Ashesi University College

Early Stages of Traffic Modeling and Simulation

Nii Boye Mensah Adjei

2013

Applied Project

Ashesi University College

Early Stages of Traffic Modeling and Simulation

By

Nii Boye Mensah Adjei

Dissertation Submitted to the Department of Computer Science,

Ashesi University College

In partial fulfillment of Science degree in Computer Science

April 2013

Declaration

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate’s Signature:...

Candidate’s Name:...

Date:...............................

I hereby declare that the preparation and presentation of the dissertation were supervised in accordance with the guidelines on supervision of dissertation laid down by Ashesi University College.

Supervisor’s Signature...

Supervisor’s Name:...

Date:..

Acknowledgements

I will like to thank my supervisor, Mr Kwadwo Gyamfi Osafo-Maafo for his help both academically and morally. I would also like to extend my gratitude to Dr Nathan Amanquah for the introduction into this area of study and his inintial contributions.

I will like to thank my family and all my friends who supported me morally and psychologically.

Finally, I will like to thank God for his love and mercies shown. Thank you.

Abstract

In Ghana, most junctions are managed by semi - Intelligent Traffic Systems with a fixed-time traffic control. This has become possible due to the advancement of the transport management. Despite these technologies, traffic control is still an issue as policemen sometimes direct traffic even when there are working traffic lights. This is so because the systems have not been adjusted properly to help with good traffic flow. This paper will not focus on exploring how to measure these intelligent systems. It will focus on how Roundabouts since they are often used at large intersections in Ghana. With no intelligent traffic system, the traffic wardens manage traffic. Performance will be based on comparison with existing models. Through the results, this dissertation will outline the benefits of using such a system and the chain reaction it will cause.

Declaration iii

Acknowledgements iv

Abstract v

CHAPTER 1: INTRODUCTION vi

1.1 Background 3

1.2 Objectives 5

1.3 Outline of Dissertation 5

CHAPTER 2: LITERATURE REVIEW 6

2.1 Queueing Theory 6

2.1.1 Little's Theorem 6

2.2 Simulation 7

2.3 Traffic Simulation 8

2.4 Microscopic Models 9

2.4.1 Free Traffic State 9

2.4.2 Following State 9

2.4.3 Braking State 10

2.4.4 Gipp's Model 10

2.4.5 Intelligent Driver Model 10

2.5 Macroscopic Models 13

2.5.1 Lighthill Whitman and Richards (LWR) Model 13

2.5.2 Aw and Rascle Model (AR) Model 13

2.5.3 Zhang Model 14

2.6 Traffic in Developed and Developing Countries 14

CHAPTER 3: METHODOLOGY AND LIMITATIONS 16

3.1 Development of Custom Simulator 17

3.1.1 Requirement Specifications 17

3.1.2 Tools 18

3.1.3 Design 18

3.1.4 Implementation 20

3.2 Limitations 27

3.2.1 Data Collection 28

3.2.2 Technical Expertise 28

3.2.3 Simulation Errors 29

CHAPTER 4: SIMULATION EVALUATION 30

4.1 Results and Analysis 30

CHAPTER 5: CONCLUSION 32

5.1 Further Work 32

References 34

APPENDICES 36

Appendix A: Installation of NS3 for Linux (Fedora) 36

Appendix B: Installation of NS2 for Windows 37

LIST OF FIGURES

Fig 2.1: Intelligent Driver Model Equation

Fig 2.2: Table of Parameters found in the Intelligent Driver Model

Fig 2.3: Free-Traffic Equation derived from the Intelligent Driver Model

Fig 2.4: Braking Equation derived from the Intelligent Driver Model

Fig 2.5: Lighthill Whitman and Richards Model

Fig 2.6: Aw and Rascle Model

Fig 2.7: Zhang's Model

Fig 3.1: Flow Chart of Main Implementation

Fig 3.2: Class Dependencies

Fig 3.3: Representation of Custom Code Simulated

Fig 3.4 Free-Traffic Equation derived from the Intelligent Driver Model

Fig 4.1 Free-Traffic Equation derived from the Intelligent Driver Model

CHAPTER 1: INTRODUCTION

It is every developing country's wish for its economy to grow. To make it possible, one of the most important factors that push these aspirations forward is a good transportation system. When transport systems are efficient, they provide economic and social opportunities and benefits that result in positive multipliers effects such as better accessibility to markets, employment and additional investments. People need to move from one place to another to make business flourish. At the macroeconomic level, transportation accounts for 6% to 12% of the GDP in developed countries. Transportation creates the necessary relationship between the producer and its customers (Rodrigue and Notteboom, 2013).

In the early times, these journeys were made by foot or riding on an animal, depending on one's geographical position. Modern transportation mostly relies on cars, trains, ships and planes. These modern inventions do not rely on themselves though, because cars need roads and trains need railway lines. The infrastructure needed for some of these vehicles to be used is expensive to construct. For example, Ghana is to work on a project to upgrade the urban traffic management with the aim of installing 182 intelligent traffic signals that will be coordinated in Accra. This need arises everywhere as population and businesses also grow. Due to the high cost of transportation infrastructure, it is important traffic management systems are developed to help manage the traffic flow. These systems are in use by many industrialized countries to help combat their congestion. A perfect example is the Eldor Contracting Corporation. Saab is also another example. They have developed a radar sensor that supports optimizing and managing traffic flows. This radar sensor is called SIRS-200 ITS.

