Master's Thesis
Traffic light prediction for TomTom devices

Paul Krijger

July 7, 2013

Management summary

For car navigation, traffic lights lead to unpredictable delays. Using traffic light phase predictions we can improve the TomTom route planning and guidance. We use traffic light data to make (distant) future predictions for both statically and dynamically managed traffic lights. The data can come from 3rd parties, FCD or V2V data. We visualize these predictions in the car to give a speed advice for the driver to catch green light more often. This makes the trip more comfortable and reduces fuel consumption. These traffic light delay predictions also give better travel time estimations when routing through networks of traffic lights. This can lead to faster routes and green wave advices. At the same time, the invention is a natural way to divide the traffic better through a city and improve the entire network. It offers a new service for TomTom based on traffic prediction and is attractive for cities due to fewer vehicle stops at traffic light and smoother driving, resulting in less exhaust emissions.

The prediction model
As traffic lights make use of green, red and orange lights and drivers can still pass the intersection at an orange light, we model the orange light as green. So we see the traffic light as a cycle of green and red times. Nowadays, most traffic lights are dynamically managed. This means that the green and red times depend on queuing cars at the traffic light. Thus the green and red times can be modeled as random variables. The distributions of the green and red times highly depend on the time of the day. During rush hour the times will be much bigger and have less variation than during the night. As input for the prediction model we need a list of (historical) green and red times and a current state of the traffic light. The input can come from 3rd parties, FCD or V2V data. Note that if the traffic light is statically managed, the green and red times are fixed and the predictions will always be correct.

Applying the predictions
By probability calculations, we can give (distant) future predictions for the phase of the traffic light. First we visualize the probabilities by a green, red and orange time window. The green intervals indicate that the probability of green light is high and we try to lead the driver to this region to have maximum probability to catch green light. The red area indicates that the probability for red light is high and the orange intervals indicate that the phase is unpredictable. By using the distance from the vehicle to the traffic light, we convert the time window to a speed advice window. Now the driver can adjust the speed to catch green light more often. The speed advice makes the trip more comfortable and reduces fuel consumption because vehicles have to stop and restart less during their journey. Furthermore, experiments in Portland have shown that for dynamically managed traffic lights the predictions can be correct for 400 seconds on average.

Improve network predictions and find faster routes

We can also use traffic light predictions to give better estimations for travel times through a route with multiple traffic lights. For each traffic light we can calculate the expected waiting time until the next green light. Use (live) traffic data of TomTom to estimate the travel time between intersections. The combination of expected waiting times and travel times between the intersections gives the total expected travel time of the route. In an example with three traffic lights, we see that the current states of the traffic lights can make a difference of almost one minute for the total travel time. The analysis also indicates areas where the driver will likely have a green wave. These predictions can be done for every possible route through a network to find the fastest route. All possible routes can be found by the TomTom device.

Example in Assen, the Netherlands

In the north of Assen, there is a network of (very statically managed) traffic lights where a green wave has been implemented. If one has to go from the center of Assen toward Groningen, the driver has two possible routes (left and right). The green wave has been implemented for both the left and the right route. The left route contains four traffic lights and the right route only two traffic lights. The right route is on average 47 seconds faster and will always be advised by the TomTom device. However, if we apply the traffic lights phase predictions, we conclude that in 15.7% of the cases the left route will be faster (with 10.5 seconds on average). If we allow more variation in traffic speeds between intersections, the probability that the left route will be faster is still 10.7% (with 11.2 seconds on average). Note that the percentage decreases, because more traffic lights will increase the uncertainty.

Example in Portland, USA

In the center of Portland, there is a network with various situations. The network has both dynamically and statically managed traffic lights. Also a tram crosses the network, which causes large variations in green times for some traffic lights. In the network nine possible routes (from the lower left to the upper right intersection) have been analyzed. The traffic light phase predictions show that for the upper left route, a green wave has been implemented and this route has the lowest average travel time. Also the TomTom devices will always advise this route. But the traffic light predictions can find a faster route in 16.7% of the cases (with 14.5 seconds on average).

The improved route advices based on live traffic light data, can also lead to a natural way for dividing traffic better through a city. Based on the predictions, for example some cars will be led over a left route and other cars over the right route. These traffic management advices can also improve the entire network of a city. So the project also fits in Praktijk Proef Amsterdam and the roadmap of the Amsterdam Group.

Future research

At this moment, the prediction models already give good results. But there is room for improvement.

It is still essential to add queuing theory to the prediction model because queues at traffic lights lead to extra waiting times, which can let the driver miss a green light. A queuing model should be developed which uses HD traffic flow and traffic light data.

If the traffic light phase prediction will be tested in TomTom devices, several mathematical decisions have to be made for a good consideration between performance of the predictions, calculation power and amount of data that have to be stored/sent.

In the thesis it is mentioned that traffic light data can come from FCD or V2V communication. At this moment we are able to determine the cycle for a statically managed traffic light. This is successfully implemented for an intersection in Portland, but it needs more testing and the speed of the method can be improved. For dynamically managed traffic lights, the model still needs to be adjusted to give approximations which are accurate enough to give reliable traffic light phase predictions.

For public interest, it will be useful to measure how traffic light predictions can improve the network and can save fuel and result in lower emissions.

Acknowledgments

After this master's project Industrial and Applied Mathematics (at the TU/e) of nine months, there are several people who I want to thank for their help and sociability to accomplish my master's thesis. I had a great period during this project and the topic has motivated me to gain this result. Hopefully in the future, the research described in this thesis will be used in car navigation systems.

I want to thank Onno Boxma and Marko Boon from the TU/e for their guidance. Regardless of their full agendas, it was always possible to make time for me. They have pushed me often toward the right direction, which led to the results we wanted to achieve. Also their help with writing and improving this thesis is gratefully acknowledged. Their endless enthusiasm for the project was a very inspiring factor for me. Hopefully, our mathematical ideas will help improving the traffic flows more often in the future. For completing our assessment committee and being very flexible for our planning, I want to thank Cor Hurkens.

During this project, I had a very useful experience with my internship at TomTom. Firstly the TU/e Smart Mobility program has given me the possibility to get in contact with TomTom. I want to thank Stefan Bollars and Ben Rutten for giving me the opportunity to prove myself with the internship and providing me an educational experience. For my internship at TomTom, I want to thank the Damascus team (Brendan Rousseau, Fabian Broekhof, Jordi Betting, Jurgen Rijsdijk, Timo van de Put and Erik van der Velden) for their guidance. The way they have taken me in their team, has given me the feeling to be a real TomTom employee. Also their input and TomTom knowledge has provided me useful ideas for my research. Furthermore the help with analyzing the data and reviewing the java codes has improved my programming skills.

Essential for our research, is the real life traffic light data we received. We used the data to test and modify our traffic light predictions. Therefore I want to thank PeekTraffic and Green Driver for providing us with the data. Also the engineers at these companies (PeekTraffic: Robin Blokpoel and Green Driver: Kevin Scavezze) helped me with understanding the data and provided me with additional information about the traffic lights.

During the master's thesis, I have shared a room at the university with fellow students. They have given me company for the days I have spent at the university. Britt Mathijsen, Jorn van der Pol and Reint den Toonder, many thanks for the good times and coffee breaks. I wish you the best in your careers. For the last several years, I also want to thank all fellow students for the good times inside and outside the classes.

Last but not least, I want to thank my parents for their support during all years at college. Whether or not they understood what I was doing, my parents always were interested in my courses and they motivated me to keep studying.

Paul Krijger
June 2013

Contents

1 Introduction

When people are talking about traffic, they mostly complain about congestions, too many red lights and other inconveniences during their journey. We all want to go as fast as possible from A to B. Moreover, the journey should be comfortable and economical. TomTom is a company that can help motorists to achieve this goal.

