CSE 6730 - Final Project Report

Simulation of Traffic on Peachtree Street

Chengwei Li, Vishnupriya Pradeep, & Ankit Srivastava
(Group 7)

Georgia Institute of Technology

Abstract

Our simulation study compares the effect of synchronized versus unsynchronized signals on the Atlanta Peachtree Street by studying average intersection waiting time from the 10th street to 14th street. Influence of aggressive drivers on the flow of traffic was also studied to provide an insight into the real life scenarios. This model could be used for analyzing traffic conditions in other streets and also to help improve the current traffic conditions in Atlanta by modifying the parameters. The vehicle and street data for the simulation process was obtained from the NGISM data set. Priority queues and central simulation engine was used to recreate the traffic conditions along the Peachtree street. Simulation engine was triggered by three events such as arrival, departure and crossed across the intersection.The Synchronized traffic simulation performed better than unsynchronized travel time distribution reducing the waiting time by 10.82%. Increasing the aggressive driver percentage from 0 to 5 , resulted in an overall increase of waiting time by 28.7%.

1. Introduction

Peachtree Street is one of the prominent roads in the city of Atlanta, running from Five Point to Midtown in the city. Many important buildings are located along or near the street like Margaret Mitchell House, Bank of America Tower, Fox Theater, Georgia-Pacific Tower, World of Coke, etc. In addition, multiple corporate offices are located along the street. Therefore, the street has to cater to a large number of vehicles, especially during the peak hours. This leads to traffic congestion on the street with longer waiting times at intersections and increased travel times in general.

Traffic conditions on the street can potentially be improved by making some changes on the street e.g. by changing the traffic signal timings. However, experimenting with these changes on the real street may prove to be too disruptive and is therefore undesirable. Instead, a simulation model of traffic on the street can be developed. This model, once validated against real data, can be used to give a good estimate of how any proposed change would effect the traffic conditions.

2. Problem Description

Primary objective of this project is to develop a simulation model for a section of the Peachtree Street (from 10th to 14th street) and then use the model for predicting travel time of vehicles traversing the section during evening peak period(4PM - 4:15PM). The validated model can further be used for predicting the travel times with different signal timings and hence determine signal timings for intersections with static signal timings such that average travel time is minimum. The model can further be refined to incorporate reactive signal timings in order to determine the benefits.

2.1. Characteristic of Area of Study

Atlanta Peachtree Street between 10th street and 14th street are considered for this study. There are 5 intersections between this area of study and they are labeled as 1, 2, 3, 4 and 5 respectively and they are labeled progressively from 10th street to 14th street. The length of each intersection is given in Table 1 and their section length in Table 2. Section refers to the roadways connecting the intersection. Intersection refers to transfer point to the next section of the street.

Intersection number	Length of the intersection (ft.)
0	99.65
2	129.79
3	73.815
4	66.979
5	117.411

Table 1: Intersection lengths

Section number	**Length of the section (ft)**
1	127.39
2	441.43
3	412.07
4	353.727
5	343.922
6	11.730

Table 2: Section lengths

2.2. Data characteristics

Totally 159 vehicles were considered assuming that the size of the vehicle does not interfere with the nature of the traffic. Primarily vehicle moving in the North and south bound was considered as our main objective is to simulate the waiting time along the Peachtree Street.

2.3. Data collection

The other parameter considered for the study are the percentage of left and right turn at each intersection, velocity of the vehicle, originating and destination zones to determine the path of the vehicle. Percentage of vehicle turning left and right is given in Table 3.

Vehicle traveling direction	Percentage of vehicle turning
North Bound(NB)	33.7766
South Bound(SB)	66.2234
NB TH at Intersection 0	29.25532
NB TH at Intersection 2	1.861702
NB RT at Intersection 2	0.265957
NB RT at Intersection 3	1.06383
NB RT at Intersection 4	0.265957
NB RT at Intersection 5	0.797872
NB LT at Intersection 5	0.265957
SB TH at Intersection 0	12.76596
SB TH at Intersection 5	34.04255
SB RT at Intersection 2	1.06383
SB RT at Intersection 3	0.265957
SB RT at Intersection 5	18.08511

Table 3: Percentage of turns

2.4. Data Cleaning and Input analysis

Unique vehicle were selected and only lane 1 and 2 was considered as the simulation has single lane. The Epoch time for the vehicle were subtracted to estimate the time difference between vehicles. The direction of whether a vehicle travels in the north and south bound was determined by considering the percentage of vehicle travelling in the respective direction.

Intersection and sections: The average length of the intersection was used for the simulation as the magnitude of difference between the different intersections was not significantly different and the average length was determined to be 97.520 Ft and the average section length was determined to be 387.78ft.

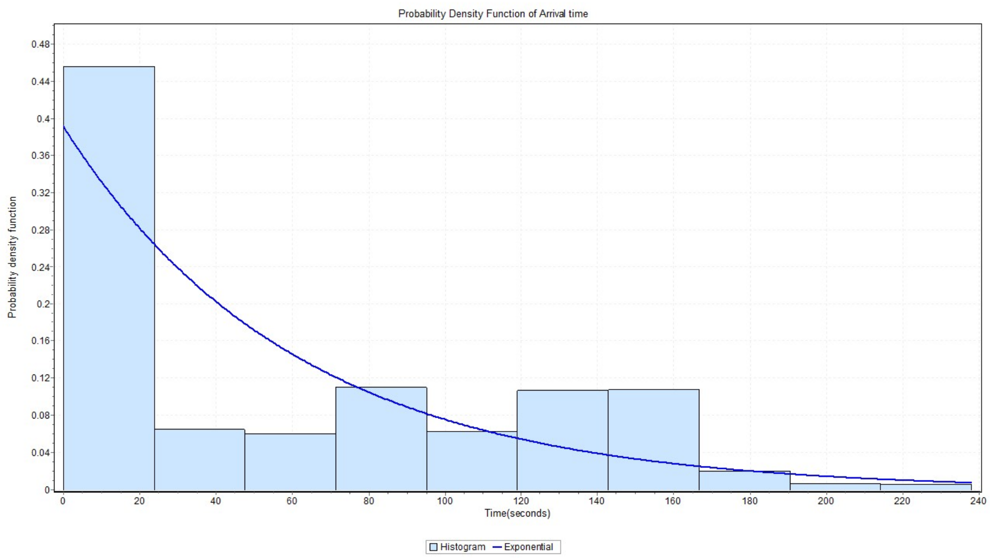

Figure 1: Travel time distribution of vehicles from the NGISM dataset.

The travel distribution and the fit of the data were obtained using exponential distribution with a chi-square value of 3.758 (as shown in Figure 1). The timestamp at the first occurrence of the vehicle along with its last appearance on the simulation was considered to obtain the travel time for each vehicle. The overall list of parameters used for the simulation is given in Table 4.

