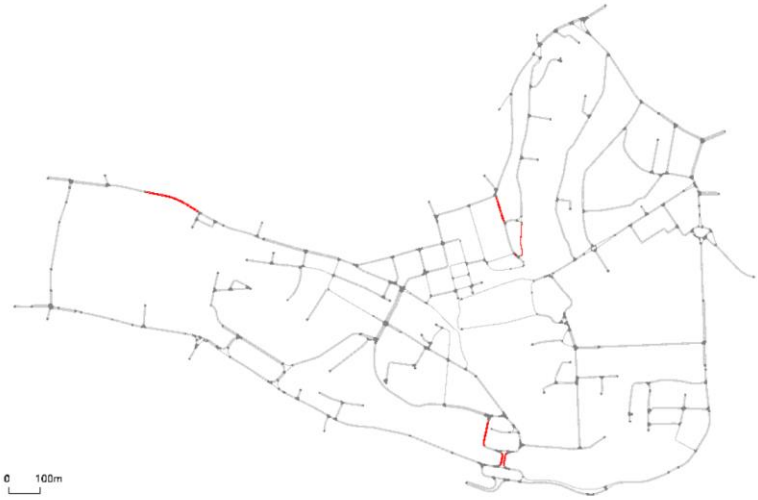


Figure 5.7 - Bath network with counted edges highlighted

Generating demand

As in described in Section 3.2, a baseline demand was generated to represent the existing traffic volume and behaviour in Bath. This was done by characterising the overall demand pattern using general heuristics and count data, then scaling it to match the existing volume.

As the network covers quite a small area with few residential areas, it was assumed that many vehicles would have sources and/or destinations outside of the network. To characterise this, the trip file was generated using *randomTrips.py* with fringe-factor five, resulting in trips being twice as likely to start or end at the fringes of the network. The trips were then passed to *duarouter* to produce edge by edge routes for each trip, which were be sampled to generate the correct demand pattern.

To gain data about what volume of traffic this randomly generated demand produces, a SUMO simulation is run on the Bath network, storing information about how many vehicles enter each edge.

Due to the increased complexity within the Bath network compared to the simple model discussed in Section 4.1.2, traffic demand is less uniform. This meant that the options

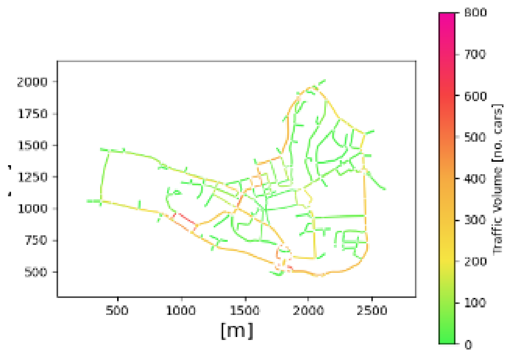


Figure 5.8 - Traffic volume distribution generated by *randomTrips.py* on the Bath network

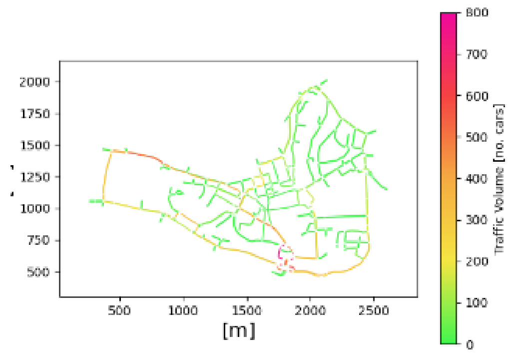


Figure 5.9 - Traffic volume distribution generated by *routeSampler.py* with count data on the Bath network

available for tuning the output of *randomTrips.py* were insufficient to generate a pattern that matched the demand in Bath. When scaling purely random demand distributions it was found that the ratio between the generated volume and counted volume would vary greatly between count points. As when using *scaleDemand.py*, an average is calculated and then used to scale across the entire network, if many

The 'write route distribution' option of *routeSampler.py* was used to generate 'route distribution files' that contain a list of routes with associated probabilities alongside the list of vehicle depart times. This, unlike the basic route file which directly links vehicles and routes, allows for combining files. To generate the final demand, the route distributions of the randomly generated rough estimate and a more precise sampling based on count data were combined; resultant traffic volumes shown in Figure 5.8 and Figure 5.9. This method provided a balance between representing demand across the whole network and fulfilling the limited count data that is available without overfitting.

A Python script, *combineDist.py*, was written to combine the two route distribution files (code available via GitHub link in Appendix D). As the number of vehicles was to be scaled up to meet the demand, the important part to merge was the routes and route probabilities. Using *sumolib* library functions, the XML route file are parsed and the route sections of the two files written to a new file with the vehicles from one of the files as a placeholder. To avoid two routes having the same ids in the new file, a configurable prefix is added to all routes from one of the files.

This combined route file had the desired traffic distributions pattern (volume data shown in Figure 5.10) but requires scaling to match the recorded traffic demand. This was done using the *scaleDemand.py* script, described in 4.1.2, on volume data generated from running a SUMO simulation with the combined route file. The output, artificial count data for all edges, was passed to *routeSample.py* to form a route file that fulfilled the count data. A final SUMO simulation was run to generate the complete traffic volume data, shown in Figure 5.11.

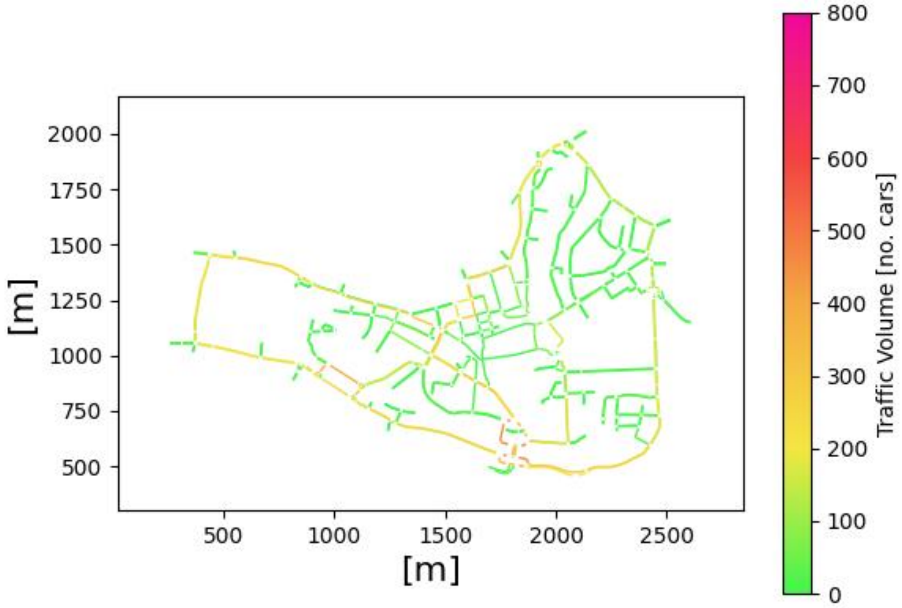


Figure 5.10 - Traffic volume from the unscaled combined distribution in the Bath network

A Bash file was created to call the relevant command line actions for generating the demand, allowing the process to be developed incrementally without the need to repeatedly enter long commands.

To generate general demand independent from the count data and thus avoid overfitting to the existing data, some faithfulness was sacrificed. This was judged to be superior to using a demand that did not adequately cover significant portions of the network. The issue may have been caused by the inconsistency in count point data gathering in terms of day of week and time of year as discussed previously in this section.