1.1 TomTom

TomTom is a globally known company that develops car navigation devices. The company was founded in 1991 and firstly TomTom developed business-to-business applications for mobile devices, including a route planner. In 2001 GPS satellite readings became accurate enough for TomTom to start their car navigation program. In the following years they became a market leading company in car navigation and now they can cover for example 99.9% of the roads in Europe and North America.

TomTom is headquartered in Amsterdam, has around 3500 employees and sells its products in over 40 countries. The company also has offices in Eindhoven (research department), Gent, Lodz, Harsum, London, Massachusetts, Edinburgh, Berlin, Leipzig and Taiwan. TomTom is listed at Euronext since May 2005.

To stay a market leader, TomTom keeps innovating its products. One of the new innovations is real live traffic information, which can be used to predict congestions, road work and traffic lights.

1.2 Glossary

In the thesis, we will use terms which are well known in traffic (light) management and by TomTom. Below we list the corresponding definitions:

Approach
: Part of the road which leads to an intersection.

BO
: Back Office, the TomTom server.

Conflicting group
: Set of signal groups that cannot have green simultaneously (for safety reasons).

Cycle plan
: Prescribed sequence for the phases of the signalized intersection. In case of a statically managed traffic light, also the sizes of the green and red times are specified.

Cycle time of traffic light
: The time during which all lights of approaches had the right to turn green. In our research often modeled by the sum of the green and red times.

Dynamically managed traffic light
: Traffic lights which react to vehicle demands (using induction loops). So the sizes of the green and red times are variable.

FCD
: Floating Car Data, live information that TomTom devices send.

Green wave
: Series of traffic lights which are specified such that vehicles can continuously drive through successive green lights.

HD Traffic
: High Definition Traffic, real time traffic information provided by TomTom.

Induction loops
: Detection system in the road that registers vehicles.

Intersection
: A set of roads that meet or cross and have a common crossing area. The intersection is called signalized if it is managed by traffic lights.

Light Traffic
: The traffic intensity is very low and the traffic light can easily handle all the vehicles.

Phase
: Signal groups which have green (simultaneously).

Profile of traffic light
: The profile of the traffic light describes how the traffic light behaves. In our research modeled by the distributions of the green and red times.

Signal group
: Set of approaches which are controlled by the same traffic light.

State
: Amount of time the traffic light is red or green (note that the state is more detailed than phase).

Statically managed traffic light

Traffic lights which do not react to vehicle demands. So the green and red times are constant.

TomTom trace

List of GPS coordinates where a vehicle has driven, the trace can be obtained from FCD.

V2V

Vehicle to Vehicle, method to communicate between vehicles. Also information that TomTom can provide.

1.3 Traffic lights

To regulate traffic on intersections often traffic lights are used. These signals improve the safety and reduce congestion. Traffic lights have cycle times which contain a green, red and orange period. During green light vehicles may pass the traffic light and during red light this is forbidden. The rules for the orange light (sometimes called yellow light) are different for some countries. For example in the Netherlands and in the United States of America, the orange light follows after the green light and indicates that the light will switch to red. During orange light it is still allowed to cross the intersection, but if possible the driver should stop the car. In Germany for example, the orange light follows after the red light and indicates that the traffic light will turn green. For each intersection a control plan is specified, that describes in which sequence the approaches can receive a green light.

We distinguish three types of traffic light control: statically managed, dynamically managed and a wireless connected control system.

In statically managed traffic lights the cycle times are constant (in more detail: the red, green and orange times are deterministic). In the Netherlands this type of traffic lights is becoming rare, but in other countries statically managed traffic lights are used more frequently.

Nowadays most signals in the Netherlands are dynamically managed, which means that the green and red times are influenced by the number of queueing cars at the traffic light. The green and red times always have a minimum and maximum value. When the traffic intensity increases, most traffic lights are specified such that the green times will increase and the system can serve more cars.

A wireless connected control system can communicate with wireless devices in vehicles. Such a system can react better to upcoming traffic and also send its next red and/or green time. These kind of traffic lights are very modern and still form a minority.

1.4 Goal

For car navigation, traffic lights will cause many errors for travel time predictions. At this moment only *average* delays at traffic lights are used in the route planner. With use of traffic light predictions, TomTom wants to improve its route planning algorithms and visualize the predictions on its devices. So we want to construct a model which TomTom devices can use for their navigation system. The amount of required data and calculations in the prediction model should also be minimized. ***TomTom wants to make an estimation module to predict the green/red time window and visualize speed advices in the car. Also better routes through networks of traffic lights can be advised using these estimations.***

1.5 Literature study

If we want to make predictions on traffic lights, we first have to know how they work. To understand traffic lights and the way we can model them, we use existing literature. We first give a summary how traffic lights are designed and controlled in the Netherlands. In science, most research has been done for statically managed traffic lights (also called the fixed-cycle traffic light). To study delays at dynamically managed traffic lights, we can make use of so-called polling systems.

1.5.1 Traffic light management in the Netherlands

In 2004, around 5400 intersections with traffic lights were located in the Netherlands. The traffic lights are mostly designed by traffic engineers. During the design process often more parties are involved (like police and bicycle unions). The traffic engineer will analyze the traffic situation at the intersection and specifies the cycle plan of the traffic lights to optimize the traffic flow. The decision making steps are described in [7]. That book further explains what the Dutch standards are for approach numbers at an intersection. The book also provides information about traffic intensities and departure processes at traffic lights, which can be used to analyze whether the intersection can handle the traffic flow.

We are mostly interested in the length of the green and red times at dynamically managed traffic lights. The book describes different phases within the green and red times, which depend on arriving traffic. The red light has mainly two phases: *Red before request* and *Red after request*. As long as no requests have been made, the lights stay in the first phase. When a vehicle arrives at the traffic light (and is detected by an induction loop), the approach will get green according to the specified cycle plan of the intersection. The green light first starts with a *Fixed green time*. We regard this time as the minimum possible green time. In the Netherlands, this time is mostly six seconds for vehicles. After the fixed minimal green time, the system switches to the *First vehicle dependent green* phase. The light remains in this phase while no requests have been made on conflicting approaches and this time is not exceeding a maximum specified time. If a conflicting approach has done a request, the green light switches to the *Second vehicle dependent green* phase. This phase is used to hold green for upcoming groups of vehicles, which also has a maximum specified time. So if the traffic intensity is very high (like during rush hour), we expect that these maximum specified times will occur more often and that makes the traffic light more predictable.

The traffic engineer may decide to implement a structural green wave for a network of traffic lights. This is possible if the traffic lights are statically managed and have the same cycle length. Using the expected travel time between the intersections, the traffic engineer can calculate when the green times should occur in the cycle. At dynamically managed traffic lights, green waves are more difficult to implement.

1.5.2 Average delay at traffic lights

At this moment navigation devices can only use *average* delays at traffic lights. This depends on the time of the day and live traffic estimations. TomTom can determine the average delay by analyzing the traces from its devices. In the literature, much mathematical research has been done on the average delay at a statically managed traffic light. The first formula yielding a good approximation was given in 1958 by Webster [6]. First define the following variables:

$G :=$ green time of traffic light,
$R :=$ red time of traffic light,
$C :=$ cycle time of traffic light ($C = G + R$),
$\lambda :=$ arrival rate,
$\mu :=$ departure rate,
$\rho :=$ occupation rate of traffic light ($\rho = \frac{\lambda}{\mu}$).