Parameter	**Description**	**Unit**
Vehicle ID	Vehicle Identification number	
Epoch	Epoch time elapsed since Jan 1, 1970	ms
Vehicle velocity	Instantaneous speed of the vehicles	Ft/s
Lane ID	Current lane position	
Origin	Origin Zone	
Destination	Destination zone	
Intersection	Intersection in which the vehicle is travelling	
Section	Section in which the vehicle is travelling	
Direction	Moving direction of the vehicle	
Movement	Movement of the vehicle	

Table 4: Parameters used in the simulation model

3. Literature Review

Existing literature on traffic modeling was surveyed in order to develop an understanding of simulating traffic before developing a model for the given problem. Depending on the objective and application of simulation, existing models for traffic simulation fall into three categories: macroscopic, mesoscopic, and microscopic [1]. Car-following model was one of the earliest models for traffic simulation which modeled behavior of drivers following a car [2]. Cellular automata approach models the traffic as vehicles moving between automata elements, depending on different parameters [3]. Cell transmission model has been used to model traffic flow in accordance with hydrodynamic theory [4]. Queuing models model the traffic as a queue in which vehicles enter and leave based on traffic conditions [5].

Queuing models have been widely used for modeling traffic because of the intuitive way in which flow of traffic can be represented using queues. In a discrete event simulation of the traffic, the queue holds the pending events which are addressed in a priority based manner. Priority queues are therefore used in queuing models. Different types of priority queues have been developed for different scenarios. Some implementations of queues include Pagodas, skew heaps, splay trees, pairing heaps, and binomial queue [6].

Modeling intersections is one of the difficult areas because of multiple factors at play like the inherent supply constraints, crossing of vehicles, pedestrians, conflicts due to speeding, lane changing and various other factors [7]. Intersection safety issue brief 2007 by Federal Highway Administration reports that 39.7 percentage of the accidents occur at the intersection or at intersection related areas [8]. A generic model developed particularly to combat the intersection modeling was not able to capture the reality, where the flow of traffic was described incorrectly or less like the real life situation [9].

Multiple lanes add to the complexity of the model because it introduces limited ability of the lane changers to accelerate their speed and also influence the traffic flow and reduces the capacity of the roads. Four parameter multi-lane model was used for modeling the traffic flow to recognize the lane changing patterns among drivers and also their effect on the road capacity in the freeway as well as signalized intersection [10].

Various simulation software have been developed for studying interaction, merging, effect of driver behavior, etc. on traffic. ARTEMis, which can be used to simulate both freeway and signalized urban arterial networks, is able to reproduce the speed, gap between two vehicles and lane change patterns in a reasonably realistic manner [11]. Some other widely used traffic simulation software include MITSIM [12], PARAMICS [13], and AIMSUN-2 [14].

4. Conceptual Model

For the purpose of modeling traffic in this project, we use a stochastic bounded-horizon discrete-event simulation. We use a simple hold model, where we maintain a future events list (FEL) in the form of a priority queue which holds the pending events in the discrete event simulation. The priority of an event in the FEL is determined by its time-stamp. Since this is a bounded-horizon simulation, we only process those events which are scheduled to happen before the cutoff time and ignore the rest of the events.

The system under investigation (SUI) is a section of Peachtree Street between 10th and 14th Street. Northbound and southbound vehicles form the entities in our study. However, we focus on northbound vehicles and only consider left turning southbound vehicles since only interaction between vehicles going in opposite direction is because of left turning vehicles. We consider an entity to have exited the SUI as soon as it leaves the section, by turning, by going south on the intersection on 10th Street, or by going north on the intersection on 14th Street.

4.1. Input

We use NGSIM data for getting the data for vehicles as well as for the signal timings. The NGSIM vehicle data is used for getting vehicle arrival times and correspondingly the inter-arrival times. Input modeling is done on the obtained data to get a distribution for the inter-arrival times. The distribution is then used for scheduling the vehicles. Other input parameters such as average time required to cross a section of the road, average time taken to cross the intersection, percentage of vehicles taking a turn (or going through) at every intersection is also obtained from the vehicle data and is passed as input to the model.

4.2. Output

The simulation model outputs all the statistics for the vehicles that exited the system before the cutoff time. This includes entry time, exit time, origin, destination, and total waiting time for all the vehicles. Both average waiting time as well as average travel time is calculated using this data.

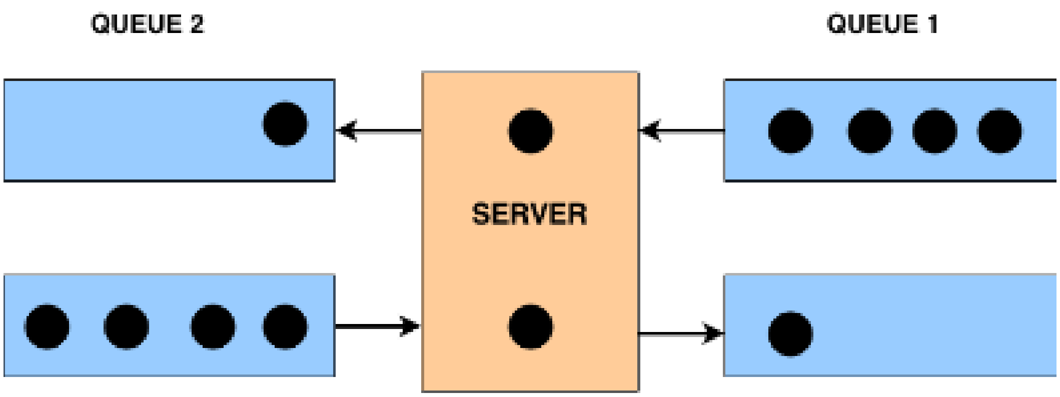

Figure 2: Representation of the server queue model

Signal Timings. We need one state each for "go" and "stop" signals. For the purpose of our simulation model, these two states are enough for indicating the state of any intersection signal at any given time. The signal state depends on the current simulation time.

Length of the Traffic. The length of the traffic would affect the travel time for the vehicle, so does the length of vehicle itself and the spacing length. Again, as the length of traffic would follow a certain probabilistic distribution, maximum number of vehicles on the road could be calculated and used as the time-interval to describe the length of the traffic.

Event Description. We use five events to describe the behavior of the vehicles, which include events for arrival, arrival for left turning vehicle, crossing, crossing to left, and departing an intersection.

Event Presenting. Once the event occurs, its existence would be in the form of adding or leaving the queue for storing the event to be processed.And these discrete events would be eventually handled by the simulation engine.

Traffic Entity. Traffic entity is the discrete event simulation would be a certain type of individual data structure that hold the representative information of the simulation.Information would be the abstracted data that would be interpreted to describe the real problem.Like we use integer to describe the current location of the vehicle. And the time-stamp to describe the intended action that the vehicle would take.

4.4. Simplifications

The SUI described above is a complex system and hence some simplifications were required for modeling the system:

- Peachtree is a four lane street. However, for the purpose of this simulation, we consider it to be a single lane each for northbound and southbound traffic. We make this simplification because lane changing between intersections isn't expected to have a significant impact on the waiting times.