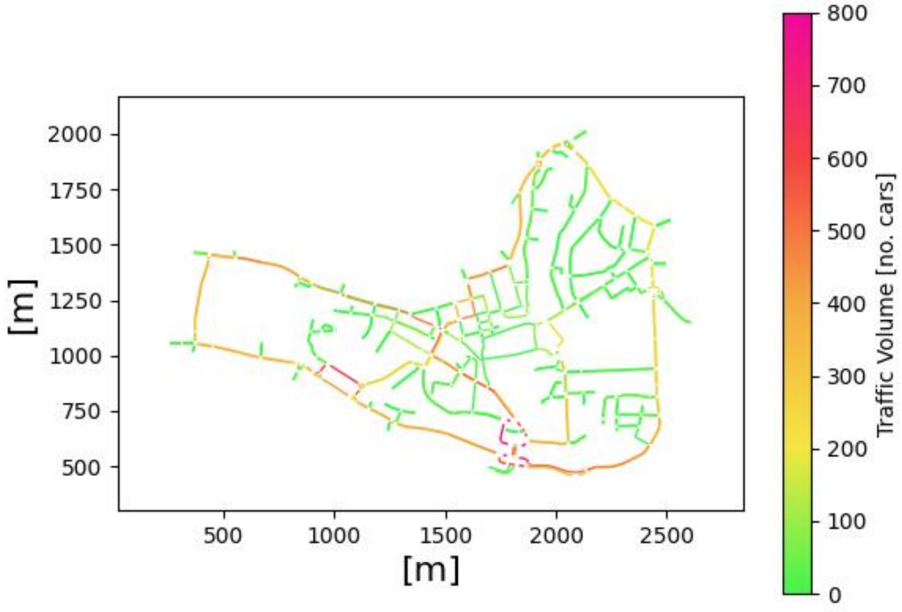


Figure 5.11 - Traffic volume from the scaled distribution on the Bath network

5.1.3 Conversion to LTN

Adapting the network

Due to the preliminary nature of the proposed traffic cell scheme in Bath, there was little information available about the specifics of how it will be implemented. Therefore, for the purposes of this study, a feasible plan was constructed in line with the stated goals and principles.

As stated in Section 5.1.1, a series of A roads were selected for use as a 'ring road' to encircle the traffic cells and act as an arterial route for traffic flowing around the network.

The area to convert into traffic cells was selected following the outlines on B&NES council website, which state the plan would include 'Grand Parade and High Street', 'Dorchester Street', 'James Street West', and 'Green Park Road'. Using these guidelines as well as the B&NES council proposal diagram, see Figure 3.1, a plan was constructed that split Bath city centre into four traffic cells, as shown in Figure 5.12. Following the guidelines, traffic cells were given "one or two" entrance/exit points to the network. These were chosen to maximize the use of existing infrastructure, such as bridges, as these points would act as bottlenecks if there is not sufficient capacity for traffic that needs to move between the ring road and cell. When forming traffic cells with multiple entrance and exit points, it was important that a path from one point to another did not form a shorter route than following the ring road. This would cause through traffic to enter the cell rather than using the ring road, thus undermining the purpose of the LTN policy.

Figure 5.12 - Converted Bath network with traffic cells highlighted

To construct the new network in *netedit*, the cleaned Bath file was first copied over exactly. It was important to ensure that the validation steps described in Section 5.1.1 were thoroughly completed before this was done as any fixes made after this point would have to be carried out identically on both networks to maintain consistency. It is important to use an exact copy so that the id numbers of junctions and edges are kept consistent between the two networks, this allows for conversion of the demand elements and comparison during the analysis process.

The traffic restrictions were then put in place by using the *netedit* 'Connections Mode' to alter junctions between cells to stop traffic moving between them, as shown in Figure 5.14. The interior junctions along the ring road were also converted to dead ends at any points where an entrance/exit point had not been designated, as seen in Figure 5.13. To ensure traffic could flow fully within a cell, some one-way streets with two lanes were altered to allow traffic to move in both directions. This was deemed feasible as the purpose of this one-way system is to control traffic flow, once the restrictions have been applied they will fulfil this purpose sufficiently. The changes to junctions meant that some small interior roads became inaccessible to vehicles following traffic rules. This is acceptable as for demand generation junction-to-junction trips are used so provided there exists a feasible path from one junction to another, inaccessible edges do not cause errors in demand conversion. If these changes were to be implemented in Bath, these streets could be converted to pedestrian causeways with vehicle access only to residents.

Once the network changes were made, *sumo-gui* was once again used with demand from *randomTrips.py* to validate the network, ensuring it was fully traversable and that no unintended traffic flow changes had occurred.

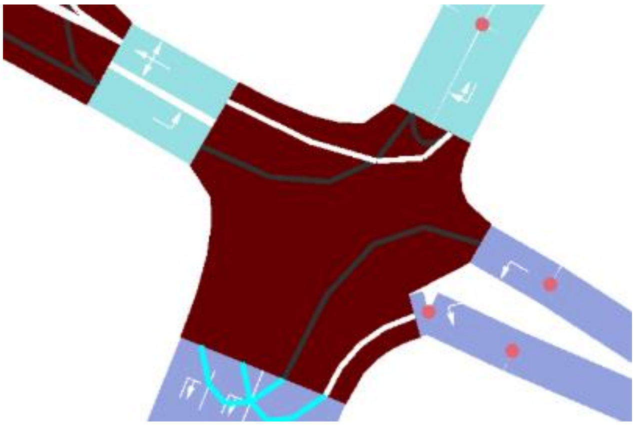


Figure 5.14 - A junction between two traffic cells.

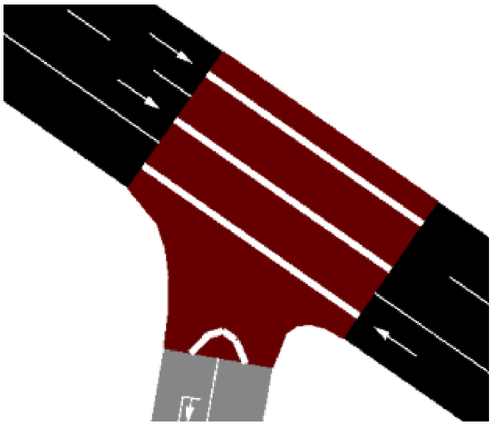


Figure 5.13 - A dead-end junction between a traffic cell and the ring road.

Applying demand elements

With a baseline demand reflecting the existing traffic behaviour and a network with updated traffic rules, the method for simulating the LTN policy was completed as laid out in Section 4.1.3.

First, *tripsFromRoutes.py* was called on the demand elements generated in Section 5.1.2 to produce a junction-to-junction trips file with the correct distribution and volume. Sumo's *duarouter* application was then used with the trip file and LTN-converted network to create shortest path routes that have the same start and end destinations as in the original network and follow the new traffic flow rules.

A *sumo* simulation was then run on the converted network with the new route file to model the traffic under proposed scheme, plotted in Figure 5.15. This simulation is the final projective model of the LTN scheme thus provides the data for analysis and the baseline for experimentation.