The average vehicle delay at a traffic light can be approximated by:

$$d = \frac{R^2}{2C(1-\rho)} + \frac{\rho C^2}{2G(\mu G - \lambda C)} - 0.65 \left(\frac{C}{\lambda^2}\right)^{1/3} \left(\frac{\lambda C}{\mu G}\right)^{2+5G/C}. \tag{1}$$

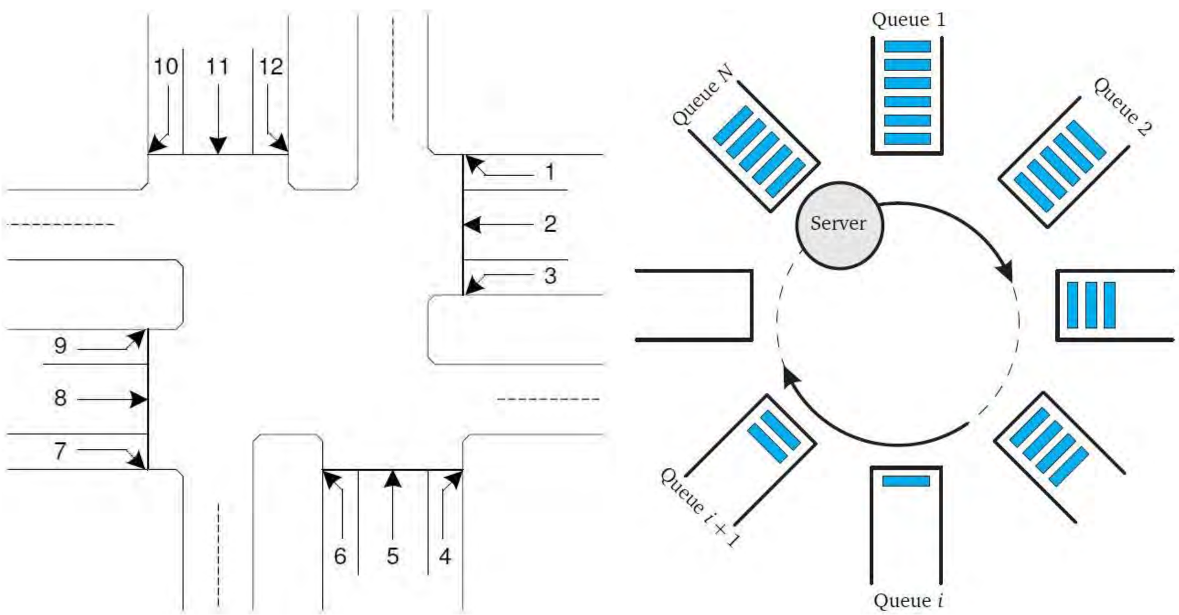
More research about the average delay at statically managed traffic lights has been done in the Master's thesis of van den Broek [2], where also formulas by Miller and Newell are studied. These are better approximations which have some additional terms.

Van Leeuwaarden [3] has derived distributions for the vehicle delay and queue length of the fixed-cycle traffic light. These distributions can be useful to analyze variations of the vehicle delay and queue length.

Van den Broek also optimized statically managed traffic lights. The goal was to determine a cycle plan which minimizes the weighted average delay. The average delay can be weighted if some approaches have a higher priority. To analyze the average delay at a dynamically managed traffic light, a polling model can be used.

1.5.3 Polling systems

In mathematics a signalized traffic intersection can be modeled as a polling system, consisting of multiple queues and a single server visiting each of these queues in some order. Note that some flows can be served simultaneously if they are not conflicting, which is an extension of the basic polling model. The model can be visualized as in Figure 1:



(a) flow directions in an intersection (b) polling system

Figure 1: The polling model

Here the numbers of the traffic flows are according to Dutch standards as described in [7]. We assume that the arrival process of the cars is stochastic and independent (normally assumed to be Poisson).

Boon wrote his thesis on Polling models [1], with some explanations on applications for traffic intersections. Mainly the influence of control policies and customer behavior to waiting times and queue lengths have been studied. The control policy can be exhaustive (the queue gets served until it is empty) or k-limited (at most k customers will be served during green light). For each queue, the distributions of the waiting time and queue length have been derived. For the traffic light case, Boon has analyzed the performance of the polling system in Heavy Traffic and Light Traffic (the two extreme cases in traffic intensities). Although the research focuses on the limiting behavior of the system, we can see how the control policy and parameters influence the performance of the traffic light. Also we can see what the effect of the (independent) arrival distribution is to the system.

1.6 Overview of the thesis

This thesis considers several aspects of traffic light predictions. In Section 2 we present the mathematical model for traffic light phase prediction. The predictions are based on probability theory. A network of statically managed traffic lights is analyzed in Section 3. In this section we will try to find faster routes for a network in Assen. In Section 4 we use TomTom traces to determine the cycle plan and current state of a statically managed traffic light. For this goal we present an algorithm which is tested by simulations and real implementation in Portland. Dynamically managed traffic lights are studied in Section 5. For this topic, real traffic light information in Helmond and Portland is used. In this section we analyze standalone traffic lights. In Section 6 we will try to reduce the travel time in a network of dynamically managed traffic lights. In Section 7 the end-to-end design is presented for the TomTom implementation. We suggest future research in Section 8.

2 The model for traffic light phase prediction

If a car is heading toward a traffic light two important questions are: (i) whether or not the light is green at the car's arrival and (ii) the amount of time the car has to wait until the light turns green. If we can answer these questions, it is possible to give a speed advice for the driver to catch a green light. We model the traffic light as a cycle with one green time and one red time, where the times are independent random variables. Our research is based on traffic lights in the Netherlands and in the United States of America. So in our situation the car is allowed to cross the intersection at an orange light and we model the orange light as part of the green period (in other situations like Germany, the orange light can be modeled as part of the red period). ***The traffic light model makes it possible to calculate the probability of arriving when the light is green and the expected waiting time until the next green time.***

In this section we first analyze the easiest case where only the green time is random and the red time is deterministic. Later we give the prediction equations where both green and red times are stochastic. For both cases examples are given for the normal distribution.

2.1 Deterministic red time

The first way to model the cycle is to take all randomness in the green time, thus to let the red time be deterministic. Let the current time be 0, then we want to give phase predictions for time $t > 0$. Assume that at $t = 0$, the state of the traffic light is exactly known. The state is the amount of time the light has been green or red, let this time be a. First we define the variables needed for the model. Let the start of the first occurring green time be G_{1s}. Further define:

G_{is} := start of ith green time, $i = 1, 2, ...$
G_j is the (stochastic) green time, $j = 1, 2, ...$
R is the (deterministic) red time,
a is the (deterministic) elapsed green/red time at $t = 0$,

$$r := \begin{cases} 0 & \text{if the light is green at } t = 0, \\ 1 & \text{if the light is red at } t = 0, \end{cases}$$

f_i probability density of $\displaystyle\sum_{j=1}^{i} G_j$, $i = 1, 2, ...$
$p(t) = \mathbb{P}(\text{light is red at time } t)$, $t \geq 0$.

If the light is green at $t = 0$, the sequence of consecutive cycles can be visualized as in Figure 2:

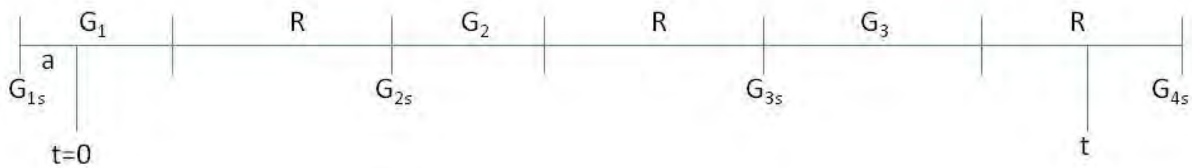


Figure 2: Model of the green/red cycles

When the light is green at $t = 0$ like in Figure 2, the distribution of G_1 is different from that of other green times because of the given elapsed green time. In other words, the a can influence the residual green time. We will consider the distribution of the entire green time, so if $r = 0$ the distribution of G_1 is conditional on a . See Section 2.4 for more details about the distribution of G_j.