Lately, a lot of roads have been constructed and upgraded to help with the easy flow of traffic. A very good example is the George Bush Junior Highway. However, sometimes the traffic lights and traffic wardens who are supposed to help regulate the traffic appear to rather cause more traffic. This often happens during the rush hours, which are from 6am to 9:00am and 3:00 to 6:00pm averagely. This results in workers getting to work late a lot of the times. Ghanaians always blame their unpunctuality on the traffic jams they experience. This inevitably disrupts production of work that can be done in a day. With good investment and research into providing better traffic management systems, the average output of work put in a day will increase.

With a good transportation system, some problems and challenges can be managed. These problems include traffic jams and pollution, both noise and air(Sampson, 2006). When it comes to the issue of the traffic jams, which can be used by the traffic light systems or even the drivers themselves, a chain reaction is created. Traffic jams are situations where the traffic flow from one point to another is halted due to increase cars or bad timing of traffic lights (look for accurate definition). It leads to drivers sitting in a queue for long hours, which eventually cause frustration. This frustration then distorts their ability to be patient thus becoming impatient

(Hennessy and Wiesenthal, 1999). When impatience sets in, the desire to leave the traffic jam introduces reckless driving, which is harmful because it can intensify the jam. To dissolve these traffic jams, traffic wardens and sometimes even ordinary civilians come out to direct the flow of the traffic. It has sometimes been the best solution.

1.1 Background

Traffic models exist and are discussed but it is important to continually evaluate and explore ways to verify and improve existing models.. In Ghana, the traffic model in use at roundabouts instructs traffic entering the roundabout to yield to traffic already in the roundabout. However, traffic wardens may manually control traffic to either use the rule of yielding to traffic or override the rule. Sometimes, they let traffic already in the roundabout yield to traffic entering the roundabout. While traffic wardens are very helpful in some scenarios, it is important to investigate their impact in such situations. For example, their actions may result in queues being formed in the roundabout. These queues eventually lead to frustration, which always sets in after long hours of staying in the jam. This frustration just leads to the recklessness of a lot of drivers, which introduces accidents. These long hours also pose a threat to people's health, as there is noise pollution from the sound of the vehicles and also air pollution from their exhaust.

In the summer of 2011, I took a job that sometimes required a commute through the Kwame Nkrumah Circle. From Monday to Friday, it was constant that I used the Kwame Nkrumah Circle. The human traffic was fierce and very frustrating but was nothing compared to the vehicular traffic. The mornings were smooth but the late afternoons and early evenings were just unbearable. I sometimes joined a friend when coming home or picked the public transport. I honestly do not know which was better, probably my friend's because he most at times had the air conditioner on.

In the car, I always kept wondering why there was so much traffic. It was so bad that it extended to Busy Internet and sometimes further, approximately 400meters. When this happens, I most at times get down from the car (if it is public transport) and walked to the lorry station. On my way, I normally glance at the circle in amazement and ask why there was so much traffic. I would traffic wardens and policemen trying to manage the flow; sometimes it just got worse instead of better.

The funny thing is, it had never occurred to me that about a three-minute drive from my home, a roundabout that could face the same tyranny of the traffic jam existed. However, it never had. The model of yielding to traffic already in the roundabout was being applied. It came to my attention after my lecturer, Dr. Nathan Amanquah, and I had a discussion on finding an alternative to the way the traffic wardens and policemen handle the traffic at the Kwame Nkrumah Circle. After careful observation, we realized that they are a contributing factor to the traffic jams.

1.2 Objectives

The main focus of this dissertation is to provide an alternative way of managing the traffic flow to the traffic wardens at the Kwame Nkrumah Circle. The application under development will illustrate firstly, the current model or situation being executed at the Kwame Nkrumah Circle. This is will be done to expose the flaws within that management system. After, the application will then simulate the alternative model being proposed to illustrate the how efficient it is. It may not be flawless but it will provide an insight on how things should be done at the circle. To execute the model into a real world situation, it will be compiled into a simulator to help with the visualization.

1.3 Outline of Dissertation

The next chapter of this dissertation talks about traffic models that have been developed. It first introduces the queuing theory, which is the basic principle of the queuing theory. It also explains the necessity of simulations. Chapter three explains the methodology used to conduct this project. It also talks about the limitations faced while implementing this project as well as the precautions to be taken when using it. Chapter four covers the results of the simulation and its analysis. Chapter five concludes the project. It also discusses possible improvements to this dissertation.

CHAPTER 2: LITERATURE REVIEW

This dissertation will provide literature mainly based on how traffic is managed in foreign countries. With the level of detail on the road serving as the basis, researchers are able provide the best ways to manage traffic flow. Nowadays, computers help visualize all possible outcomes with existing models. These outcomes help provide solutions or alternatives to these existing models. This may require creating a new model or simply modifying an existing one. To do this, one needs to understand the concept of the queuing theory with Little Laws serving as its basic principle.

2.1 Queueing Theory

Congestion is found almost everywhere there needs to be movement from one place to another. It is present in every system that is involved in transportation. The queuing theorem provides tools to help analyze and reduce this congestion because it reduces the service time of the system; usually resulting in delays. Queuing systems requires one to know about Little's Theorem.

2.1.1 Little's Theorem

This theorem states that the average number of entities can be determined by the equation:

$$N = \lambda T$$

The lambda (λ) here represents the average arrival rate. '**T**' represents the average service time of an entity, which means averagely, how long it takes an entity to go through the system. With any changes in either λ or '**T**', the average number of entities will also change corresponding to their change (Little and Graves, 2008).

2.2 Simulation

Simulation represents an imitation of an operation of real-world system overtime. Before a simulation is done, it requires the creation of a model. The model captures key characteristics of a system. The operation of the system is the simulation.