- We don't consider variations in vehicle lengths and instead consider all vehicles to be of uniform length. Further, we change all vehicles to point sized vehicles and consider effects of lengths, if any, in different crossing times. This is a valid simplification because most of the vehicle used in the city limits are of standard size, with a few exceptions.

- We assume constant velocity in all the sections of the Peachtree Street and assume infinite acceleration and deceleration at the intersections.

4.5. Assumptions

In our simulation, we made few assumptions as it would be complicated to model a very detailed system and since the overall objective to understand the travel time distribution.

- We assume a symmetric system and focus on modeling northbound vehicles, only considering left turning southbound vehicles. This seems like a valid assumption to make because there is no other way a vehicle from northbound street can interact with a vehicle on the southbound street.

- We assume infinite capacity of waiting queues at the intersections. This assumption will only come into play if there are multiple vehicles at the intersection and can be easily removed if this assumption leads to unrealistically large queues.

- We consider the traffic to be constant on each section i.e. we don't model departure or arrival of vehicles between intersections.

- We assume the system to be working perfectly i.e. we don't take into account any traffic violations, breakdowns, accidents, weather hazards, etc.

5. Software Description

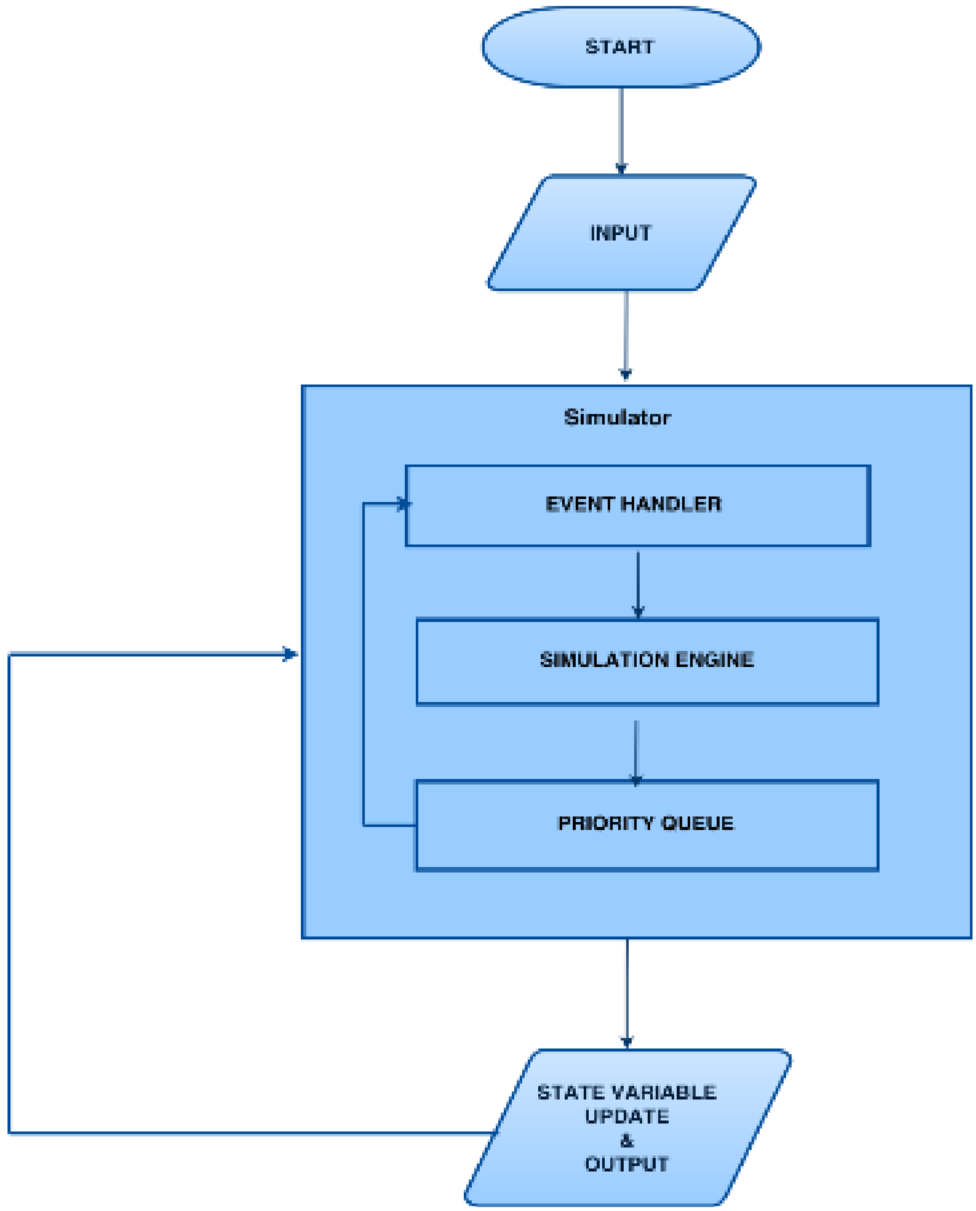

Figure 3: Schematic of the software architecture

5.1. Resources

The simulation model is coded in *C++11*. *Boost.Program_options* [15] library is used for parsing command line arguments passed to the simulator. Even though the executable generated from the C++ code can be used for running the simulator, there is a wrapper script written in *Python2.7*, which can be used with the simulator for ease of experimentation and enhanced usability.

Since C++11 has different libraries for generating random number corresponding to different distributions, we used the standard functions for getting random numbers wherever possible. However, since it can be done in C++11, we configured all these functions to use the same underlying default random number generator engine. We did Chi-square test for the random values generated using the default random number generator using $k = 1000$ and $n = 10000000$, we got $\chi^2 = 988.978$ for which the corresponding $p = 0.41 (> 0.05)$. Therefore, we accepted the random number generator.

5.2. Architecture

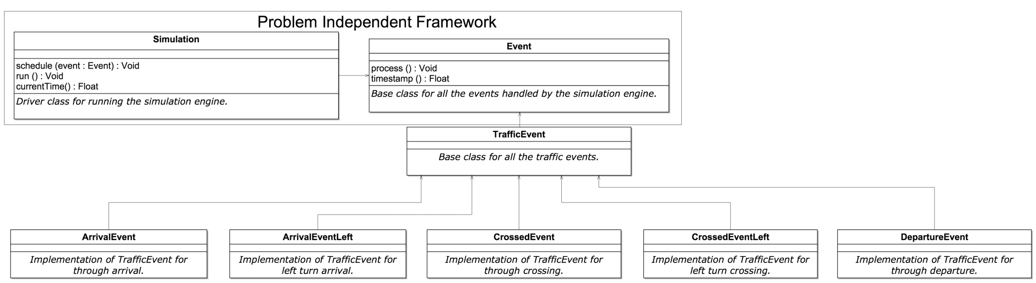

Figure 4: UML diagram of the software framework

For implementing the discrete-event simulation at hand, we designed an application independent simulation engine for processing all the scheduled events. In C++, the simulation engine was implemented in a class *Simulation* which operates on an abstract class *Event*. This, as can be seen, is independent of the application at hand and can be used for implementing any discrete-event simulation. It can be done by implementing the abstract *Event* class with classes for processing application specific events.