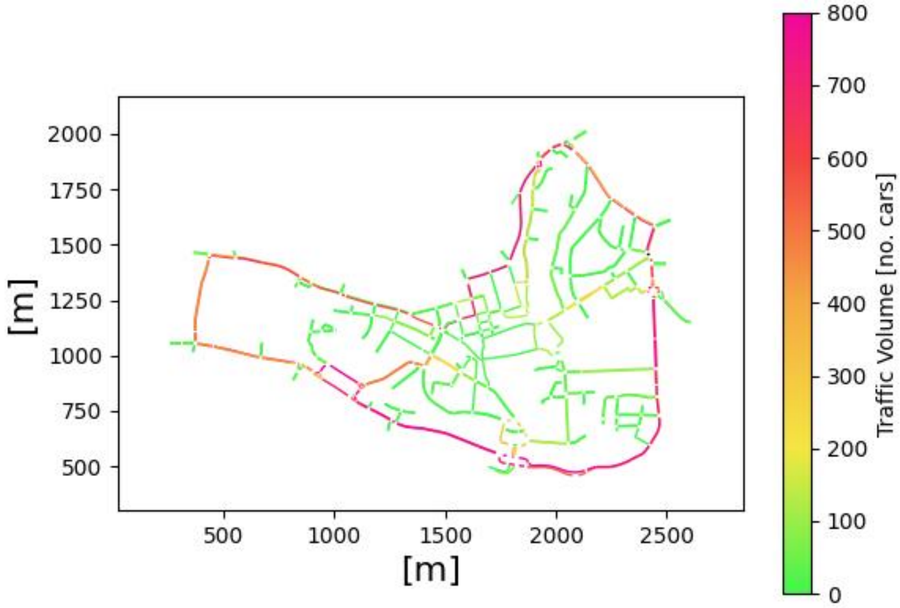


Figure 5.15 - Traffic volume in the converted network

5.1.4 Implementation Summary

This summary serves as a review of the above sections regarding the method's implementation as well as a guide to accessing the scripts, configuration files, input data, and output files that were discussed. Files pertaining directly to this section can be accessed via the link in Appendix D in the folder 'bath'. With the correct SUMO installation, the Bash script 'bath_commands.bash' can be run (with the option -f 2) to automatically execute the necessary steps once the networks have been designed. Files regarding the network import, including the typemap files used, can be found in the folder 'mapConversion'.

1. Network construction
    - Export the network from the OSM website
    - Import the network to SUMO using *netconvert.py* with correct typemap and options
    - Refine the network in *netedit,* refencing Google Maps and StreetView to create an accurate representation
2. Demand generation
    - Generate a random set of trips for sampling using *randomTrips.py* with fringe-factor 2
    - Route those trips using *duarouter* with option "junction—taz"
    - Run a *sumo* simulation with the random routes to get the distribution output data
    - Run *routeSampler.py* with the output data, specifying the output type as a route distribution file
    - Run *routeSampler.py* with the counted data, specifying the output type as a route distribution file
    - Use *combineDist.py* to combine the two files
    - Run a *sumo* simulation with the combined routes

- Using *scaleDemand.py,* scale the combined output data to match the count data
    - Generate a route file that fulfils this scaled demand using *routeSampler.py*
    - Run *sumo* to simulate the baseline traffic system
3. Network conversion
    - Use *netedit* to edit junctions to match the proposed scheme
    - Ensure edge and junction ids are kept the same
4. Demand conversion
    - Run *tripsFromRoutes.py* to get trip data with the volume matching the baseline demand element
    - Route the trips with *duarouter* with option "junction—taz"
    - Run *sumo* to simulate the projected model

A Bash file, 'bath_vis_commands.bash' includes a series of commands that can be utilised for visualisation.

5.2 Validation

5.2.1 Validating the base demand

As in Section 4.1.2, output data was extracted to compare the original count data to the values produced by the simulation. Similar to the simple investigation, there are few count points to consider so the process was carried out manually, see Table 5.2.

Edge ID	Count Value	Output Value	Difference
188924487#0	220	94	-126
-444272568#0	510	360	-150
837573373#4	564	405	-159
1025513686#0	74	70	-4
7985210#1	75	47	-28
37777462	543	522	-21
4479201#0	487	483	-4
3441738#0	766	742	-24

Table 5.2 - Input vehicle count and simulated output

Most of the compared locations displayed a difference between the desired and output values within the acceptable range (~20) discussed in Section 4.2.1. However, three locations showed significant variation of over 100 vehicles: 188924487#0, -444272568#0, and 837573373#4. All these edges, shown in Figure 5.16 and Figure 5.17, are located close to the boundary of the simulated area. This resulted in the finite number of randomly generated routes being insufficient to fulfil the demand without overflowing at other edges. Nearer to the boundary, the number of available routes that pass an edge decrease thus the counts cannot be match exactly.

For this investigation, a conservative estimate was preferred to overflow because any congestion that is observed with less cars in the network would certainly be present in a simulation with more vehicles. Additionally, edge 188924487#0, is very close to the edge removed to reflect time-based traffic restrictions. Similar to the restricted edge, it may have been beneficial to limit the times used to generate the count data as the cars using the restricted road would often also travel this edge. By limiting the time's used, a smaller count value would have been calculated and there would be a smaller difference with the simulated value.

Given the increased scale and complexity of the Bath network results were not expected to be as precise as produced in Chapter 4. Additionally, whilst the count data for the Simple model was artificially produced to match the route distribution, the Bath count data is naturally recorded and subject to random variance in traffic volume. As discussed in Section 5.1.2, there are also inconsistencies present in the environment gathered that mean it is likely impossible to precisely fulfil all the data points simultaneously.

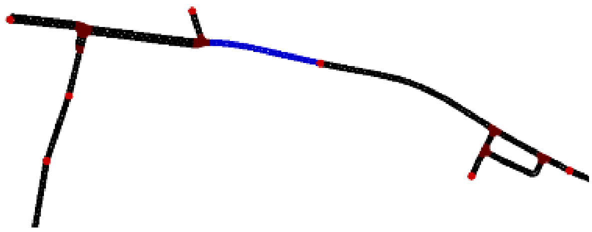


Figure 5.16 - Edges -444272568#2 and 444272568#1 (each direction)

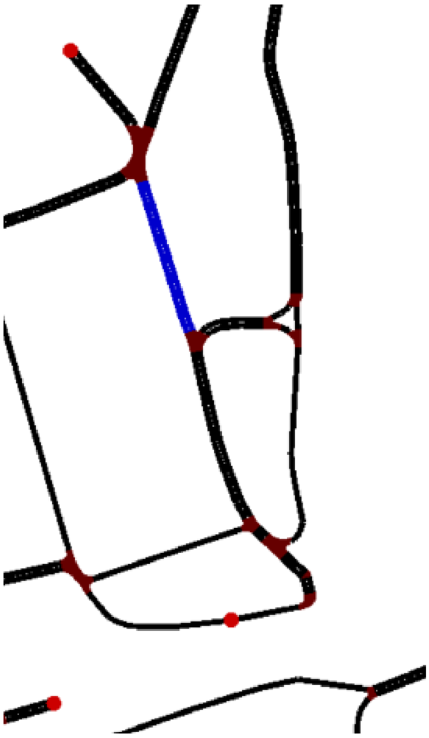


Figure 5.17 - Edge 1025513686#1

Overall, the demand element was judged to provide a sufficient representation of traffic behaviour and scale in Bath as it forms a balance between matching the available count data and considering the distribution of traffic in the network as a whole.

5.2.2 Validating the demand conversion

Following the same procedure as outlined in Section 4.2.2, the baseline and LTN model simulations were implemented gathering verbose output for validation (output available in Appendix C). It was confirmed that the total number of vehicles was identical in both, with a value of 5058. This confirms that the traffic volume used in any analysis of the LTN model is identical to that in the baseline.