Simulation has been used in many contexts to try and capture important details of a system. These details help with decision-making as provides insight on the system. The main aim is performance-optimization. Simulations are used in many contexts such as training and education. This dissertation will focus its use on traffic. There are a number. These were the ones I explored:

- MITSIMLab: a simulation-based lab that was developed for evaluation of impacts of alternative traffic management system designs at the operational level. It also helps in the refinement of subsequent designs. It three main modules from which various components are organized. These are – Microscopic Traffic Simulator (MITSIM), Traffic Management Simulator (TMS) and a

Graphical User Interface (GUI). It was developed at the MIT Intelligent Transportation Systems (ITS) program (Ben-Akiva, Koutsopolous and Yang, 2002).

- Network Simulator (NS) 2 and 3: It is a discrete event simulator that is targeted at research on networks. It began as a variant of the REAL Network Simulator in 1989 and has undergone some evolution over the years. It provides substantial support for simulation o TCP, routing and multicast protocols over wired and wireless networks. It comes with an animator that helps with visualizing the simulation being done (McCanne, Floyd and Fall, 1997).

2.3 Traffic Simulation

To create a model to be used as a traffic management system, it is necessary to understand the queuing principles. Little's Law serves as the basis of the creating a queue, which represents the entities in the system. This representation nowadays is simulated using computers. In the past, mathematical simulations were used. This new era however embraces three main approaches – microscopic, mesoscopic and macroscopic models. Each classification has a level of detail with which it comes. The microscopic is the most detailed and the macroscopic is the least detailed. The mesoscopic provides less detail as compared to the microscopic but more when it comes to the macroscopic. For this dissertation, I will focus more on the microscopic and macroscopic models.

2.4 Microscopic Models

These models are also referred to as car-following models because a lead vehicle usually dictates them. These models are simulated to capture the motion of each vehicle in the system. It also takes into consideration, how the driver reacts to the situation on the road. Acceleration, position and the velocity functions of the vehicles are involved. With the aim of trying to capture how a human behaves with respect to the traffic situation, these models have been put into different states. Each state describes a traffic situation.

2.4.1 Free Traffic State

This driving state depicts a low vehicular density without the presence of a lead vehicle. Thus, velocity, acceleration and the position cannot be influenced. Individual vehicles can accelerate to their desired velocity (Miller, 2011).

2.4.2 Following State

With a medium to high vehicle density, this state represents the everyday traffic. In this state, each vehicle reacts to how the lead vehicle reacts. The following vehicle tries as much to keep a minimum and maximum time or vehicle gap from the leading vehicle.

2.4.3 Braking State

This state kicks in when the leading vehicle starts to slow down or come to a stop. Automatically, the following vehicle starts to responds these actions by applying some level of brake so as not to collide with the leading vehicle. It is sometimes referred to as an Emergency Response.

2.4.4 Gipp's Model

The most basic of the microscopic models is the Gipp's Model. This models uses the driving states to model traffic flow. Here, acceleration of the following car depends on the speed and position of the leading car. This was introduced in 1970.

2.4.5 Intelligent Driver Model

With the aspiration to improve upon the older models, a newer model was created and published in 2000. It is known as the Intelligent Driver Model (IDM). This model introduces an acceleration strategy with a braking strategy, which covers the three driving states. Below is its equation:

$$\dot{v}_{\mathrm{IDM}}(s, v, \Delta v) = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*(v, \Delta v)}{s}\right)^2\right]$$

$$s^*(v, \Delta v) = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}$$

Fig 2.1: Intelligent Driver Model Equation

The s^* term below the main function is an expansion of s^* in the numerator of the main function. It is an acceleration function. The following is a table of the variables and what they represent.

Parameter	**Representation**
s	Vehicle gap
v	Velocity
Δv	Velocity Difference
v_0	Desired Velocity
T	Safe Time Headway (in seconds)
a	Maximum Acceleration
b	Maximum Deceleration
δ	Acceleration Component
s_0	Jam Distance
s_1*(not shown but exists)*	Jam Distance
$l=1/p_{max}$	Vehicle length

Fig 2.2: Table of Parameters found in the Intelligent Driver Model

The equation can be broken down into the free traffic state equation and the braking equation. Below is the free traffic state equation:

$$\dot{v}_{\text{free}}(v) = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta}\right]$$

Fig 2.3: Free-Traffic Equation derived from the Intelligent Driver Model

The acceleration v_{free} (v)->0, as the v (velocity) - > v_0 (desired velocity). As the desired velocity of a car is being reached, acceleration of the car reduces.

The braking or interaction equation is responsible for the braking and following driving states. It is given as:

$$\dot{v}_{\text{brake}}(s, v, \Delta v) = -a\left(\frac{s^*}{s}\right)^2$$
$$s^*(v, \Delta v) = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}$$

Fig 2.4: Braking Equation derived from the Intelligent Driver Model

The term vT attempts to maintain a specific time gap/safe time headway (T) from the car being followed. The $v\Delta v/2\sqrt{ab}$ attempts to brake within the limits of b but will exceed its value when there is the danger of collision, thus preventing a hit. The vT term dictates in a normal driving situation while the other kicks in when approaching an object at a high speed rate (Treiber, Hennecke and Helbing, 2000).

2.5 Macroscopic Models

Developed in the 1950's by scientists, macroscopic models emulate the traits of the fluid mechanics. They are intended to capture the behavior of a system not a specific vehicle. These models are concerned with how drivers react to changes in their immediate environment. They make use of aggregate variables like velocity, density and flow to describe traffic. They are categorized into partial differential equations.