We implemented five classes for our traffic simulator, corresponding to the five events that we use in our simulation model: *ArrivalEvent*, *ArrivalEventLeft*, *CrossedEvent*, *CrossedEventLeft*, and *DepartureEvent*. Each of these classes implement *process* method from the *Event* class corresponding to the action that needs to be taken when this event is triggered. These classes will schedule future events based on how the events are processed. UML diagram for the simulation engine, along with all the implementations, is shown in Figure 4.

We have defined an *Intersection* class which keeps track of the queues at all the intersections as well as the state of the signal. Since we have multiple parameters that can be varied, we have defined a class *TrafficParameters* which parses all the parameters from a file and stores them, which can then be accessed by all the other classes.

5.3. Methodology

The procedure to process the interaction between the events is called schedule procedure. And the schedule procedure would pass the data of the event as an entity to the simulation engine as a discrete event. All the discrete event would be listed as a priority queue in accordance with their priority, in our case the priority would be the time-stamp of the discrete event. And simulation engine would indeed process the discrete event one after another, thus let the simulation process running. The event handler would be in the function to update the variables that are defined to describe the total simulation. State variables would be in the form of simplification and abstraction for the conditions and state of the real event to be modeled.

Event handlers are related to each other and would trigger its related event one after another after it has received the related data and begin to update state variables. And state variables in return would affect the sequence of the event to be generated. And these event would affect the following cars event after they have arrived at the intersection, like if the previous car stopped at the intersection, other car that arrive after it would join the waiting queue instead of drive through the intersection. These states variables and event can affect each others behavior and once properly defined, would be the ideal description to model the behavior of the vehicles passing through the intersection.

6. Verification

For the verification of the model that we built, we look at several aspects of the model to determine whether we have build the right one. One process is to test whether our model has perform the right behavior that we have defined. The process of confirming it is correctly implemented with respect to the conceptual model . During verification the model is tested to find and fix errors in the implementation of the model. For example, we define our model to have the major three event , which is Arrival, Crossed, Departure event. However, These event have sub-events that is related to it and in the callback functions the event handler should decide to process its sub-event according to the state variables that could change its behavior.

Take the arrival event as an example, when the arrival event handler is called, the function has to decide that whether the vehicle should perform crossed event, however, in order to perform the crossed event, the vehicle first should decide if there is queue in front of it, and if not, the specific vehicle should jump into the queue that is designed to be the container for the vehicles at the intersection. And then, as the crossed event has many sub-event, like the left turn , right turn, and the crossed event designed for the opposite directions. The vehicle should determine its action first and then check the state variables like whether the intersection is occupied and the traffic light is green, or if not , if it would perform the right turn behavior ,should follow some kind of distribution model to make the right turn even if it is not green light. These behaviors of the vehicle are the conceptual model that we have derived from the reality. And in order to test the model is exactly behaving like we have designed, we could focus on the one specific vehicle and trace its behavior. And print out all the actions it has taken during the event.

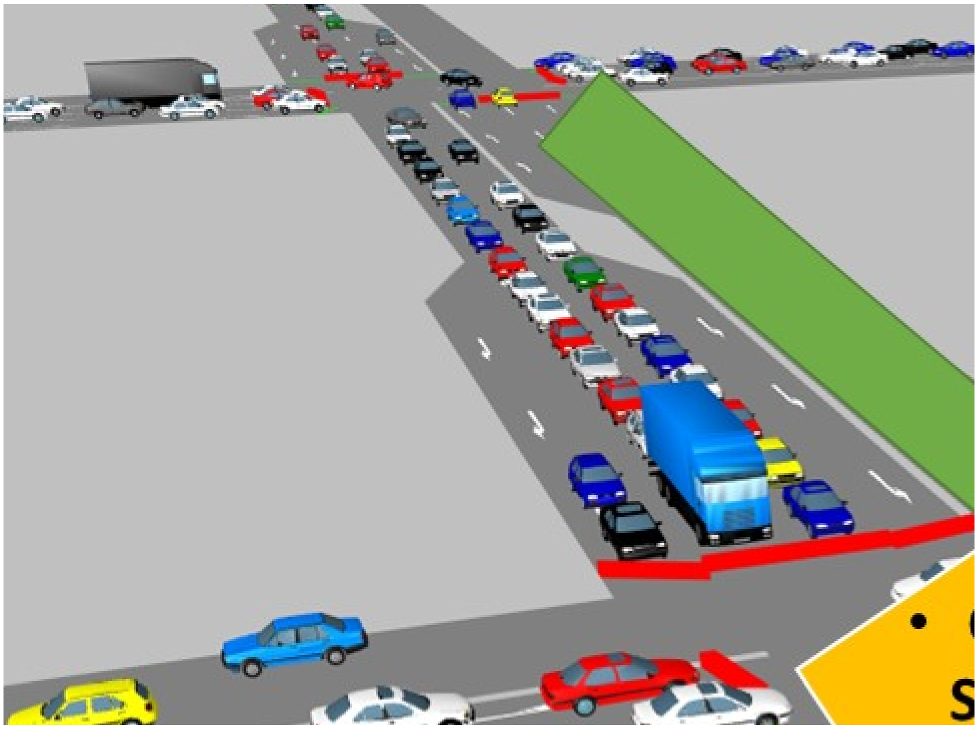

Figure 5: Changing of the event and their related state variables

We could see the action of the vehicle is making while in the system, and according to its time-stamp and its event to decide whether our model is making the right action. And to test its correctness, we also have to compare its with other vehicles and their interactions with each other.

7. Validation

Various processes and techniques are used to assure the model matches specifications and assumptions with respect to the model concept. The objective of model verification is to ensures that the implementation of the model is correct

7.1. Input Validation

First, we took a look at the data that we collected, the distribution model that we derive for the input varibles, we may take the arrival interval time for the vehicles traveling from south to north as an example. The data that we got from the distribution model that we derived from the data set. Then compare it.