The start of the ith green time is distributed as follows:

$$G_{is} \quad \sim \quad \sum_{j=1}^{i-1} G_j + (i-1+r)R - a.$$

To compute the probability that the light is red at t, we sum over all possible cycles:

$$\begin{aligned}
p(t) \quad &= \quad \mathbb{P}(\text{light is red at time } t) \qquad (2)\\
&= \quad \mathbb{P}(G_{1s} > t) + \sum_{i=1}^{\infty} \mathbb{P}(G_{is} < t - G_i \text{ and } G_{(i+1)s} > t)\\
&= \quad 1 - \mathbb{P}\left(0 < t - rR + a\right) +\\
&\quad\quad \sum_{i=1}^{\infty} \left\{ \mathbb{P}\left(\sum_{j=1}^{i} G_j < t - (i-1+r)R + a \right) - \mathbb{P}\left(\sum_{j=1}^{i} G_j < t - (i+r)R + a \right) \right\}\\
&= \quad \mathbf{1}\{t + a \le rR\} +\\
&\quad\quad \sum_{i=1}^{\infty} \left\{ \mathbb{P}\left(\sum_{j=1}^{i} G_j < t - (i-1+r)R + a \right) - \mathbb{P}\left(\sum_{j=1}^{i} G_j < t - (i+r)R + a \right) \right\}.
\end{aligned}$$

We can compute the probability that the next starting green time is G_{is} and the light is red at t:

$$\begin{aligned}\mathbb{P}(G_{is} < t + R \text{ and } G_{is} > t) &= \mathbb{P}\left(\sum_{j=1}^{i-1} G_j < t - (i-2+r)R + a\right) - \\ &\quad \mathbb{P}\left(\sum_{j=1}^{i-1} G_j < t - (i-1+r)R + a\right).\end{aligned}$$

Let t be given. The next expression can be explained as the expected time until G_{is}, given that G_{is} is the next following starting green time and the light is red at t. Define $WG_i(t)$ as follows:

$$WG_i(t) := \begin{cases} \mathbb{E}[G_{is} - t | G_{is} < t + R \text{ and } G_{is} > t] & \text{if } \mathbb{P}(G_{is} < t + R \text{ and } G_{is} > t) > 0, \\ 0 & \text{if } \mathbb{P}(G_{is} < t + R \text{ and } G_{is} > t) = 0, \end{cases} \tag{3}$$

where:

$$\mathbb{E}[G_{is} - t | G_{is} < t + R \text{ and } G_{is} > t] = \begin{cases} R - a - t & \text{if } i = 1, \\ \dfrac{\displaystyle\int_{t-(i-1+r)R+a}^{t-(i-2+r)R+a} x f_{i-1}(x) dx}{\displaystyle\int_{t-(i-1+r)R+a}^{t-(i-2+r)R+a} f_{i-1}(x) dx} + (i-1+r)R - a - t & \text{if } i > 1. \end{cases} \tag{4}$$

In our model we assume that the queue at the traffic light is empty. So if we arrive at the traffic light when the light is green, we can immediately cross the intersection. If the light is red at arrival, we can continue the trip when the light turns green. This leads to the following expected waiting time until green light, given that the car arrives at a red light:

$$\mathbb{E}[W(t) | \text{red light at arrival}] = \sum_{i=1}^{\infty} WG_i(t) \frac{\mathbb{P}(G_{is} < t + R \text{ and } G_{is} > t)}{p(t)}. \tag{5}$$

In theory we must sum from 1 to ∞, but in practice we only sum over a few terms to retrieve a good approximation. For the implementation it will be necessary to bound the sum, else we will divide by numbers which are nearly zero. See Section 7 for more details. This leads to the following equation for the expected waiting time until the next green light (if a car arrives during green time it waits for 0 seconds):

$$\begin{aligned}
\mathbb{E}[W(t)] &= \mathbb{E}[W(t)|\text{red light at arrival}] \cdot p(t) + \mathbb{E}[W(t)|\text{green light at arrival}] \cdot (1 - p(t)) \\
&= \mathbb{E}[W(t)|\text{red light at arrival}] \cdot p(t) \\
&= \sum_{i=1}^{\infty} WG_i(t) \cdot \mathbb{P}(G_{is} < t + R \text{ and } G_{is} > t). \qquad (6)
\end{aligned}$$

2.1.1 Special case: green time normally distributed

A reasonable distribution for the green time could be a normal distribution, because as described in [7] the green and red times consist of multiple vehicle dependent phases. So we can consider the green time as sum of random variables. Also in the model we sum over multiple cycles and according to the central limit theorem, the sum of independent random variables converges to a normal distribution. The main advantage of this choice is that the sum of independent normal random variables is again normally distributed. Only if the light is green at $t = 0$, G_1 will not be normally distributed. If a is sufficiently small, G_1 is approximately normally distributed. Now let $G_j \sim \mathcal{N}(\mu_G, \sigma_G^2)$, with μ_G the mean green time and σ_G^2 the variance of the green time. The sum of i independent green times will be approximately distributed as follows:

$$\sum_{j=1}^{i} G_j \sim \mathcal{N}(i\mu_G, i\sigma_G^2).$$

Using these summations we can rewrite the probability for a red light:

$$p(t) \approx \frac{1}{2} \sum_{i=0}^{\infty} \left\{ \operatorname{erf}\left(\frac{t - (i - 1 + r)R + a - i\mu_G}{\sqrt{2i\sigma_G^2}} \right) - \operatorname{erf}\left(\frac{t - (i + r)R + a - i\mu_G}{\sqrt{2i\sigma_G^2}} \right) \right\},$$

with $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$.

2.2 Stochastic red time

A more realistic way to model the traffic light is to also assume that the red time is a random variable. The definitions for the model are similar as before:

G_{is} := start of ith green time, $i = 1, 2, ...$
G_j is the stochastic green time, $j = 1, 2, ...$
R_j is the stochastic red time, $j = 1, 2, ...$
C_j is the stochastic cycle time, $j = 1, 2, ...$
a is the deterministic elapsed green/red time at $t = 0$,
$r := \begin{cases} 0 & \text{if light is green at } t = 0, \\ 1 & \text{if light is red at } t = 0, \end{cases}$
g_i probability density of $\displaystyle\sum_{j=1}^{i} G_j + \sum_{j=1}^{i+r} R_j$, $i = 1, 2, ...$
$p(t) = \mathbb{P}(\text{light red at } t)$.