2.5.1 Lighthill Whitman and Richards (LWR) Model

Published in the 1950's, this model is viewed as a scalar, time-varying, non-linear, hyperbolic partial differential equation. It is has number of basic assumptions. One is that velocity depends traffic density(Miller, 2011). The model is given by:

$$\rho_t(x,t) + (\rho(x,t)V(\rho(x,t)))_x = 0$$

Fig 2.5: Lighthill Whitman and Richards Model

2.5.2 Aw and Rascle Model (AR) Model

This model attempts to move away from the fluid-flow standard found in the LWR model (Miller, 2011). The model is given by:

$$\rho_t + (\rho v)_x = 0$$
$$(v + P(\rho))_t + v(v + P(\rho))_x = 0$$

Fig 2.6: Aw and Rascle Model

2.5.3 Zhang Model

This model completely excludes the use of fluid-flow model and implements a second equation derived from the microscopic model. Thus, creating a macro-micro link (Miller, 2011). The model is given by:

$$\rho_t + (\rho v)_x = 0$$
$$v_t + v v_x + \rho V\prime(\rho) v_x = 0$$

Fig 2.7: Zhang's Model

2.6 Traffic in Developed and Developing Countries

Countries that have reached some level of development always strive to improve their status quo. They do this by improving sectors that impact most on their economy. Transportation is one contributing sector. How the transportation is managed is very essential. Systems or road infrastructure may be built to mange the flow of traffic as that would determine how quick things can move from one destination to another. In both developed and developing countries, these solutions are highly valued.

In the city of Buffalo, USA, a project with a timeline was set up. It started from the fall in 2005 to the winter in 2007. The aim of the project was to synchronize the traffic lights within the city so as to provide an efficient traffic management system. Recording the patterns from the areas of interest did this. The signals from the patterns were then connected to the IQ Central Signal software, which provides a super traffic management control. It provides a centralized management for a mix of the signals coming in from the areas of interest.

In Lagos, Nigeria an entity was established to handle the traffic in the state. This entity is known the Lagos State Traffic Management Authority (LASTMA). It was created about a decade ago. It is currently the major solution to the their road traffic issue in the state.

Every country whether developed or developing seeks to manage its transportation. This project is important since the aim is to study how to measure how these transportation systems work.

CHAPTER 3: METHODOLOGY AND LIMITATIONS

This dissertation will look at three levels of simulation. With the fact that a roundabout is a combination of lanes, each simulation will obviously provide a different feedback. The simulations are as follows:

- **Single Car in Single Lane:** This simulation basically takes in a car with a constant speed, makes it enter the roundabout and goes through it to reach the exit. All cars have the same entry and exit lanes.
- **Multiple Cars in Single Lane:** This simulation takes in multiple cars with their constant speeds. A car then completes the cycle of going by through the roundabout after which the next car enters.

- **Semi-Advanced Multiple Cars in Single Lane:** This simulation is similar to that of the Multiple Cars in a Single Lane. The only difference is that this time when a car enters starts the cycle, it quickly followed by the next car, and then the next. This is done until all the cars that have been created are exhausted.
- **Advanced Multiple Cars in Single Lane:** In this simulation, the cars that are created have entry times. Each car has an entry and an exit lane. All cars have a constant speed.

The best solution was to use an already existing simulator for these levels of simulation. Implementing the real thing was just not feasible. MITSIMLab, NS2 and NS3 were all explored. MITSIMLab proved to be the most complex. It was very difficult to install. Installing NS2 and NS3 were almost difficult. Both were almost fully installed. However some libraries could not be installed to make it successful. If they had been installed, then I would have to face the lack of expertise as well. As a result, a new simulator had to be developed to implement the resolve of this project.

3.1 Development of Custom Simulator

3.1.1 Requirement Specifications

The simulator should be able to run in any operating system if all the necessary applications to help support and run it are installed. The developer turned out to be the only user. It will direct the user on how to input the vehicle names, their speed and their time of entry. Depending on the size of the roundabout, which is also a combination of lanes, the user should be able to specify the number of lanes and also name them. The purpose of this simulator is as follows:

- Generate cars onto lanes from specifications provided by the developer.
- Print information from the simulation that can be input into the NS-NAM (the NS2's animator) so as to get a visual representation.
- Provide a statistical report to help show average service time, and if possible average delay.
- Display progress of the simulation using a Graphical User Interface (GUI).

3.1.2 Tools

The main tools used for this project are as follows:

- Java Development Kit (build 1.6.0): This is a strong programming language that a good documentation to help with development.
- NetBeans IDE 7.1: This is a free integrated development environment and open source. It helps to quickly develop in Java and other programming languages. It can be downloaded from its website, which is netbeans.org.
- MITSIMLab: An open source simulator laboratory used for simulating traffic.
- Network Simulator: A discrete event simulator directed at research in networking. I used versions 2 and 3.
- Fedora: An open source linux operating system.
- Ubuntu: An open source linux operating system.
- VirtualBox: An open source application used for running virtual machines. It can be downloaded from its website, which is virtualbox.org.

3.1.3 Design

3.1.3.1 Simulation Components

The following are the main components of the simulation.

- Car
- CarFactory
- Lane
- LaneMaker

The next part explains the design of each of the components stated above. After, it focuses on the main component, which uses all of them to form the system.

Carfactory

This component creates cars for the simulation. It requires certain parameters to this. These parameters are gotten from a text file within the project package. It takes in the speed, name and entry time of the car. If I had enough time, I would have altered it to take both the source and exit lane of each car. With this file, it is easy to populate the large number of cars you would want to create. It also has functions for displaying the features of each car and even destroying it if not needed. So basically, it reads in the all the parameters of the cars and then creates them.