7.2. Travel time validation

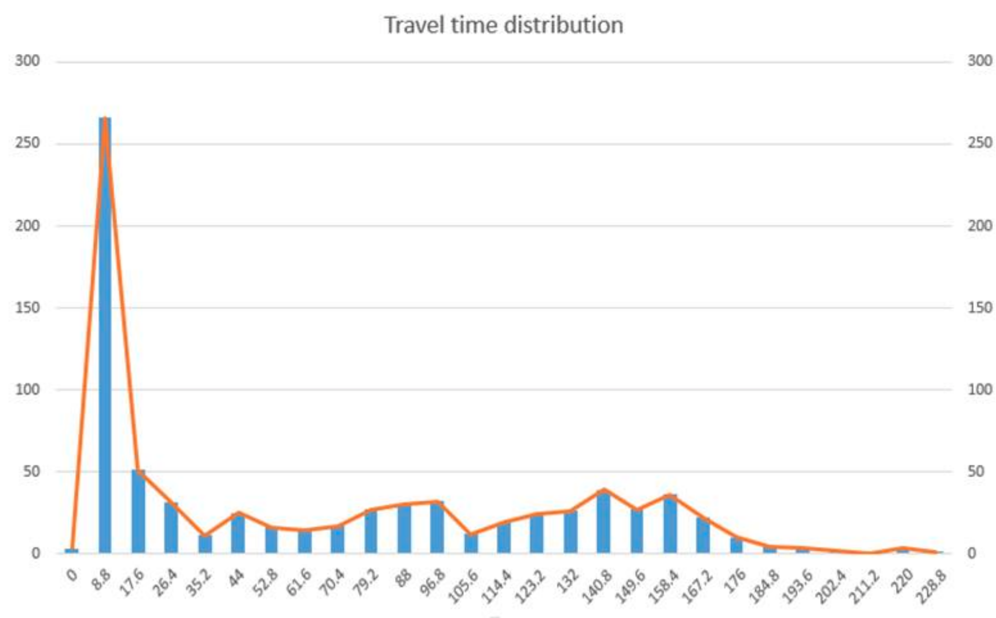

Figure 6: Travel time distribution of original dataset

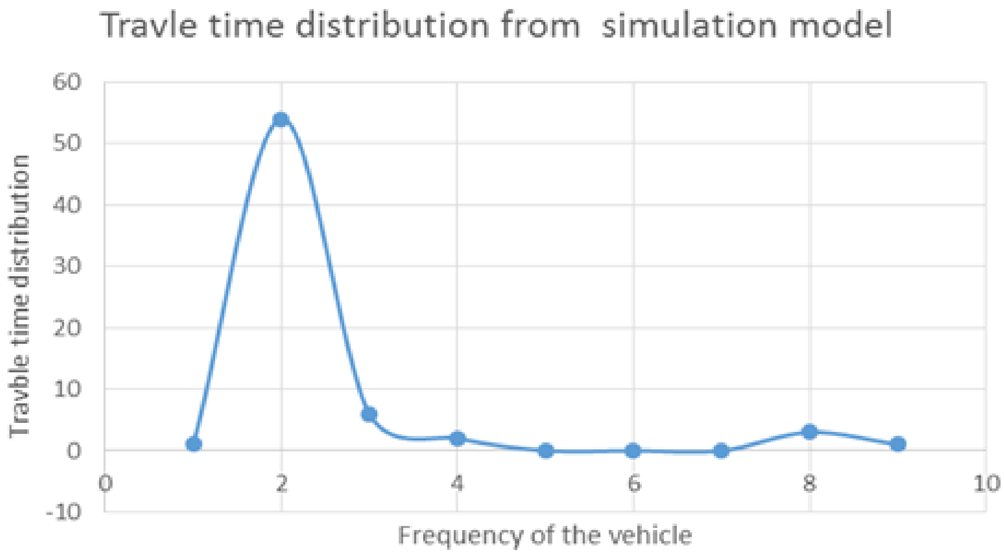

Figure 7: Distribution of the traveling vehicles traveling time

7.2.1. Synchronized Timing

Figure 8: Signal timing

We use the synchronized timing segment to run our experiments and efine all the time line of the traffic light at each intersection to act the same way. In order to make the traffic signal timing synchronized, we just assume that all the traffic light would follow the same timing sequence, and in our modeling, we assume the time length for the green to be 50s, and 60s and compare its effect on the travel time. For simplicity and convenience, we assume the red light timing to be constant 40s, and get the results. Just slightly changing of the green light time sequence would result in huge difference in the traveling time and average waiting time of the traveling vehicles. To make the model be close to the reality, we decide to set the green light to be 60s.

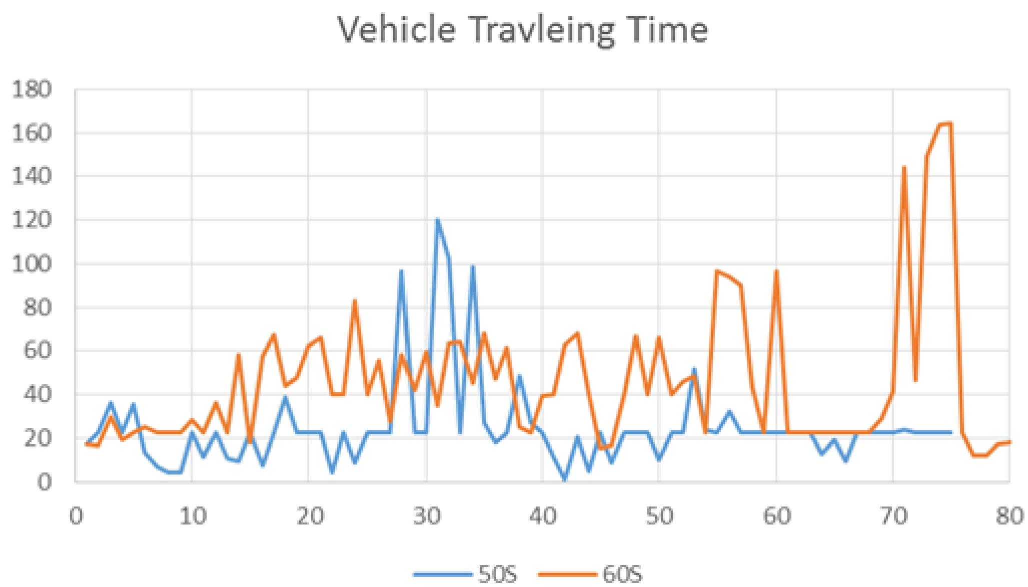

Figure 9: Vehicle travel time

Green light signal timing (s)	Average waiting time (s)	Average traveling time (s)
50	39.78	75.41
60	13.34	50.75

The time length was different with the data that we used for validation is that because we increased the right turn and left turn possibility of the vehicle, making its destination varies greatly than reality. So we set the green timing to be 60s and the red light timing to be 40s to make the traffic light signal timing to be the same to model the synchronized traffic timing behavior.

7.2.2. Unsynchronized Timing

For the unsynchronized traffic timing, we use the time off-set to make the synchronized timing, and still keep the timing segment of the red and green traffic light to be the same to control the variables. Then use this data to compare with the synchronized data to get the effect of the signal timing on the travel.