If the light is green at $t = 0$, the cycle can be visualized as in Figure 3:

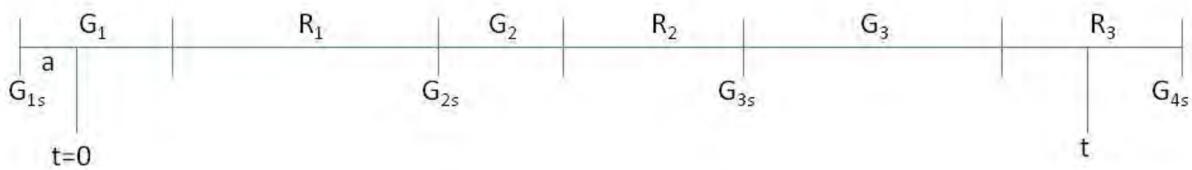


Figure 3: Model of the green/red cycle

Depending on the starting state of the traffic light, the distribution of G_1 or R_1 will be different. See Section 2.4 for more details. The following equations can be derived similarly as in Section 2.1:

$$G_{is} \sim \sum_{j=1}^{i-1} G_j + \sum_{j=1}^{i-1+r} R_j - a = \sum_{j=1}^{i-1} C_j + rR_i - a,$$

$$\mathbb{E}[W|\text{red light at arrival}] = \sum_{i=1}^{\infty} \mathbb{E}[G_{is} - t | G_{is} < t + R_i \text{ and } G_{is} > t] \frac{\mathbb{P}(G_{is} < t + R_i \text{ and } G_{is} > t)}{p(t)}, \quad (7)$$

$$\begin{aligned}
\mathbb{P}(G_{is} < t + R_j \text{ and } G_{is} > t) &= \mathbb{P}\left(\sum_{j=1}^{i-1} G_j + \sum_{j=1}^{i-2+r} R_j < t + a\right) - \mathbb{P}\left(\sum_{j=1}^{i-1} G_j + \sum_{j=1}^{i-1+r} R_j < t + a\right) \\
&= \mathbb{P}\left(\sum_{j=1}^{i-1} C_j - (1-r)R_i < t + a\right) - \mathbb{P}\left(\sum_{j=1}^{i-1} C_j + rR_i < t + a\right),
\end{aligned}$$

$$\begin{aligned}
p(t) = &= \mathbb{P}(G_{1s} > t) + \sum_{i=1}^{\infty} \mathbb{P}(G_{is} < t - G_i \text{ and } G_{(i+1)s} > t) \\
&= 1 - \mathbb{P}\left(rR_1 < t + a\right) + \\
&\quad \sum_{i=1}^{\infty} \left\{ \mathbb{P}\left(\sum_{j=1}^{i} G_j + \sum_{j=1}^{i-1+r} R_j < t + a\right) - \mathbb{P}\left(\sum_{j=1}^{i} G_j + \sum_{j=1}^{i+r} R_j < t + a\right) \right\} \\
&= 1 - \mathbb{P}\left(rR_1 < t + a\right) + \\
&\quad \sum_{i=1}^{\infty} \left\{ \mathbb{P}\left(\sum_{j=1}^{i} C_j - (1-r)R_i < t + a\right) - \mathbb{P}\left(\sum_{j=1}^{i} C_j + rR_i < t + a\right) \right\}. \qquad (8)
\end{aligned}$$

Let t be given. The next expression can be explained as the expected time until G_{is}, given that G_{is} is the next following starting green time and the light is red at t. Define $WG_i(t)$ as follows:

$$WG_i(t) := \begin{cases} \mathbb{E}[G_{is} - t | G_{is} < t + R_i \text{ and } G_{is} > t] & \text{if } \mathbb{P}(G_{is} < t + R_i \text{ and } G_{is} > t) > 0, \\ 0 & \text{if } \mathbb{P}(G_{is} < t + R_i \text{ and } G_{is} > t) = 0, \end{cases} \qquad (9)$$

where the expected waiting time until next green light can be approximated by using the expected value of the red time in the boundaries of both integrals:

$$\mathbb{E}[G_{is} - t | G_{is} < t + R_i \text{ and } G_{is} > t] \approx \begin{cases} \mathbb{E}[R_1] - a - t & \text{if } i = 1, \\ \dfrac{\displaystyle\int_{t+a}^{t+a+\mathbb{E}[R_j]} x g_{i-1}(x) dx}{\displaystyle\int_{t+a}^{t+a+\mathbb{E}[R_j]} g_{i-1}(x) dx} - a - t & \text{if } i > 1. \end{cases} \qquad (10)$$

Note that Equation (10) is an approximation because we use the expected value of the red time in the boundaries of both integrals. If we want to calculate the expected waiting time until next green light precisely, both integrals will become double integrals (see Appendix A). But our model should be implemented in TomTom devices, so we want to calculate the predictions faster. That is why we will present approximations more often.

Like in the deterministic case, the expected waiting time until the next green light can be calculated as follows:

$$\mathbb{E}[W(t)] = \sum_{i=1}^{\infty} WG_i(t) \cdot \mathbb{P}(G_{is} < t + R_i \text{ and } G_{is} > t). \tag{11}$$

2.3 Limiting behavior

An alternating renewal process is a process that alternates between independent stochastic on and off times. In the case of the traffic light the green lights are the on times and the red lights the off times, which are also independent in our model. So for the limiting behavior we can use the results as in Ross [4]. According to the alternating renewal theory we have for the limiting probability for red light:

$$\lim_{t\to+\infty} p(t) = \lim_{t\to+\infty} \mathbb{P}(\text{light red at } t) = \frac{\mathbb{E}[R_j]}{\mathbb{E}[R_j] + \mathbb{E}[G_j]}.$$

For the probability for green light we obviously have:

$$\lim_{t\to+\infty} \mathbb{P}(\text{light green at } t) = 1 - \frac{\mathbb{E}[R_j]}{\mathbb{E}[R_j] + \mathbb{E}[G_j]} = \frac{\mathbb{E}[G_j]}{\mathbb{E}[R_j] + \mathbb{E}[G_j]}. \tag{12}$$

The expected waiting time until next green light has the following limit:

$$\begin{aligned}
\lim_{t\to+\infty} \mathbb{E}[W(t)] &= \left(\lim_{t\to+\infty} p(t)\right) \frac{\mathbb{E}[R_j^2]}{2\mathbb{E}[R_j]} + \left(\lim_{t\to+\infty} (1 - p(t))\right) 0 \\
&= \frac{\mathbb{E}[R_j]}{\mathbb{E}[R_j] + \mathbb{E}[G_j]} \frac{\mathbb{E}[R_j^2]}{2\mathbb{E}[R_j]} = \frac{\mathbb{E}[R_j^2]}{2(\mathbb{E}[R_j] + \mathbb{E}[G_j])}. \qquad (13)
\end{aligned}$$

In case that G_j and R_j are normally distributed, we have:

$$\lim_{t\to+\infty} \mathbb{E}[W(t)] = \frac{\mu_R^2 + \sigma_R^2}{2(\mu_R + \mu_G)}.$$

Note that this limit is similar to Webster's equation (1), where no other cars arrive and the queue is empty ($\lambda = 0 \Rightarrow \rho = 0$).

2.4 Other distributions for G_1 or R_1

If the traffic light is green at $t = 0$, the elapsed time a will influence the remaining green time. Because the first green time must be larger than a, we get a conditional probability. Let $F_{G_j}(x)$ be the cumulative distribution function of G_j, which can be written as:

$$F_{G_j}(x) := \begin{cases} \mathbb{P}(G_j \leq x) & \text{for } j > 1, \\ \mathbb{P}(G_1 \leq x | G_1 > a) = \frac{\mathbb{P}(a < G_1 \leq x)}{1 - \mathbb{P}(G_1 < a)} & \text{for } j = 1. \end{cases}$$

For the case that the light is red at $t = 0$, the distribution of the first red time will be different. The cumulative distribution function of R_j can be calculated in the same way:

$$F_{R_j}(x) := \begin{cases} \mathbb{P}(R_j \leq x) & \text{for } j > 1, \\ \mathbb{P}(R_1 \leq x | R_1 > a) = \frac{\mathbb{P}(a < R_1 \leq x)}{1 - \mathbb{P}(R_1 < a)} & \text{for } j = 1. \end{cases}$$

2.5 Examples

To study the behavior of the equations, we study some examples in this section. Because the normal distribution is easily implementable, we use this example to illustrate the behavior. The implementation issues are discussed in Section 7.