Lanemaker

This component is responsible for creating the lane for the simulation. It serves almost the same purpose as compared to the carfactory. However, it focuses on creating lanes. It uses the same paradigm as the carfactory, that is, reading in the necessary parameters to create a lane. These parameters are read from a file, which is also found in the project package. It also provides functions for knowing about the lanes being created. There is also a function to destroy a lane (though I have not found a use for that yet).

Lane

This component allows to also create a lane but in units. Its main purpose however, is to put cars into the lanes created by the lanemaker component. It has a sub-component known as "inLane()". This sub component takes in a parameter, which is a car, and then put its in the lane and records its progress. It captures every second with the distance traveled. It takes into the consideration the entry and exit times, and the distance covered at each point of the car. This allows it to transfer those details on the current lane to the next lane. For example, if car1 enters laneA at 0.0meters in 0second(s) and exits at 44.5meters in 12seconds, then car1 enters laneB at 44.5meters in 12seconds. This process follows through till the car1 exits the last lane recording each stage.

3.1.4 Implementation

The following components have been currently implemented:

- Car
- CarFactory
- Lane
- LaneMaker

To handle the simulation, a Main was implemented. This component is where the major actions are called upon to perform the simulation. In this component, I was able to carry out three levels of the simulation. The first level, which is the single car in a single simulation, puts a car in at the start of a lane. It then records its travel through the lane by capturing the distance traveled with every second. To move to the next lane, it took the current lane's distance covered and total time and used it as a start for the next lane. The car and lanes were created in the class. They were not read from a file.

In the multiple cars in single lanes level, a number of cars and lanes were manually created. This time, once a car completes the cycle of going through all the lanes, automatically, the next car goes through the cycle. This process ends as soon as the all the cars go through all lanes. Manually creating the cars and lanes made the work cumbersome because if I wanted to create about thirty cars meant that I had to put a lot more lines of code to make it possible. To curb this problem, I decided to introduce File I/O functions. The idea was to put the specs of the cars(any number) in a file from which I the component could read from to create the cars. I applied it to the creation of the lanes as well. It made the work less cumbersome and bit complex. This was the start of the Semi-Advanced Multiple cars in Single lanes. With all the lanes created, as soon as a car was created, it went through all the lanes. Adding more cars and lanes became much easier. I later introduced the time a car starts the parameters of the car. The aim was to allow some part of the car-following model, where right after a car enters a lane, the next car follows it, might be after two seconds. This is close to everyday traffic.

The final level of simulation, which is Advanced Multiple cars in single lanes, simulates a more realistic situation as compared to the semi-advanced. At this level, cars that are created can enter from any source lane and exit any. This would have been accomplished by adding more parameters, which would have been an entry lane and exit lane to that of the car class. The specs with the addition of the extra parameters of each car will still be found in the text file. A lot of conditional statements would have been introduced to make this possible. At this level, the work will inevitably cumbersome. Due to the lack of time, I could not reach this level. It would require more time as compared to the semi-advanced.

Below is a representation of the custom code simulation.

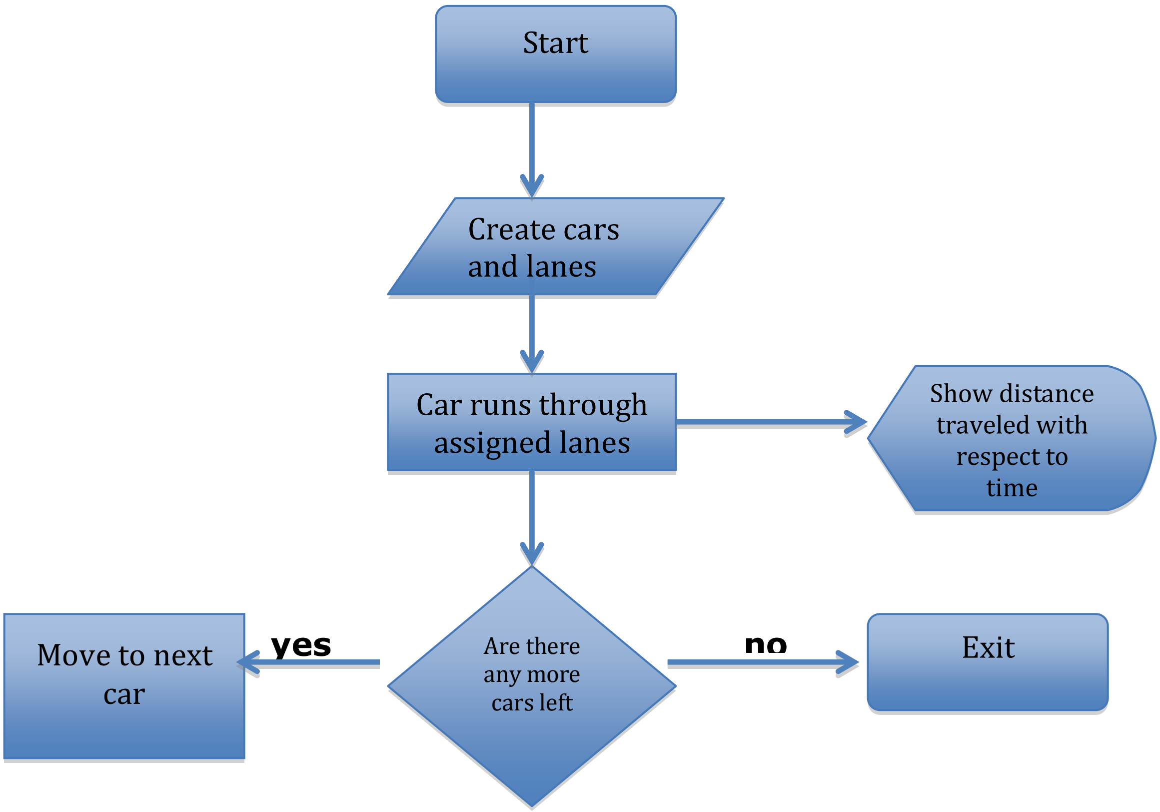

Fig 3.1: Flow Chart of Main Implementation

The diagram below represents the class dependencies.