Once again, individual items in the route files used in the two simulations were manually compared to confirm that there was a one-to-one correspondence and that the routes for the LTN simulation had been properly diverted. This is supported by comparison of the traffic volume plots produced by the two simulations, shown in Figure 5.11 and Figure 5.15. These clearly show that in the LTN simulation traffic has been diverted away from the traffic cells onto the ring road.

6 RESULTS AND DISCUSSION

By using the Bath investigation as an example, output analysis and interpretation is described. This includes evaluation and visualisation of the basic output data, experimentation with input parameters, and explanation of how a governing body may use results to inform policy making. Additionally, an assessment of the proposed Bath city centre traffic cell policy is carried out, considering the choices made in this investigation regarding implementation details.

6.1 Analysis of results

Much like the analysis carried out in Chapter 4, information about the effects of the LTN policy was gleaned by comparing the traffic behaviour in the two simulations. This was done by investigating the outputs of each simulation using the comprehensive tools provided SUMO.

6.1.1 Journey analysis

To analyse the changes that the new traffic rules made on individual journeys, simulations of the baseline and LTN models were run using the '--duration-log.statistics' *sumo* option, as described in Section 4.3.1 (exact output available in Appendix C). The results, shown in Table 6.1, depict similar results to those seen in the simple model investigation.

Value (avg.)	Baseline	LTN
Route Length	1276.81 m	1870.25 m
Speed	8.47 m/s	8.08 m/s
Duration	155.00 s	242.48 s
Waiting Time	15.05 s	26.17 s
Time Loss	47.01 s	89.22 s

Table 6.1 - Statistical output from baseline and LTN Bath simulations

The average journey length increase by 593.44 meters clearly demonstrates that vehicles are forced to take significantly more circuitous routes in the LTN model. This increase of 46% would have a significant effect on decision making when considering between car usage and alternate methods of transport.

In comparison to the Simple model, there is significantly less change in average waiting time and speed. This is due to the scale and complexity of junction control in the Bath network which reduces the number of bottlenecks present and allows vehicles to move more freely. However, the average duration still sees a significant increase of 64%, this is a desirable outcome as it means that car journeys are disincentivized without causing large congestion issues.

6.1.2 Edge-based analysis

As with the Simple model, the difference in traffic volume for each edge was visualised to better understand the changes in traffic distribution that the policy had caused, shown in Figure 6.1. When following the method described in Section 4.3.2, an issue occurred whereby some edges with traffic in the baseline model did not have any traffic in the LTN simulation. This issue was due to a mismatch in edge attributes, with traversed edges having attributes like 'travelTime' while unused edges do not. To avoid this issue, another SUMO supplied script, *edgeDataDiff.py,* was used instead. This script is more resilient to such issues but can only be used for edge data, which fit the desired use. To ensure the edge IDs matched, a one-way street that had been split into two opposite directions to allow traffic flow within cells required special attention. There was only one such street, for which the two volume values were manually summed and combined under a single id in the LTN output file to correspond to the baseline. This is valid, as the sum of the two edges is the value of total cars passing that point, which is the value that was to be compared.

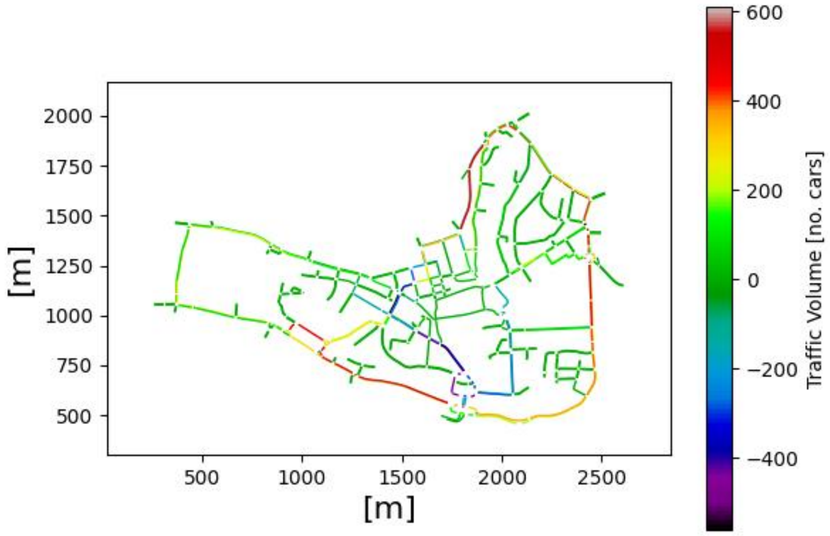


Figure 6.1 - Difference in traffic volume after policy implementation

The results demonstrate the LTN policy's significant impact on traffic behaviour in the network, shifting traffic away from the interior of the network onto the arterial ring road. Areas that saw particularly large increases in traffic concentration were the sections of ring road between certain traffic cell entrance/exit junctions. The central areas that originally had high through traffic such as James Street West and Charles Street saw significant reductions in traffic volume, by up to 48%.

6.2 Experimentation

A benefit of the method is that it provides a baseline for experimentation and a framework for making alterations and comparing predicted results. When announcing schemes and evaluating their success, policy makers often sight the decrease in traffic volume that is observed after people are given reasonable time to adapt their transportation choices by opting for active or public transport. The City of Ghent reported a 12% traffic reduction after implementing their cell-based scheme and Living Streets claims a reduction of 15% for London LTN trials. By use of the *scaleDemand.py* script developed as part of this investigation, it is possible to simulate traffic behaviour in the network taking into account this reduction. This would allow policy makers to plan based on the long-term future of traffic in the area, once the benefits of the LTN has been realised, rather than only when a scheme is first introduced and the demand for vehicle usage is static. Additionally, it could be used to demonstrate to sceptical parties the potential benefits of an LTN scheme during the increase of congestions when its first implemented.

This reduction in demand was modelled and analysed in order to demonstrate the utility of the method for assisting experimentation. The output data from the LTN simulation was scaled by a factor of 0.85 using the *scaleDemand.py* explicit scale factor parameter. This data was then used with the LTN route file as artificial count data for *routeSampler.py* run with the '—optimize full' option. The resultant simulation consisted of 4279 vehicles (84.6% of the 5058 present in the baseline simulation) with a similar traffic volume distribution, shown in Figure 6.2, to the standard LTN simulation. To provide further clarity on the differences between the traffic volumes in the two models, *edgeDataDiff.py* was used to plot them, as shown in Figure 6.3. This shows that with the high density of vehicles on the ring road, this 15% reduction can mean a decrease of over 150 cars per hour. This is a significant reduction in the areas that would see the highest increase in traffic immediately after the implementation of an LTN scheme.

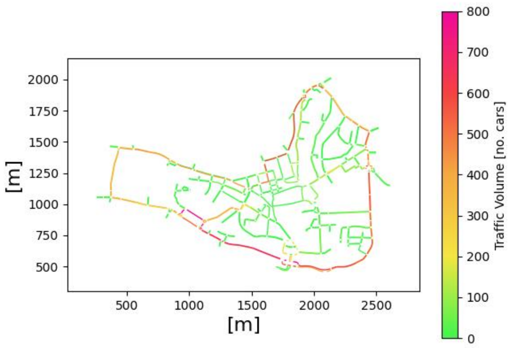


Figure 6.2 - Traffic volume for LTN network with a 15% reduction

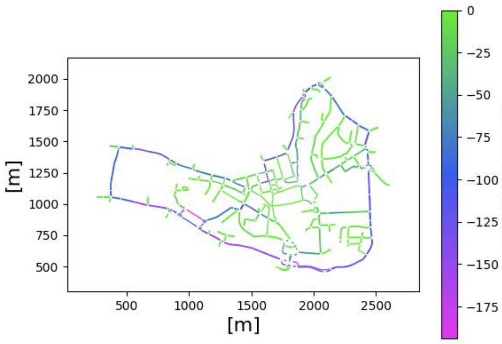


Figure 6.3 - Difference between traffic volume in the LTN and 15% reduced simulations

This experiment is more valuable than simply calculating a 15% reduction for each edge as the result formed is an entirely independent model that can be further analysed and experimented on, additionally it can be used with *sumo-gui* for detailed visualisation.