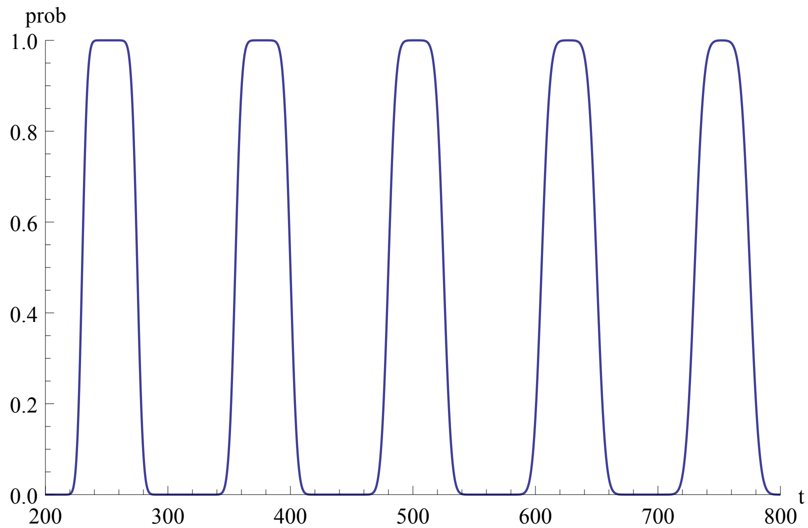
2.5.1 Example 1

Let the green and red times be normally distributed and take the following input for the model:

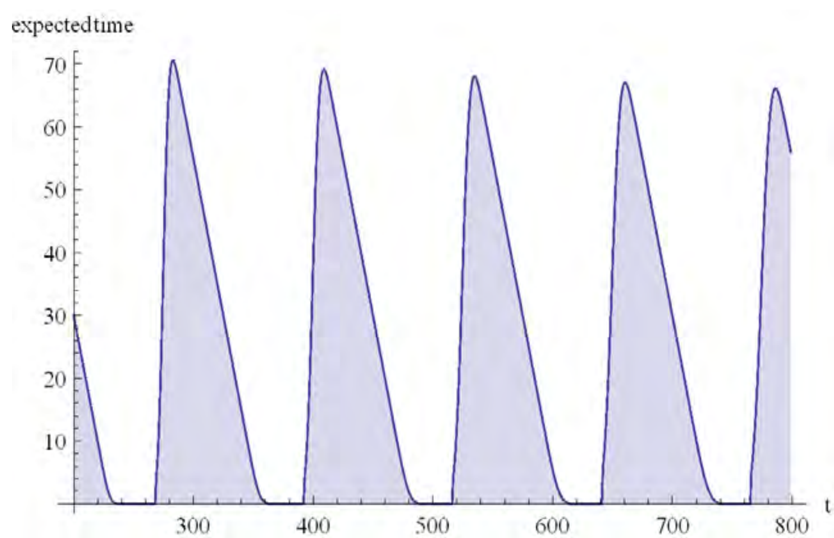
$a = 20$,
$\mu_G = 45$,
$\sigma_G = \frac{5}{3}$,
$\mu_R = 80$,
$\sigma_R = \frac{5}{3}$,
$r = 0$.

With these choices we can say that 99.7% of the random green times will have values between 40 and 50. Also a is small enough compared to the average green time, such that G_1 has nearly the same probability distribution as other green times. In Figure 4 we can see $p(t)$ and $\mathbb{E}[W(t)]$ for $200 \leq t \leq 800$.

In the figure the behavior of the function is harmonic. This is logical because we have a repeating red/green cycle. Moreover when $p(t)$ decreases, $\mathbb{E}[W(t)]$ will increase rapidly. This is caused by the cars that arrive just too late and have to wait almost the entire red time. Then a period of high probability for red light follows where $\mathbb{E}[W(t)]$ decreases linearly.



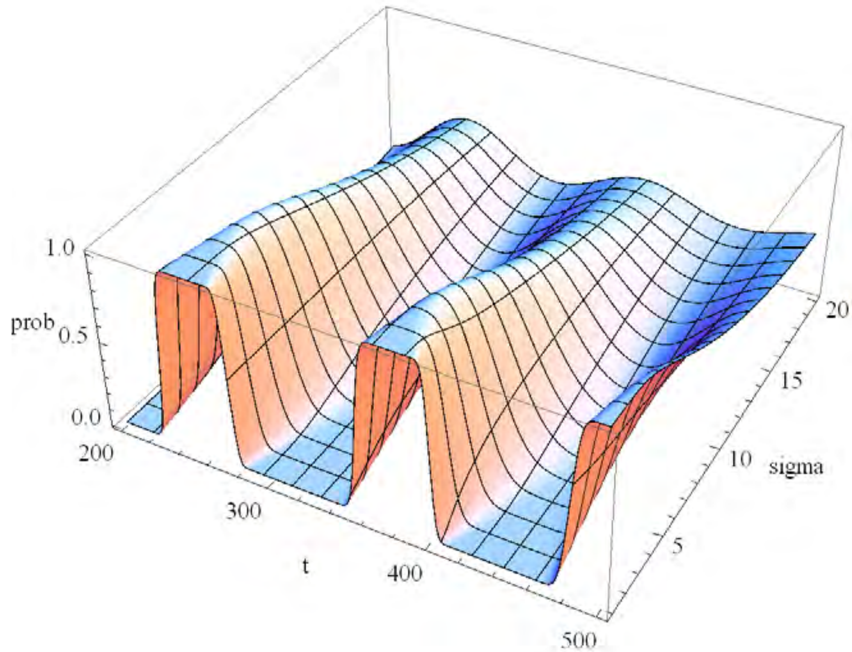
(a) $p(t)$ at time t



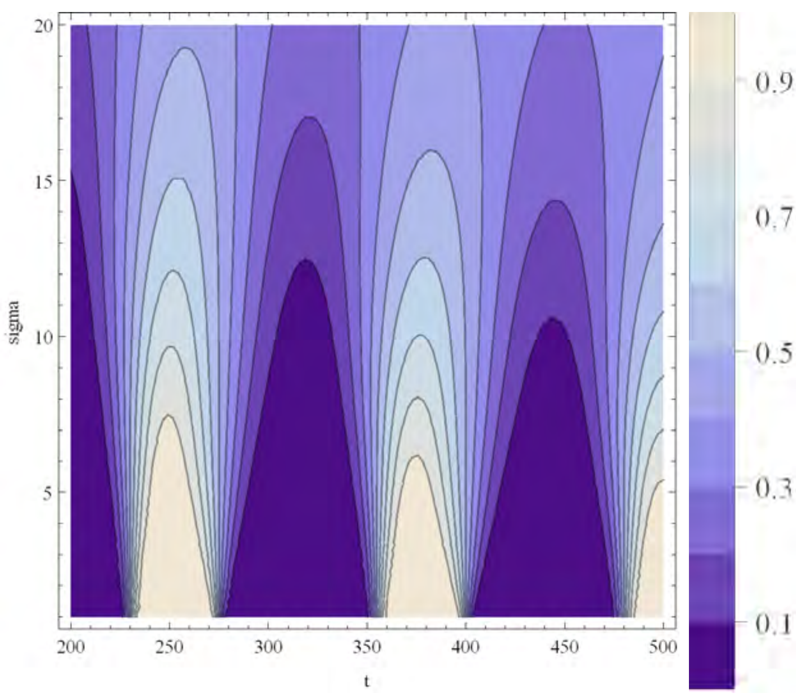
(b) $\mathbb{E}(W(t))$ at time t

Figure 4: Plots

In Figure 5, $p(t)$ is plotted for different values for t and $\sigma_G = \sigma_R$. We see that for bigger values for $\sigma_G = \sigma_R$, $p(t)$ will converge faster toward $\frac{\mu_G}{\mu_G+\mu_R} = \frac{45}{125} = 0.36$ (see Equation (12)). This effect is also very clear in Figure 6, where for $\sigma_G = \sigma_R = 5$ the convergence is much faster than for $\sigma_G = \sigma_R = \frac{5}{3}$.