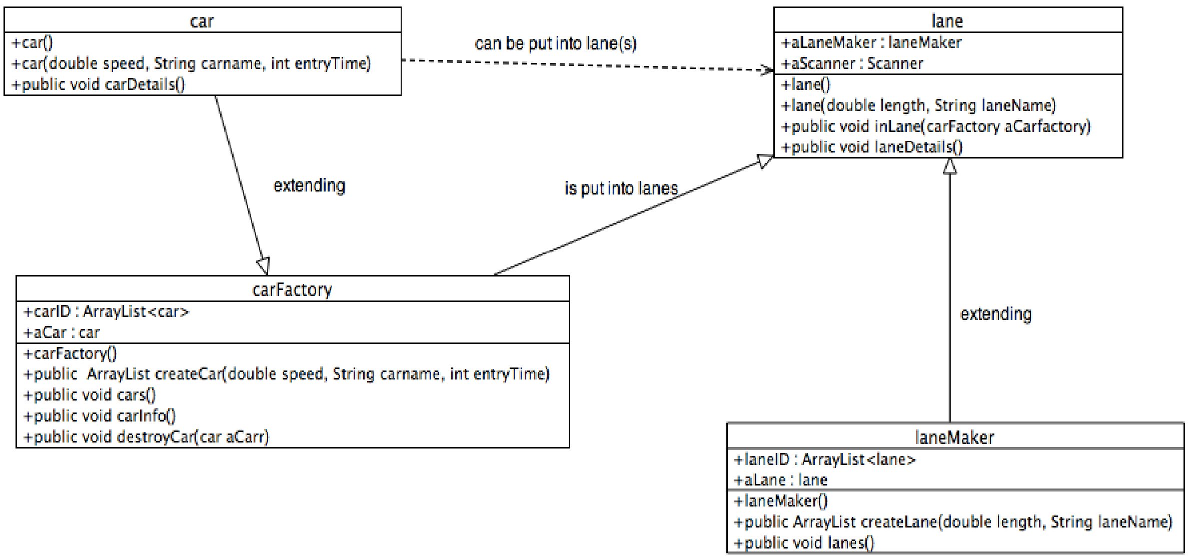

Fig 3.2: Class Dependencies

To make the comparison, I had to install already existing simulators. These are the simulators:

- MITSIMLab: The necessary components were successfully downloaded. The script to install the simulator however, failed to run. It complained about not having the right permissions to run the script. Changing to root still did not solve the problem. Being developed for Ubuntu, I could install in another operating system. There were attempts made though.

- NS 2 and 3: Getting to use these simulators was almost possible. The necessary libraries were successfully downloaded, and the install script ran successfully. In the NS3, though the script ran, not all the modules were successfully built. In the NS3, the modules that could not build were – brite, click and openflow. There were also issues of getting its animators to work. In the NS2, all the packages were built successfully, however, the animator could not run. It complained about placing it in the right environment before it could be used. NS2 was run in Microsoft Windows 7 and NS3 on Fedora.

On the next page is the representation of the custom code being simulated. With the assumption that a roundabout is a combination of lanes, vehicles generated are put on the starting lane and run through. They travel from one lane to the other recording the time and distance travelled.

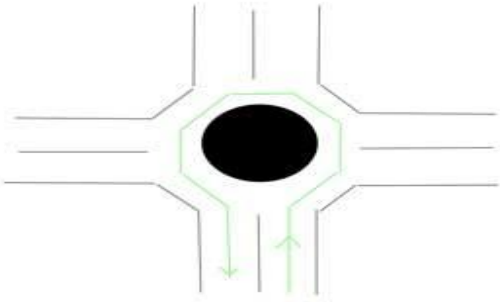

Fig 3.3 Representation of Custom Code Simulated

3.1.5 Installation of Simulators

Due to lack of time, I started to examine the use of already existing simulators to help with the project. I found four of them. First problem I really encountered was bad Internet service. All the simulators were not that big but required a good service to get chance to be downloaded. After which, the simulators require you to download certain specific libraries. There was always a risk of the Internet failing to cooperate.

Installing MITSIMLab required some amount of expertise and time. Though I ventured, I had root permission problems hindering me from installing the necessary scripts. Designed only for Ubuntu, it became clear to me to look for another simulator. NS2 looked promising. It forced to install Windows XP with which there were so many problems so I decided to use upgrade to Windows 7. I had to acquire a program that would put in a UNIX environment. Cygwin was recommended. It also required the right packages to be installed to be able to install the NS2. With every done, the animator however failed to run. Installing NS3, I downloaded all the necessary libraries needed. After, I run the script to install the ns3. It run well but could not build three modules. Thinking they may not be that important, it tried running it but it failed. The discovery of SimTraffic came very late not allowing me to dig further. It is a powerful and easy-to-use application that performs micro-simulation. It also comes with an animator to help visualize modeled traffic when simulated. It seemed to be the most promising of the simulators.

3.2 Limitations

This section talks about the caution that needs to be taken when using this study. There were a number of problems that came up within the project. The following outlines these restraints and their effect on this paper. The major part of it all was the gap between the different models and the custom model developed. Even though some parts were captured, there was still a lot more to introduce. Not all the variables were considered thus; only a part of the equation from the model was captured. This part was the free traffic state as given:

$$\dot{v}_{\text{free}}(v) = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta}\right]$$

Fig 3.4 Free-Traffic Equation derived from the Intelligent Driver Model

The maximum acceleration (a) and the velocity (v) were not varied thus it was obvious the state could be captured.