All relevant files for this section are available via Appendix D in the folder 'bath/ltn/scalingExperiment'.

6.3 Assessment of proposed LTN scheme

When assessing the success of an LTN scheme, it is important to understand that the goal in not optimal flow of traffic around the network. In fact, traffic is deliberately diverted to longer routes to disincentivise travel by car and comparatively improve the desirability of other transport methods which are elevated through infrastructure improvements made in conjunction with the scheme. To assess whether the proposed traffic rule alterations achieve the goals of the Bath proposal and consider potential alterations and improvements, the specific results of the Bath model will be evaluated. As such, an example is provided of how policy makers may utilise the method to inform decision making.

As demonstrated in Section 6.1.1, the traffic rule changes succeeded in the goal of increasing average journey duration and distance. If realised in Bath, this would incentivize the use of pedestrian or public transport as a more direct and faster route could be used for the same journey.

A huge decrease in traffic volume was seen throughout the central areas of the city centre, as shown in Section 6.1.2, roads such as James Street West saw a decrease in traffic by up to 175 vehicles per hour. This significant change in the traffic concentration would make pedestrians significantly safer and potential pave the way to expanding the city's current pedestrianisation usage in the area. As a direct result of this decrease in traffic concentration, roadside pollution like NO2 would decrease significantly, thus achieving one of the goals set out by B&NES council.

Sceptics of the Bath proposal have cited the lack of a proper ring road as a potential issue with the LTN scheme, the severity of this problem can be analysed using the generated model. As noted in Section 6.1.1, the average waiting time for journeys did increase by 15 seconds, suggesting there is some amount of congestion at the high-density junctions along the outer boundary of the traffic cells. This is not a significant consequence as the new waiting time is still a similar percentage (~10%) of total trip duration when compared to the baseline data. Some changes could be made at key points to increase junction throughput utilising visualisations such as Figure 5.15 to identify where the largest impact would be made.

A worry when designing the traffic rules to implement the proposal was that vehicles would use the two entrance/exit points of the bottom right traffic cell as a cut through. This would result in increased traffic in the area in opposition to the goals of the scheme. By modelling the projected traffic flow, it has been demonstrated that this is not a potential risk, thus providing more confidence to the validity of the policy design.

Some interior roads near entrance/exits saw slight increases in traffic volume, consideration would be required as to whether this is acceptable for these roads. For example, Midland Bridge Road may be suitable as it is already a thoroughfare, but Vane and Edward Street may not be due to their more residential nature. To alleviate the traffic concentration in these areas, other roads could be selected as entrance/exit points instead of or in addition to the residential streets. Although it is important to note that a traffic cell should have as few entrances/exits as possible to avoid incentivizing through traffic.

Overall, the proposed traffic scheme was judged to be successful in achieving its goals of increasing journey length for cars and decreasing traffic concentration in central areas of the city. A number of possible alterations were also suggested to improve the traffic flow for the projected road structure.

7 EVALUATION

To assess the quality of the produced method a series of evaluation criteria were formulated. This chapter will discuss them and assess whether they were successfully achieved.

7.1 How accurate and reliable are the results for the Bath proposal?

7.1.1 How well does the baseline model reflect the existing traffic in Bath?

When validating the results from the baseline demand there was found to be some major differences between the counted data points use and the traffic volume data modelled. On average there were 64.5 fewer cars for each edge in the simulated data. This difference was mainly caused by three count points which were deemed to have larger variance due to their location close to the edge of the network and therefore near traffic outside of the simulation. Without these three values the average variation was 16.2 vehicles per hour, a much more reasonable difference that shows that the modelled traffic in most areas reflects the input data closely.

All the modelled traffic volumes consisted of values lower than the count point data, underestimating traffic numbers was a decision made so that it can be more certain that the areas where congestion and traffic problems are observed would exist if the scheme was implemented.

Due to the limited nature of count data and inconsistencies in collection times, it is very difficult to be confident that an accurate demand model has been produced. However, the results produced are sufficient to show the application of the proposed method and to make inferences from the projective modelling as the overall volume and general nature (such as having high though traffic) are near accurate.

To improve the accuracy of the baseline demand and therefore allow for more confident and precise modelling of the results of the scheme, an extensive study would need to be made involving the gathering of data at least 10 additional key areas. The study would need to acquire the data at consistent times to ensure a demand element could be made considering all values.

7.1.2 How confident can one be in the extrapolation process from these results?

A benefit of using such controlled structure and input data when implementing the grid-based model in Chapter 4 is that we could clearly see the correct results were produced when the method was implemented without having to worry about intricacies caused by a more complex problem. As the methods used to transform the baseline demand elements and network were identical in the Bath model, we can be confident that the extrapolation is correct. This is further supported by the similarity in the trends of results that were observed in Sections 4.3 and 6.1.

7.2 How useful/actionable were the results produced by the method?

This evaluation point judges the method as a whole, in terms of the quality of results it produces and the ease of use for further application.

7.2.1 Are results clear, showing direct areas where roads could be improved?

Throughout the investigation, both easy to understand visualisations and in-depth statistical conclusions were made. This demonstrates the flexibility of the method in producing results that can be tailored to the desired purpose and therefore are clear wherever they may be used. In particular, the evaluation in Section 6.3 shows how conclusions can be drawn regarding specific roads and with sufficient data supplied to clearly state the projected nature of traffic in each geographical area.

7.2.2 Can the results capture sufficient scale?

By applying the method to both the Bath and grid-based case studies, it was demonstrated that it is effective regardless of the scale with which is applied.

One limitation to the scale of application is the network import and refinement stage which involves a significant amount of manual inspection, with a very large network this would take a significant amount of time and it would be easy for conversion errors to result in inaccurate traffic flow. However, when applied to areas that have already been modelled in SUMO for other purposes this would not be an issue.

7.2.3 How re-creatable are the results?

This evaluation sub-point asks how easily the method could be used by another researcher to investigate LTN schemes in other geographies.

Due to the largely command-line based interface for running the simulation and extracting data, a baseline level of technical ability is required. However, due to the production of specifically tailored scripts during this study, no programming skills would be required. In addition to these scripts a variety of tools were provided for further application, including configurations files, Bash scripts, and visualisation parameters. These mean that any further investigation using this method would require far less complex development than was exhibited in this project.

All tools used were chosen in part for either being Open-Source, such as SUMO and OSM, or otherwise freely accessible like Google Maps and StreetView. This results in it being feasible for any organisation or individual to repeat the method as stated.

As seen in this project, a major limitation to the accuracy of the baseline-demand generation is the availability accurate, consistent traffic data. Many cities have more sophisticated, induction-loop, based counting systems that would allow for much more accurate representation of existing traffic behaviour and subsequently more reliable projections about the effects of an LTN scheme. However, LTN schemes are not always implemented in urban areas with such advanced infrastructure. In smaller areas without count data, it may be difficult to replicate the method as stated and alternate demand generation based on population numbers and other demographic information would need to be used.