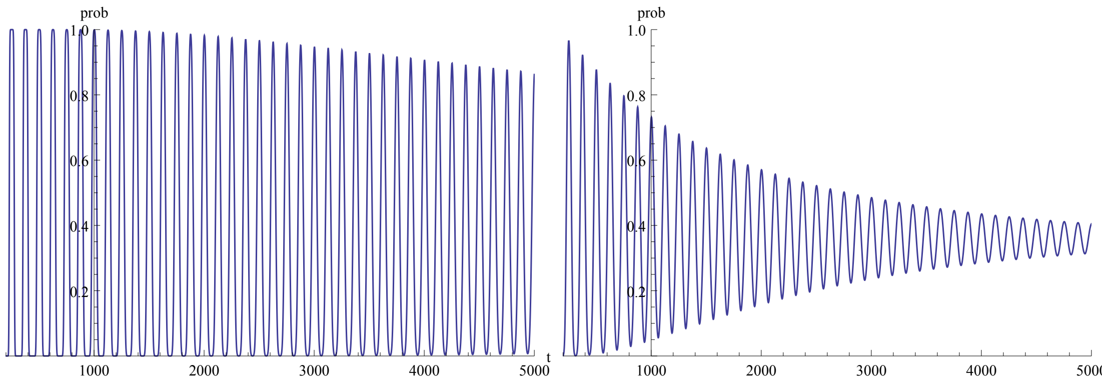


(a) 3D plot



(b) Contour plot

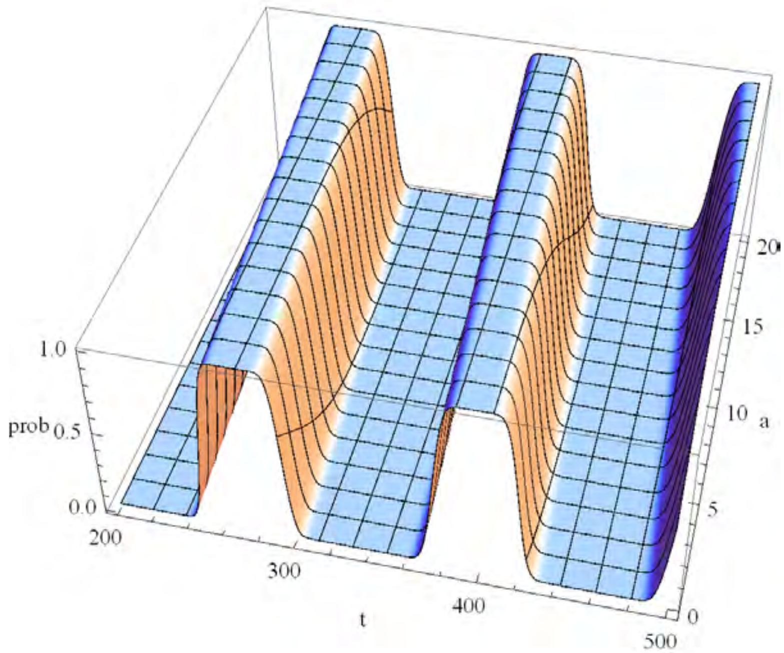
Figure 5: $p(t)$ for various values of t and σ



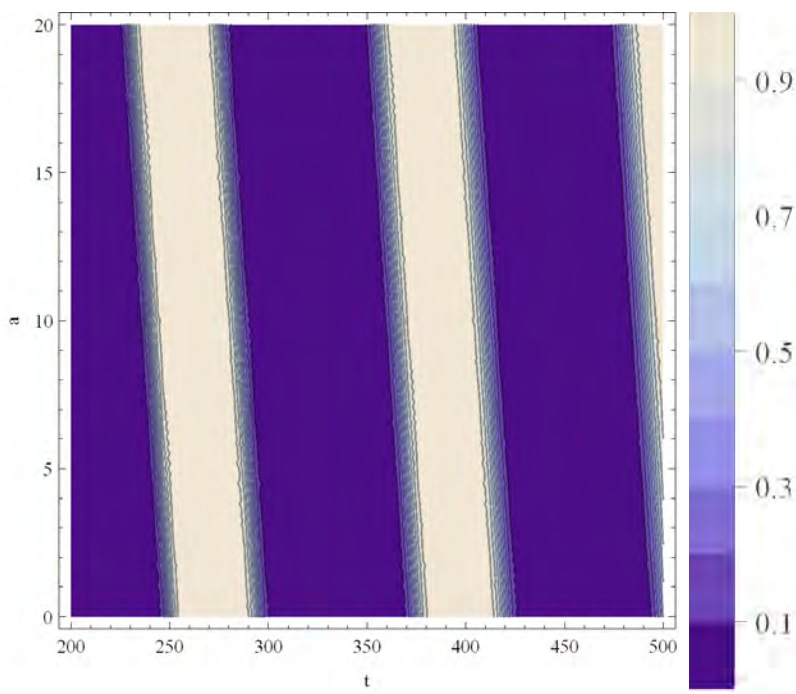
(a) $p(t)$ at t for $\sigma_G = \sigma_R = \frac{5}{3}$ (b) $p(t)$ at t for $\sigma_G = \sigma_R = 5$

Figure 6: Plots

Figure 7 shows $p(t)$ for t and a. Here we clearly see that the probability shifts linearly if a changes, so the parameter is important for the transient behavior.



(a) 3D plot



(b) Contour plot

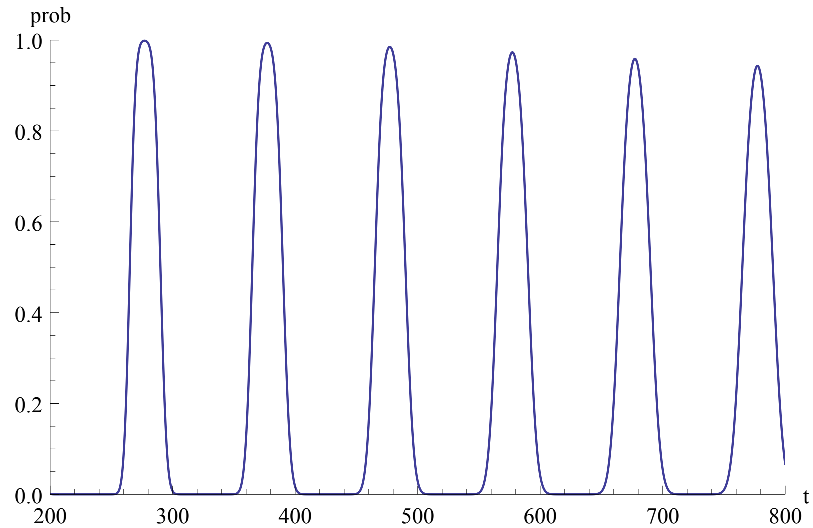
Figure 7: $p(t)$ for various values of t and a