The custom code failed to capture the time of the vehicles entering the lanes. Consequently, a good analysis of the system on a macroscopic level could not be made. Analysis could only be made to an extent.

3.2.1 Data Collection

The main objective was to observe and record delays being caused by the traffic wardens and general traffic information at the Kwame Nkrumah. It would have required a knowledgeable group of 4-5 personnel. However, there was only one unskilled personnel. The alternative was to find an already existing dissertation with the necessary information with the right. The dissertation was found, however, it also complained about not being able to acquire the desired information. It explained that the information was released but was nowhere to be found. This made things difficult.

3.2.2 Technical Expertise

The literature found on the subject is technical. Fluid mechanics is an advanced subject in physics, which serves as the basis for most of the traffic flow models. The intelligent driver model is one of these models. Being able to formulate a model would require a substantial understanding of the mechanics. Due to the difficulty of coming up with new models, several variables described in simpler models (such as maximum acceleration and desired velocity derived from the microscopic model) were used to guide some of the models looked at.

A lot of the simulators require using linux operating systems. They also require some level of compiling code and certain libraries. For example, installing NS3 in Fedora required downloading some libraries like mercurial, which is needed to work with the ns-3 development repositories.

3.2.3 Simulation Errors

A lot of assumptions were made in this simulator. Data being printed sometimes looked like vehicles overtaking each other were implemented in the design of the system. A combination of lanes was generated to create a mock circle with only two exits, which is not a real circle.

Due to time constraint, the simulator was not fully developed. With the hope of printing information in a particular pattern, it could have been input into the NS's animator (NS-NAM). Having never developed a simulator before, even made the work more cumbersome but exciting.

CHAPTER 4: SIMULATION EVALUATION

4.1 Results and Analysis

In the simulation of the levels, I started with a fixed speed for all the cars. Though all the lanes were of different, no matter how many cars were put in, they were all able to complete the cycle of going through the roundabout. The completion of the first level of simulation made it possible to develop to the second and third levels where the car-following models are introduced. These levels were able to cover some aspect of the microscopic models. The final stage of the simulation would have done the same. To analyze the results, I decided to use both approaches. The simulator was designed to cover some aspects of the macroscopic model.

With the microscopic approach, the simulator designed was able to cover some aspects of the Intelligent Driver Model. It was able to emulate the free traffic state, which is:

$$\dot{v}_{\text{free}}(v) = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta}\right]$$

Fig 4.1 Free-Traffic Equation derived from the Intelligent Driver Model

From the equation, my desired velocity (v_0) was represented each cars own velocity. Having no acceleration, there is the assumption that the car's velocity (v) had reached the desired velocity. Thus, giving a value of

1 even though the acceleration component (δ) may be present. With this, the equation then gives a value of 0. Alternatively, with the maximum acceleration still being 0, the outcome of the equation would still be 0. This is so because when the desired velocity is reached, acceleration stops and gets the value 0. The higher the velocity, the lower the acceleration; this is an inverse relationship between velocity and acceleration.

On the macroscopic level, the simulation was able to capture little's theorem. The average number of cars could be determined. The average service rates to complete some of the lanes were manually input. The user determines the number of cars and their arrival rate as well. This captured all parts of little's theorem, though it was on a controlled level. This made it possible to know the rate of cars being put into the system.

Not being able to install any of the simulators, no comparison could be made. Simulators included sample code so it was possible to look at how these may be modified. Exploring one of them, some similarities were found. Especially providing the details of vehicles, the lanes, and their creation. The vehicles were then put into the nodes. There was also the fact that a lot of assumptions were made. This provided some level of feedback and analysis on both levels of simulation. This made only made the situation slightly surreal as not all the variables needed to create a more realistic situation was implemented.

CHAPTER 5: CONCLUSION

The dissertation was set out to explore the explore traffic modeling and simulation as a means of providing an alternative method for conducting traffic at the Kwame Nkrumah Circle at peak hours. The developed custom simulator was to be compared with already existing simulators, which were the MITSIMLab, NS2 and NS3. The results were to be analyzed using equations provide in the microscopic and macroscopic models.

The custom simulator tried to emulate the Intelligent Driver Model (IDM). It was successful in capturing a part of it, which was the free traffic acceleration equation.

5.1 Further Work

For further work, the other levels of simulation would be completed. A lot more variables, such as varied acceleration, would have to be introduced. With a complete understanding of the microscopic and the macroscopic models, a better custom simulator can be developed to emulate everyday traffic. It would introduce the full equation provide by the Intelligent Driver Model (IDM), the microscopic. It would then use the macroscopic to analyze the overall the system that would be generated.

Currently, the simulation is run using NetBeans IDE 7.1, which will make the introduction of a graphical user interface (GUI) possible. The output of the simulation would then modeled in way to allow graphing and at some point used as input into other simulators if possible. Some of them come with an animator component that makes it possible to visualize the simulation. This would require studying some other simulators and how data is input to them to allow simulation.

Other parts of the intelligent driver model will be introduced, systematically. A gradual process will help succeed in capturing the other variables such as the jam distance and the safe time headway. This will help capture a more realistic representation of everyday traffic thus eliminating all assumptions. With no simulation errors, the custom simulator will implement a more realistic situation producing a more realistic feedback that can be used for better analysis.

The dissertation has more room for improvement. Future developments may be able to simulate all kinds of traffic at roundabouts. It would also aim at making it not too complex so as to allow future users to simulate their own traffic.