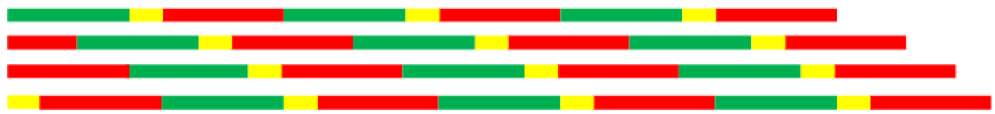

Figure 10: Signal timing

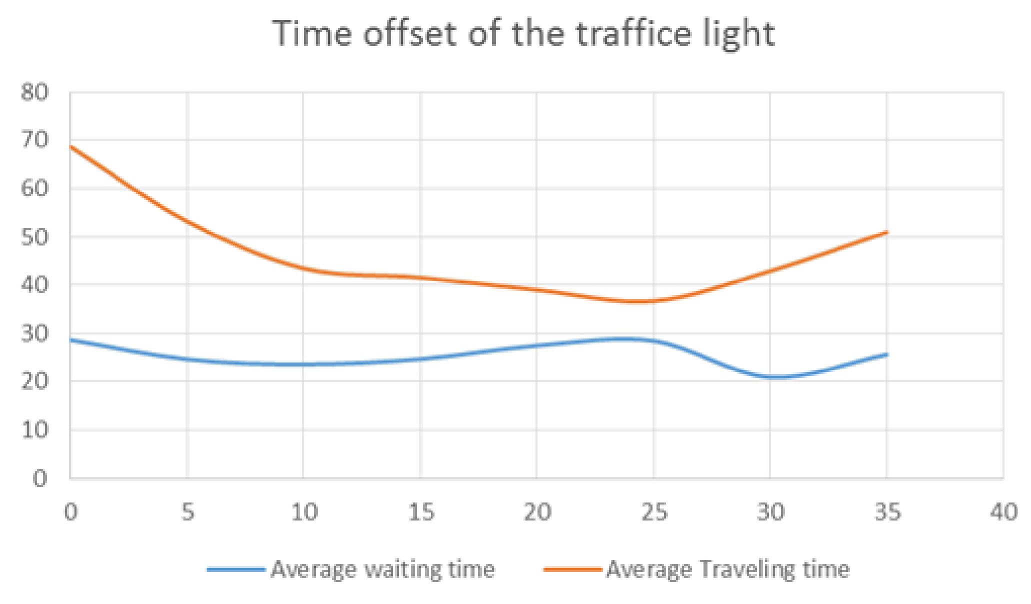

Figure 11: Vehicle travel time

Offset time	Average waiting time (s)	Average traveling time (s)
5	24.65	53.16
10	23.61	43.48
15	24.71	41.56
20	27.47	38.98
25	28.44	36.71
30	20.94	42.94
35	25.64	50.97

Thus we could see that the offset of the signal timing would help to reduce the time for the traveling vehicles. We adopted different time offset to test the effectiveness of the unsynchronized time method. For the offset that is near the 25 seconds, we saw the smallest travel time, but for the waiting time, there is no apparent pattern for it. This phenonemo was caused by that the waiting time was actually affected by many factors. Like the left turn vehicles aggressiveness, percentage of the right turn and left vehicles in front of it etc.

Thus, the unsynchronized signal timing would be more efficient for the management of the traffic and the improvement of the transportation system.

8. Results

8.1. Comparison of synchronized and unsynchronized traffic signals

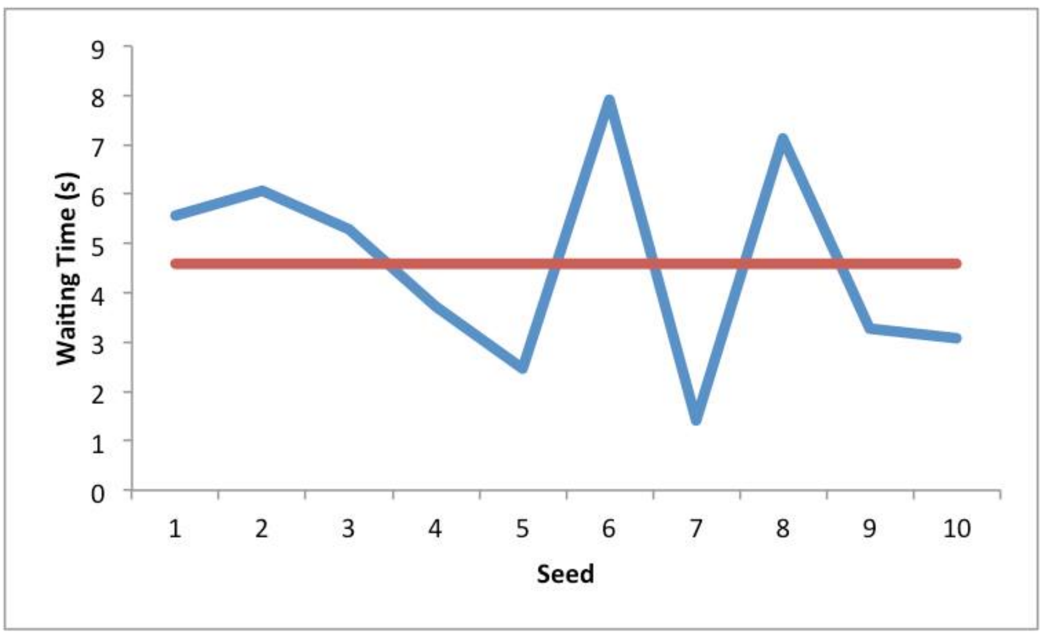

Figure 12: Average waiting time for different random seeds with unsynchronized signal timings

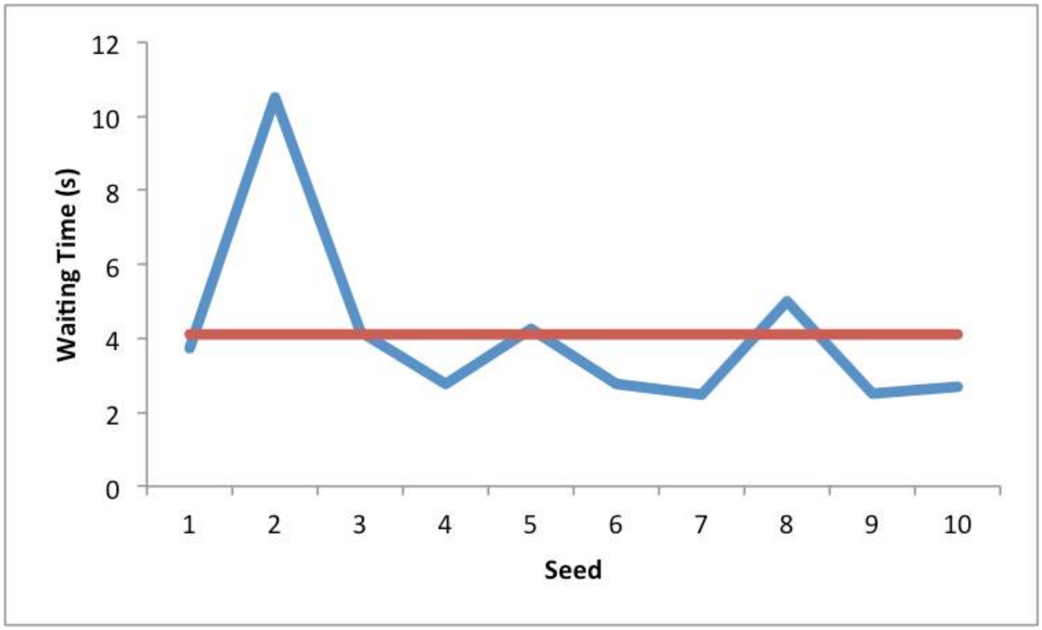

Figure 13: Average waiting time for different random seeds with synchronized signal timings