7.3 How similar were the projected results of implementing LTN policy to what is seen in the real-world?

Asking this question helps to understand how well the changes shown in the simulation reflect those caused by actual LTN implementations. To assess this, comparisons were made to existing LTM implementation to judge whether the method accurately projects the impact of LTN schemes. In particular, a comparison to the Waltham Forest scheme in London for which claims are made that:

> "The average road within the Village saw a 44.1% reduction in vehicles on the road and a reduction in speed from 21.6mph to 19.5mph" - Chris Proctor, Programme Manager, Enjoy Waltham Forest (Jubb and Mulis)

To demonstrate the success of this evaluation point, values generated in Sections 4.3 and 6.1 were used to calculate the equivalent measures for the simulated schemes which were then compared to the Waltham Forest figures, as shown in Table 7.1. The values used for reduction in vehicle numbers were chosen by selecting routes near the centre of traffic cells. They notably do not account for any reduction in demand that may be caused by the shift to alternate transport methods that is present in the Waltham Forest value and was projected for the Bath model in Section 6.2. This choice was made to provide a more conservative estimate for traffic reduction and therefore increase the confidence with which conclusions can be made.

	Simple Model	Bath Model	Waltham Forest
Vehicle Reduction	44.8%	48%	44.1%
Speed Reduction	3.4 mph	0.87 mph	2.1 mph

Table 7.1 - Modelled and actual effects of LTN schemes

Although some differences would be expected due to the varying scales and implementations of each LTN scheme, the results from the predictive model still closely match the data gathered from a real-world implementation.

8 CONCLUSIONS

Building on the academic foundations of traffic modelling and SUMO usage, a method has been successfully developed for analysing and assessing proposed LTN schemes. This method generates detailed representations of projected traffic behaviour, such as that shown in Figure 5.15, and can be used to make actionable inferences to guide policy proposals.

Using both novel and established techniques and tools, a process for formulating a baseline model representing existing traffic behaviour and volume was outlined. The core innovation that allowed for predictive modelling was the development of the Python script, *tripsFromRoutes.py,* that generates a representation of traffic demand that is not tied to the intricacies of a traffic system and can be reapplied to an altered network. The creation of such altered networks was demonstrated, describing how principles of LTN policy and their aims were applied. By applying the proposed method to a simple artificial scenario, a foundational understanding of the effects was formed. This allows for more confidence in inferences made from emergent behaviour when the method is applied to more complex scenarios. Furthermore, the validity of the method was shown by its ability to produce results that align with what is observed in real-life applications of LTN policy.

In parallel with the account of method development, the application of the model to two separate scenarios is described. This includes depictions of an extensive set of tools that have been developed for the use of the method including Python and Bash scripts that automate much of the process. These examples use cases provide a step-by-step depiction of the method, including the experimentation and analysis of results to garner predictive information. It was shown how city planners may utilise these results, applying the principles of LTN design, to inform policy making and assess potential proposals. By providing this facility, the method could allow governing bodies to evaluate potential LTN schemes without the need to implement costly trials. By showing the application of the method using a current proposal from B&NES council for a traffic-cell scheme in Bath city centre and utilising it to suggest specific changes, the relevance of the project goals is clearly demonstrated, and they are shown to be achieved.

When applying the method to the Bath proposal, generating a suitable baseline demand description was challenging due to the limited and inconsistent form of the data available. To account for this, heuristics were used to randomly generate the traffic outside the areas with available data. Combining the two demand elements resulted in a trade-off between accurately reflecting the real-life count data and forming feasible demand for the rest of the network. Other methods could be considered for combining the two demand elements without such divergence from the recorded data, perhaps by using SUMO's *duaiterate* which allows the modelling of non-shortest path routes. This limitation did not have a significant impact on the success of the project as the demand element modelling is modular to the overall predictive method. Sufficiently accurate demand was generated for the purposes of modelling the effects of the proposed scheme and inferring policy decisions from the results. Due to the inherent modularity, difficulty of modelling the demand element does not have a bearing on the quality of method or the accuracy of future result. In areas where data is more available or a SUMO demand representation already exists, the method can be implemented very easily.

Care was taken to automate much of the process and tools are provided to make its application straightforward for those with a base level of Computer Science knowledge. However, for the stated use by councils for modelling potential schemes additional guidance may be required as it is unlikely that these organisations already have the required familiarity with mechanisms like command-line based interfaces.

This research provides a method for the predictive simulation and analysis of LTN schemes as well as a set of tools and guidelines for its implementation. It can thus be used to help reach the goal of lowering emissions and improving neighbourhoods by greatly decreasing the cost and time needed to analyse and assess potential proposals.

9 FUTURE WORK

9.1 Further Experimentation and Analysis involving Bath proposal

The approach demonstrated in Section 6.1 provides a thorough analysis of the impacts of the proposed traffic cell policy on general traffic flow and distribution. However, by producing a fully parameterised model of the projected traffic system, this method provides a platform for many alternate avenues of investigation like that shown in Section 6.2. Some examples are detailed in this section.

In this study, a single interpretation of the Bath city centre proposal was investigated. However, being in its early stages, there are different candidate road structures that could fit the goals and description provided by B&NES council. A simple further application of the method would be to construct these plans in *netedit* and carry out the method as described. This would be a simple process due to the straightforward and largely automated process. The results from each experiment could each be analysed and compared using the SUMO visualisation tools demonstrated in this study. This would allow planners to discuss the benefits and drawbacks of each potential implementation and may also result in less instability at the start of the process as slight alterations will not need to be tested on live traffic.

As mentioned in Section 4.3.1, to improve the detail of journey base analysis, routes could be classified based on their nature in the baseline model. Distinguishing routes that start and end outside the network from routes that require access to a traffic cell would allow for clearer results to be drawn. For example, ideally the changes made by LTN policy should have little to no impact outside of the area of interest, i.e., that that only uses the ring road, categorisation would confirmation of this as currently the data can only be analysed as a whole. Additionally, the most important journeys to impact are those that would represent 'through traffic', when calculating values based on overall averages, the information relating to these vehicles is diluted by all other agents in the simulation. To implement these changes, specific vehicle ids could be provided to SUMO based on how the route was generated by *randomTrips.py*, this would maintain the ability to analyse the simulation as whole whilst allowing for more specific route analysis.

A key element in the success of an LTN policy is the resultant shift to active and public transport methods reducing the total number of vehicles in use. Information about how the policy will affect these forms of travel would be useful for planning specialised infrastructure such as those proposed by B&NES as part of the traffic cell policy. These developments are often implemented in parallel to the LTN road changes (as in the Bath proposal), so planners do not have the opportunity to react to the impacts of the scheme when making decisions on placement and capacity. Adding these modes of travel to the simulation models would facilitate decision making and reduce reliance on rough projections. This change would also allow planners to explain the benefits of these modes of transport more easily by quoting journey length and time comparisons, thus persuading a faster switch and achieving the policy goal more quickly.

SUMO provides features for distinguishing agents like buses and cyclists as well as separate routing models for pedestrians. Expanding the simulation to include these forms of travel would require alterations to nearly every stage of the process. Count data would be required to model the demand for these agents. This data is available from the Department (Department of Transport, 2020b) of Transport alongside the data used for this investigation . This demand would then need to be modelled using a network adapted to contain any mode specific edges and rules such as pedestrian bridges and bike lanes. Once the simulation is produced, the suite of analysis tools provided by SUMO and used throughout this investigation could be applied in a similar fashion.