References

City of Buffalo. (n.d.). *Traffic Signal Synchronization Project Timeline*. Retrieved April 14, 2013, from http://www.ci.buffalo.ny.us/Home/City_Departments/Public_Works_Parks_Streets/TrafficSignalTimeline

Eldor Contracting Corporation wins New York State Department of Transportation (NYSDOT) contract. (2012, December 10). *Public Works*. Retrieved April 14, 2013, from http://www.pwmag.com/traffic-control/eldor-contracting-corporation-wins-nysdot-contract.aspx

Hennessy, D. A., & Wiesenthal, D. L. (1999). *Traffic congestion, Driver stress, and Driver aggression. Aggressive behavior*, *25*(6), 409-423

Installation. (n.d.). *- Nsnam*. Retrieved March 28, 2013, from http://www.nsnam.org/wiki/index.php/Installation

IQ Central Ultimate Control. (n.d.). *J.O. Herbert Co., Inc.* Retrieved April 14, 2013, from http://www.johinc.com/products/traffic-controllers/iq-central-ultimate-control/

Little, J. D., & Graves, S. C. (2008). *Little's Law*. Retrieved April 6, 2013, from http://web.mit.edu/sgraves/www/papers/Little's%20Law-Published.pdf

McCanne, S., Floyd, S., & Fall, K. (1997). *The LBNL network simulator.Software.* Retrieved March 30, 2013, from http://www.isi.edu/nsnam

Miller, S. (2011). *Traffic Modeling*. Retrieved April 2, 2013, from http://home2.fvcc.edu/~dhicketh/DiffEqns/Spring11projects/Scott_Miller/Project.pdf

Ns-3. (n.d.). *Ns3 RSS*. Retrieved April 11, 2013, from http://www.nsnam.org/

Queueing Theory Basics. (n.d.). *Queueing Theory Basics*. Retrieved February 3, 2013, from http://www.eventhelix.com/realtimemantra/congestioncontrol/queueing_theory.htm

Queueing theory. (n.d.). *Wikipedia*. Retrieved February 3, 2013, from http://en.wikipedia.org/wiki/Queueing_theory

Sampson, H. (2006). *Investigating The Effectiveness Of Traffic Learning Agents In The Management Of Urban Vehicular Traffic*. Accra, G.A.

SIRS-200 ITS – TRAFFIC MANAGEMENT SENSOR. (n.d.). *SIRS-200 ITS Traffic Speed Sensor*. Retrieved April 14, 2013, from http://www.saabgroup.com/en/Civil-security/Land-Transport-and-Urban-Security/Traffic-Management-System/SIRS-200-ITS-Traffic-Speed-Sensor/

Stephen, D. (n.d.). LASTMA and Lagos Traffic Solution. *INFORMATION NIGERIA*. Retrieved April 14, 2013, from http://www.informationng.com/2011/11/lastma-and-lagos-traffic-solution.html

Traffic signal system to be centrally controlled. (n.d.). *Ghana HomePage, Resource for News, Sports, Facts, Opinions, Business and Entertainment*. Retrieved April 14, 2013, from http://www.ghanaweb.com/GhanaHomePage/NewsArchive/artikel.php?ID=264625

Trafficware. (n.d.). *Trafficware*. Retrieved April 6, 2013, from http://www.trafficwareinc.com/transportation/product/simtraffic-8-0

Welcome to VirtualBox.org! (n.d.). *Oracle VM VirtualBox*. Retrieved April 10, 2013, from https://www.virtualbox.org/

APPENDICES

Appendix A: Installation of NS3 for Linux (Fedora)

1. Download the NS3 from its website: http://www.nsnam.org . It downloads with the filename ns-allinone-3.16.
2. Put the file in an appropriate location, which will not harm any other files you might have. Preferably in your admin folder.
3. Open terminal and move into the directory where the file is located.
4. Use the command: "tar xzvf <filename>" to extract the file into the current directory.
5. After, there are some libraries you will need to download. A list of those libraries can be found if you follow the link: www.nsnam.org/wiki/index.php/installation
6. To download these libraries use the command: "yum install <library>". You need to be connected to the Internet for this to be successful.
7. After a successful download, check to see that you are in the ns3 folder by typing 'pwd'.
8. List the files and folders in the directory by typing 'ls', and look for 'build.py'. It is a script that checks the libraries present in the system and installs the necessary ones for the simulator to run.
9. When I ran the script, some modules were not built. They were 'brite', 'click' and 'openflow'. After having a look at the ns3-tutorial that was recommended, it realized it was quite complex. Thus I Put it on hold and explored the other simulator.

Appendix B: Installation of NS2 for Windows

1. Download the NS2 from http://www.isi.edu/nsnam/ns/ns-build.html. It is recommended to download the packages separately, though there is an all in one version. Its filename is ns-allinone-2.35.
2. Extract the file into drive C: .
3. Download Cygwin from http://www.cygwin.com
4. During the installation of Cygwin you will need to select these packages manually. If not, it will only install their default packages, which are not enough when it comes to development. These packages can be located at http://wsnlab.ir/how-to-install-cygwin-ns-2-in-windows/.
5. (You may not be able to find some of the packages that have been listed. I think they were added to some others so as not to make it look a lot.)
6. After selecting them all, start the installation.
7. After installing, you will find a shortcut of the application on your desktop. Run it.
8. Change to the directory where you copied the extracted NS2 folder.
9. Type "cd <directoryname>" to enter where directory name is ns-allinone-2.35.
10. When you list the files and folders using the 'ls' command, you will find an install script. Run it by typing './install'. This will build the necessary modules to make the simulator work.