Average intersection waiting time for unsynchronized and synchronized traffic signal was observed to be 4.59 and 4.09 seconds. Making the traffic signals synchronized reduces the average waiting time by 29.67%. It was as expected that synchronized signals reduce the travel time, as we increase the offset time. The offset time for the synchronized signal simulation was set to be 17.1 seconds and it was found that increasing the offset time beyond this value led to an increase in waiting time, therefore 17.1 was chosen as the optimal offset time. This kind of behavior of the travel time distribution of synchronized data was also observed by another research conducted in [16].

Waiting time over the different simulation runs is changing drastically and its not sufficient to draw a valid conclusion from it. While comparing figure 5 and figure 6, it can be observed that the waiting time for the synchronized signal has a declining trend line and we could possibly observe a steady state in this data if the run time was increased to few hours.

8.2. Effect of changing mean interarrival times

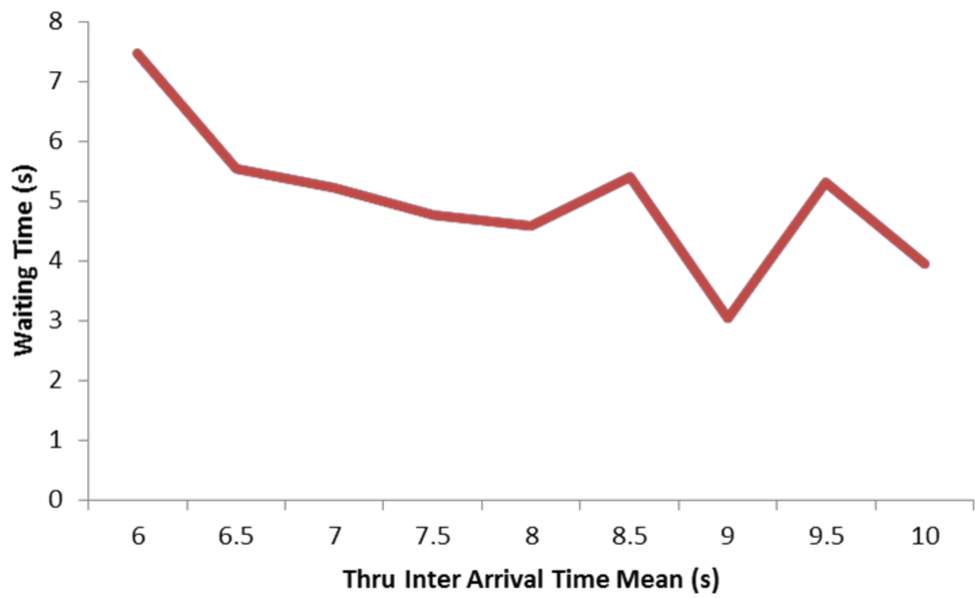

Figure 14: Variation in average waiting time with changes in mean through interarrival time

We had got a mean through interarrival time value by input modeling. In order to see the effect of changes in average waiting time with changes in mean through interarrival times, we ran the simulator with a few different values for the mean and found that, as expected, the average waiting time decreases as the interarrival time increases, since the density of the vehicles decreases (Figure 14).

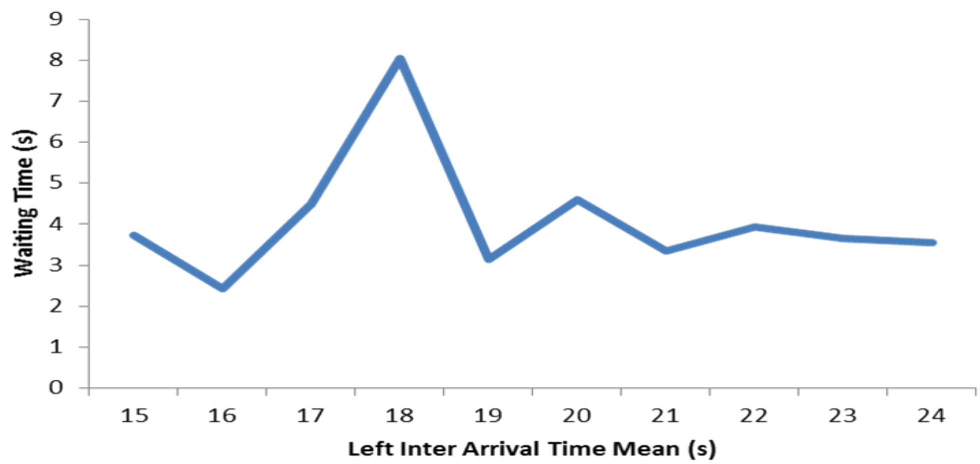

Figure 15: Variation in average waiting time with changes in mean left interarrival time

We had assumed a value of left inter arrival time based on the inter arrival time for southbound vehicles and the percentage of southbound vehicles making a left turn. Since this wasn't very reliable data, we decided to vary left inter arrival times around the assumed mean and it turned out that the average waiting time didn't vary much (Figure 15), thus validating our assumption.

8.3. Aggressive Behavior

We consider other factors that would actually influence the traveling time of the vehicles. For this part we have to consider the aggressive drivers behavior of the opposite direction. For our modeling, we ignore the aggressive driver in the North bound direction for the reason that the these aggressiveness that we consider would only decrease the traveling time, but for the aggressive driver that in the left turn directions, if they take the aggressive actions, they would block the intersection, which would change the state of the intersection to be occupied, and block the vehicle that is going through the intersection.

From the chart we could see that when we increase the aggressiveness of the vehicles that is making the left turn in the opposite direction. We could get the conclusion that the average travel time for the north bound traveling vehicles would be positively related to the aggressive behavior of the left turn vehicles. And the changes are huge as when the aggressiveness increase to

80%, the mean time increase to more than twice of the rate when set to 20%. And these results also agree with our daily experience.