9.2 Further method validation

As discussed in Sections 3.3 and 3.4, the two scenarios used to develop the method were chosen for their clarity and to justify the relevance of the project aims. To further increase confidence in the validity of the method for producing accurate projections of the effects of LTN policy, a further case study could be done using the City of Ghent. Creating a model applying the method to historical data from before the Circulation Plan was implemented in April 2017 would allow comparison to current traffic data. As the policy has actually been implemented in Ghent, a valid predictive model would closely follow the current traffic behaviour. This could be used to provide a clearer verdict on the accuracy and validity of the method.

Due to the limited counted traffic data available, validation of the baseline demand element was performed by comparing results to the input data. Whilst this is useful for evaluating the ability of the method to match provided input data, it does not account for potential overfitting to the limited data. This would result in the result data being accurate to the data points used for input but being significantly divergent from the actual traffic behaviour outside of this limited data set. To ensure this is not the case, a comparison could be made to another form of traffic data, specifically Google Maps' congestion metric. Using a set of tools developed by the University of Bath's Nick Mullen data could be gathered about congestion during the relevant time period. This data could then be mapped and compared to a congestion analysis of the base demand element in order to improve confidence in the validity of the traffic volume and pattern generated.

9.3 Improvement of the technical elements

Currently the slowest part of carrying out the method is the running of the *tripsFromRoutes.py* script due to its requirement to iterate through multiple long XML files. Optimizations to the run-time of this script would allow for changes in the baseline demand to be realised in the LTN model without friction, thus providing a better environment for experimentation. Normally this is not an issue as the baseline demand is constructed then converted to follow the altered network traffic rules only once. However, to promote adaptability of the method, parallelisation could be implemented to reduce conversion times by allowing values to be generated concurrently. This is a tool utilised by other SUMO scripts which, as they work with similar data, could be used as a guide for this development. Parallelisation is suited to this problem as each individual route can be calculated independent to other values, thus avoiding any major issues with race conditions and other parallel specific problems caused by shared state between threads of execution.

An additional step that could be taken to improve the sophistication if this method is adding support for simulations of periods longer than a single hour. This would allow for more precise information to be gather about the timings of congestions and demand limits via modelling the traffic in an area over the course of a day or week. SUMO supports this feature natively and many of the processes would need only minor alterations such as adding options to define simulation intervals. The most significant change needing to be made would be altering the scripts developed throughout the investigation to support interval base files and parameters. Many SUMO provided scripts already provide these features so could therefore be used as a guide for this process.

BIBLIOGRAPHY

Aldred, R., Croft, J. and Goodman, A., 2019. Impacts of an active travel intervention with a cycling focus in a suburban context: One-year findings from an evaluation of London's in-progress mini-Hollands programme. *Transportation Research Part A: Policy and Practice,* 123, pp. 147-169.

Aldred, R. and Goodman, A., 2020. Low Traffic Neighbourhoods, Car Use, and Active Travel: Evidence from the People and Places Survey of Outer London Active Travel Interventions. *Transport Findings.*

Ames, C., 2020. Wandsworth suspends LTN trials one month in. *LocalGov*. Available from: https://www.localgov.co.uk/Wandsworth-suspends-LTN-trials-one-month-in/51090.

Bath and North East Somerset Council, 2022. *Journey to Net Zero: Reducing the environmental impact of transport in Bath* [Online]. Available from: https://beta.bathnes.gov.uk/journey-net-zero/improved-places-future-projects#Traffic [Accessed 04/22].

Ben-Akiva, M., De Palma, A. and Kanaroglou, P., 1986. Dynamic model of peak period traffic congestion with elastic arrival rates. *Transportation Science,* 20(3), pp. 164-181.

Castillo-Manzano, J.I., Lopez-Valpuesta, L. and Asencio-Flores, J.P., 2014. Extending pedestrianization processes outside the old city center; conflict and benefits in the case of the city of Seville. *Habitat International,* 44, pp. 194-201.

Chen, B. and Cheng, H.H., 2010. A Review of the Applications of Agent Technology in Traffic and Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems,* 11(2), pp. 485-497.

City of Ghent, *The Circulation Plan* [Online]. Available from: https://stad.gent/en/mobility-ghent/circulation-plan.

Codeca, L., Frank, R., Faye, S. and Engel, T., 2017. Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation. *IEEE Intelligent Transportation Systems Magazine,* 9(2), pp. 52-63.

Department of Transport, 2020a. Active travel fund: final allocations.

Department of Transport, 2020b. *Local authority Bath and North East Somerset* [Online]. Available from: https://roadtraffic.dft.gov.uk/local-authorities/115.

Dias, J.C., Abreu, P.H., Silva, D.C., Fernades, G., Machado, P. and Leitão, A., 2013. Preparing data for urban traffic simulation using SUMO. *1st SUMO User Conference*. p. 25.

Donovan, T., 2022. Will Low Traffic Neighbourhoods drive London's elections? *BBC News*. Available from: https://www.bbc.co.uk/news/uk-england-london-61191213.

Flötteröd, G. and Kickhöfer, B., 2016. Choice Models in MATSim. In: A.e.a.e. Horni, ed. *Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press.

Fournier, N., 2021. Hybrid pedestrian and transit priority zoning policies in an urban street network: Evaluating network traffic flow impacts with analytical approximation. *Transportation Research Part A: Policy and Practice,* 152, pp. 254-274.

German Aerospace Center (DLR), 2022. *SUMO User Documentation* [Online]. Available from: https://sumo.dlr.de/docs/index.html [Accessed 12/05].

Gipps, P.G., 1981. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological,* 15(2), pp. 105-111.

Goodman, A., Urban, S. and Aldred, R., 2020. The impact of Low Traffic Neighbourhoods and other active travel interventions on vehicle ownership: findings from the Outer London mini-Holland programme. *Findings*.

Haklay, M. and Weber, P., 2008. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing,* 7(4), pp. 12-18.

Holland, C., 2014. Question and answers for Mini Holland programme.

Hoogendoorn, R.G., Hoogendoorn, S.P., Brookhuis, K.A. and Daamen, W., 2011. Adaptation longitudinal driving behavior, mental workload, and psycho-spacing models in fog. *Transportation research record,* 2249(1), pp. 20-28.

Jubb, C. and Mulis, R., A *Guide to Low Traffic Neighbourhoods*.

Judge, C., 2015. *The Experiment of American Pedestrian Malls: Trends Analysis, Necessary Indicators for Success and Recommendations for Fresno's Fulton Mall.*

Kerner, B.S., 1999. The physics of traffic. *Physics World,* 12(8), p. 25.

Kerner, B.S., Klenov, S.L. and Wolf, D.E., 2002. Cellular automata approach to three-phase traffic theory. *Journal of Physics A: Mathematical and General,* 35(47), pp. 9971-10013.

Krajzewicz, D., 2010. Traffic Simulation with SUMO – Simulation of Urban Mobility. In: J. Barceló, ed. *Fundamentals of Traffic Simulation.* New York, NY: Springer New York, pp. 269-293.

Krajzewicz, D., Brockfeld, E., Mikat, J., Ringel, J., Feld, C., Tuchscheerer, W., Wagner, P. and Woesler, R., 2005. *Simulation of modern Traffic Lights Control Systems using the open source Traffic Simulation SUMO.*

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

Larsson, U., 1972. An urban traffic simulation model. *SIGSIM Simul. Dig.,* 4(1), pp. 76–81.

Laverty, A.A., Aldred, R. and Goodman, A., 2021. The Impact of Introducing Low Traffic Neighbourhoods on Road Traffic Injuries. *Findings.*

Leutzbach, W., 1972. *Einführung in die Theorie des Verkehrsflusses.* 1 ed.

MATSim Community, 2022. *MATSim Muti-Agent Transport Simulation* [Online]. Available from: https://www.matsim.org/about-matsim [Accessed 25/04].

Open Street Maps, *Open Street Maps export page* [Online]. Available from: https://www.openstreetmap.org/export.

Pipes, L.A., 1953. An operational analysis of traffic dynamics. *Journal of applied physics,* 24(3), pp. 274-281.

PTV Group, *PTV Vissim* [Online]. Available from: https://www.ptvgroup.com/en/solutions/products/ptv-vissim/.

Robertson, K.A., 1991. Pedestrian streets in Sweden's city centres. *Cities,* 8(4), pp. 301-314.

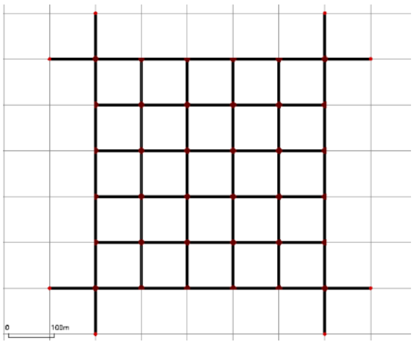
Sumner, S., 2022. Radical rethink of roads could see Bath split into four cells. *SomersetLive*. Available from: https://www.somersetlive.co.uk/news/radical-rethink-roads-could-see-6495045.

Tang, T., Shi, W., Shang, H. and Wang, Y., 2014. A new car-following model with consideration of inter-vehicle communication. *Nonlinear Dynamics,* 76(4), pp. 2017-2023.

APPENDICES

Appendix A

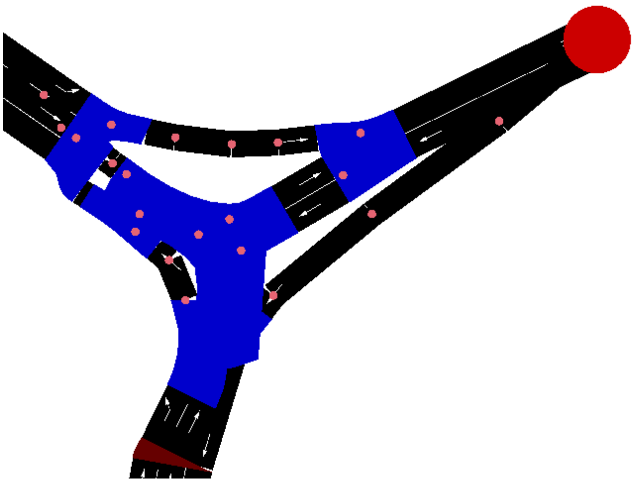
Early prototype grid with fewer fringe edges:



This was decided against as it resulted in the fringe traffic being too concentrated for the desired demand distribution.

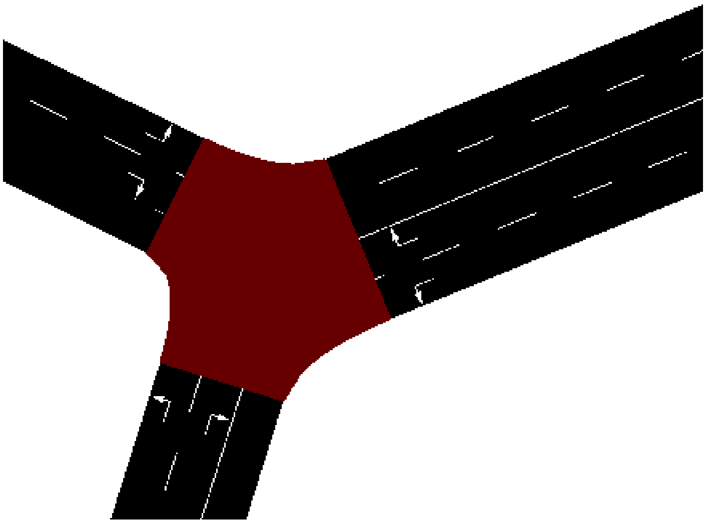
Appendix B

The junction as imported by *netconvert:*



The multiple sections highlighted in blue are all separate junctions that would results in vehicles becoming stuck within the centre of the cluster.

The junction after merging:



Although not geographically identical, with the correct alterations to lane rules the merged junction is much more accurate for traffic modelling.

Appendix C

Simulation of simple model baseline with verbose output:

```
Performance:
 Duration: 2.54s
 Real time factor: 1483.68
 UPS: 158003.145891
Vehicles:
 Inserted: 4925
 Running: 0
 Waiting: 0
```

Simulation of simple model LTN with verbose output:

```
Performance:
 Duration: 1.71s
 Real time factor: 2143.52
 UPS: 134714.702450
Vehicles:
 Inserted: 4925
 Running: 0
 Waiting: 0
```

Simulation of simple model baseline with statistical output:

```
Performance:
 Duration: 1.74s
 Real time factor: 2117.58
 UPS: 133084.149856
Vehicles:
 Inserted: 4925
 Running: 0
 Waiting: 0
Statistics (avg):
 RouteLength: 460.45
 Speed: 9.76
 Duration: 46.88
 WaitingTime: 1.22
 TimeLoss: 12.41
 DepartDelay: 1.03
```

Simulation of simple model LTN with statistical output:

```
Performance:
 Duration: 2.36s
 Real time factor: 1597.38
 UPS: 170110.922947
Vehicles:
 Inserted: 4925
 Running: 0
 Waiting: 0
Statistics (avg):
 RouteLength: 569.43
 Speed: 8.25
 Duration: 81.58
 WaitingTime: 15.66
 TimeLoss: 38.45
 DepartDelay: 4.91
```

Simulation of Bath model baseline with verbose output:

```
Performance:
 Duration: 12.04s
 Real time factor: 351.49
 UPS: 101839.574857
Vehicles:
 Inserted: 5058
 Running: 0
 Waiting: 0
```

Simulation of Bath model LTN with verbose output:

```
Performance:
 Duration: 7.97s
 Real time factor: 487.832
 UPS: 98341.570497
Vehicles:
 Inserted: 5058
 Running: 0
 Waiting: 0
```

Simulation of Bath model baseline with statistical output:

```
Performance:
 Duration: 14.67s
 Real time factor: 265.081
 UPS: 53437.325336
Vehicles:
 Inserted: 5058
 Running: 0
 Waiting: 0
Statistics (avg):
 RouteLength: 1276.81
 Speed: 8.47
 Duration: 155.00
 WaitingTime: 15.05
 TimeLoss: 47.01
 DepartDelay: 1.65
```

Simulation of Bath model LTN with statistical output:

```
 Real time factor: 190.925
 UPS: 55317.937847
Vehicles:
 Inserted: 5058
 Running: 0
 Waiting: 0
Statistics (avg):
 RouteLength: 1870.25
 Speed: 8.08
 Duration: 242.48
 WaitingTime: 26.17
 TimeLoss: 89.22
 DepartDelay: 6.02
```

Appendix D

All code, configurations files and input/output data can be accessed via the public GitHub repository: https://github.com/samMolyneux/LTNsimulation in the file 'Simulation'.

Note that all scripts in the folder 'tools' are SUMO provided and have been copied in for ease of use.

Python scripts written or altered for the purposes of this project are located in the folder 'mytools'.