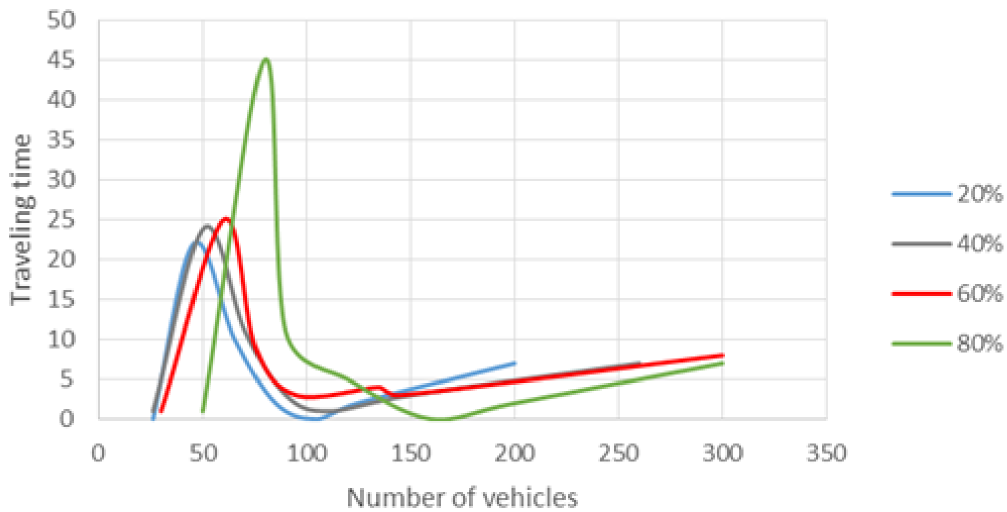

Figure 16: Aggressive driver behavior

8.4. The Changing Arrival Time for the north bound travel vehicle

We define the arrival interval time to be the variable and determine how the arrival interval would actually affect the travel time of the vehicles. A small value for the interval time means that there would be more cars traveling on the road, making the road more congested and thus would increase the traveling waiting time of the vehicles. From the graph above we could see the influence on the travel time . To be more specific , we defined these variables to be changing to model the rush hours traffic condition. In which case the road would actually congested with traffics.

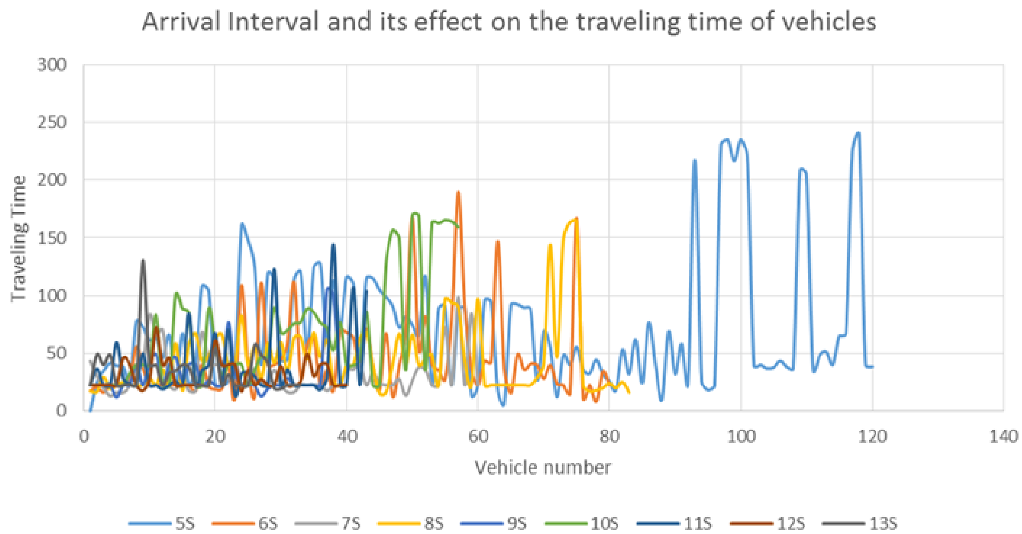

Figure 17: Aggressive driver behavior

9. Conclusion

Following were some of the salient points that we concluded from our results:

- Waiting time analysis indicated great variation across different simulations
- Synchronized signal gives lower waiting time than unsynchronized traffic signals.
- Waiting time shows a declining trend with increase in through inter arrival time. Waiting time remains relatively unaffected on changing left inter arrival time.
- A total of 159 vehicles exited the simulation system within 900 seconds time frame.

Analysis of the aggressive driver behavior is not conclusive, as the waiting time becomes constant after increasing the aggressive driver percentage by 5%.

References

[1] M. Pursula, "Simulation of traffic systems-an overview," *Journal of Geographic Information and Decision Analysis*, vol. 3, no. 1, pp. 1–8, 1999.

[2] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[3] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de physique I*, vol. 2, no. 12, pp. 2221–2229, 1992.

[4] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[5] N. Cetin, A. Burri, and K. Nagel, "A large-scale agent-based traffic microsimulation based on queue model," in *IN PROCEEDINGS OF SWISS TRANSPORT RESEARCH CONFERENCE (STRC), MONTE VERITA, CH*, Citeseer, 2003.

[6] D. W. Jones, "An empirical comparison of priority-queue and event-set implementations," *Communications of the ACM*, vol. 29, no. 4, pp. 300–311, 1986.

[7] R. Corthout, G. Flötteröd, F. Viti, and C. M. Tampère, "Non-unique flows in macroscopic first-order intersection models," *Transportation Research Part B: Methodological*, vol. 46, no. 3, pp. 343–359, 2012.

[8] U.S. Department of Transportation Federal Highway Administration, "Intersection safety issue briefs," 2009.

[9] G. Flötteröd and J. Rohde, "Operational macroscopic modeling of complex urban road intersections," *Transportation Research Part B: Methodological*, vol. 45, no. 6, pp. 903–922, 2011.

[10] J. A. Laval and C. F. Daganzo, "Lane-changing in traffic streams," *Transportation Research Part B: Methodological*, vol. 40, no. 3, pp. 251–264, 2006.

[11] P. Hidas, "Modelling vehicle interactions in microscopic simulation of merging and weaving," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 1, pp. 37–62, 2005.

[12] Q. Yang and H. N. Koutsopoulos, "A microscopic traffic simulator for evaluation of dynamic traffic management systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113–129, 1996.

[13] G. Duncan, "Paramics wide area microscopic simulation of att and traffic management," in *Proceedings of the 28th International Symposium on Automotive Technology and Automation (ISATA)*, 1995.

[14] J. Barceló, J. Casas, E. Codina, A. Fernandez, J. Ferrer, D. García, and R. Grau, "Petri: A parallel environment for a real-time traffic management and information system," in *Intelligent Transportation: Realizing the Future. Abstracts of the Third World Congress on Intelligent Transport Systems*, 1996.

[15] "Boost.Program_options." `http://www.boost.org/doc/libs/1_57_0/doc/html/program_options.html`. Accessed: 2015-02-20.

[16] J. A. Hillier and R. Rothery, "The synchronization of traffic signals for minimum delay," *Transportation Science*, vol. 1, no. 2, pp. 81–94, 1967.

[17] L. G. Birta and G. Arbez, *Modelling and simulation*. Springer, 2007.

[18] R. M. Fujimoto, *Parallel and distributed simulation systems*, vol. 300. Wiley New York, 2000.