2.5.2 Example 2

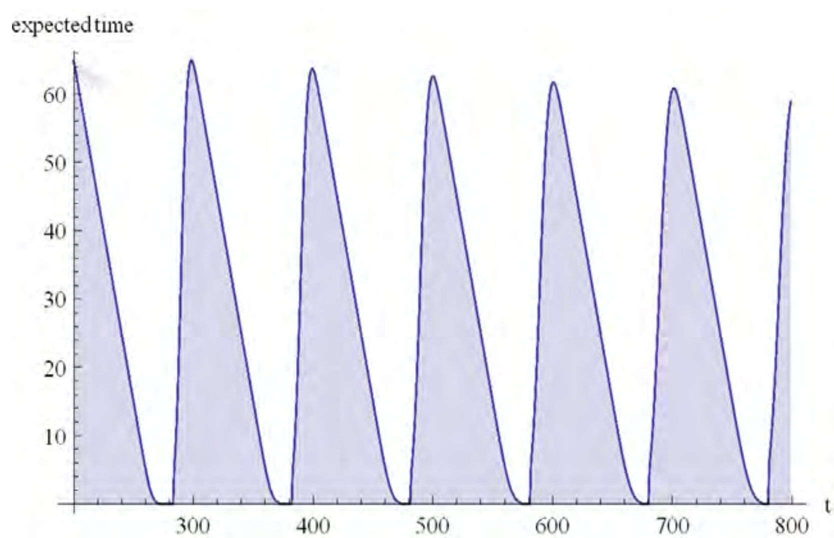
In this example we take different input to see if the functions will behave the same. In this case the traffic light starts with a red light and the cycle length is a bit smaller. As input we take:

$a = 10,$
$\mu_G = 25,$
$\sigma_G = \frac{5}{3},$
$\mu_R = 75,$
$\sigma_R = \frac{5}{3},$
$r = 1.$

We see the same kind of behavior as in Example 1, only the function converges faster in t. This is caused by the variance having more effect on a smaller cycle length.

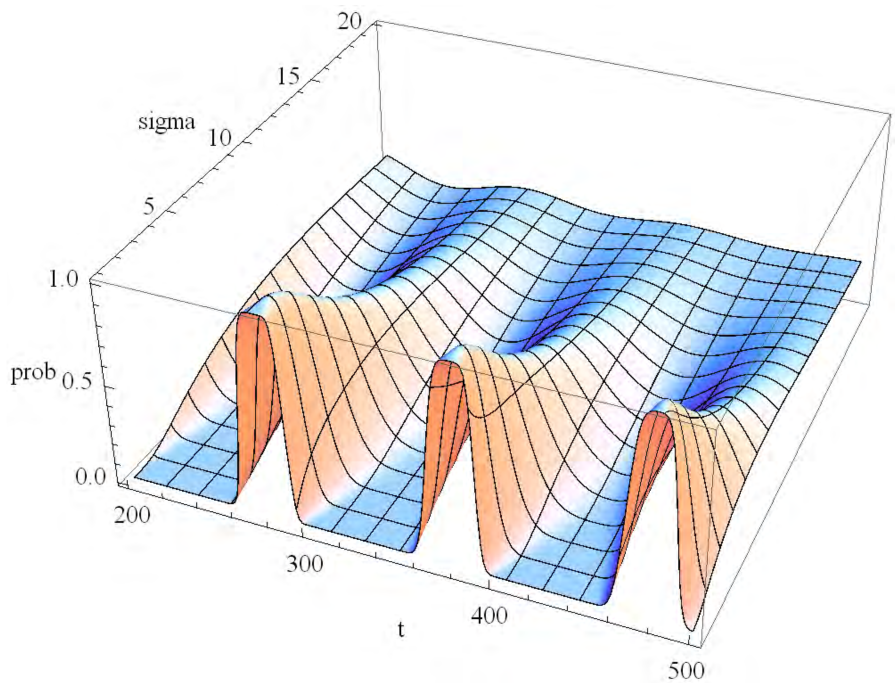


(a) $p(t)$ at time t

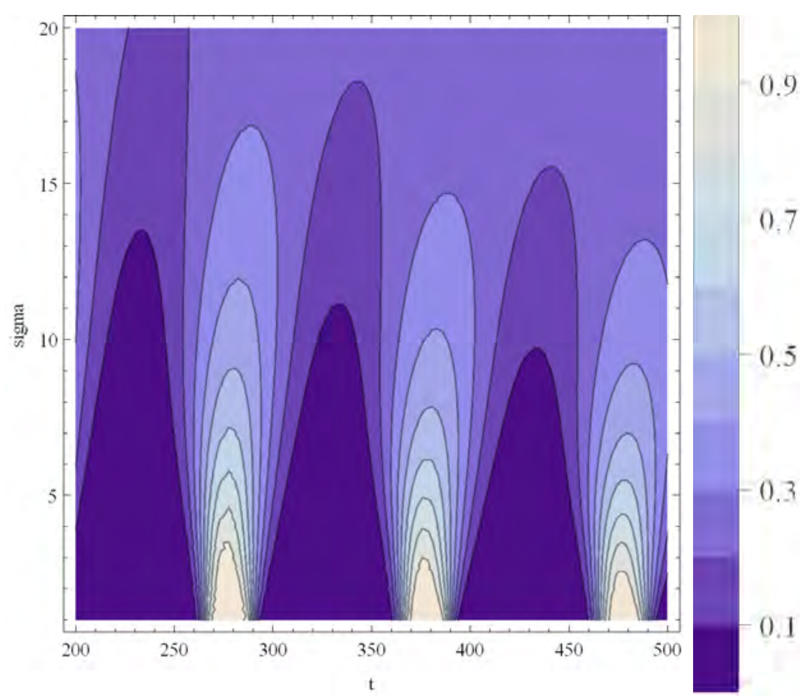


(b) $\mathbb{E}(W(t))$ at time t

Figure 8: Plots

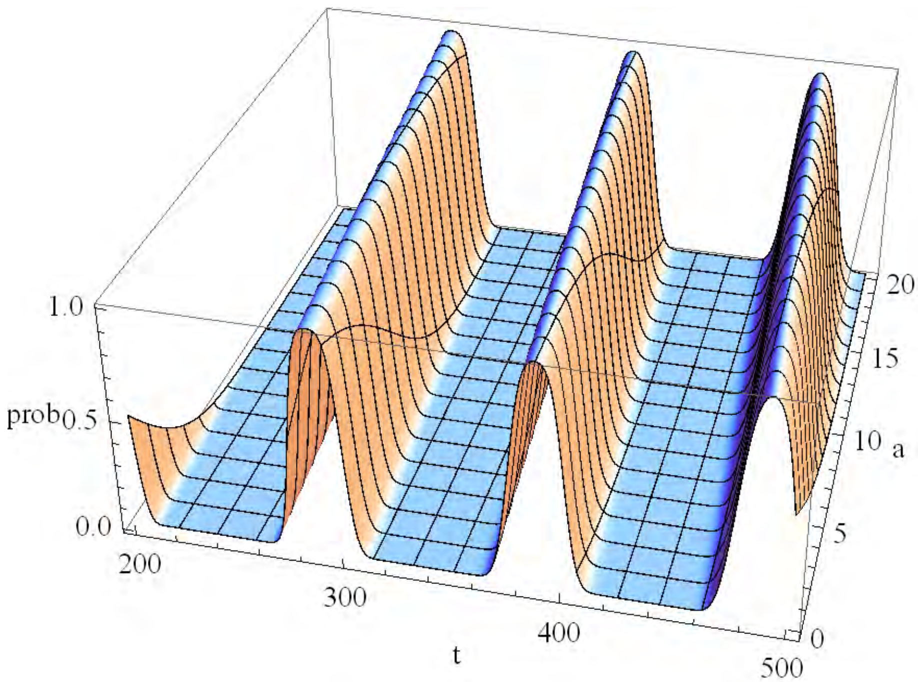


(a) 3D plot

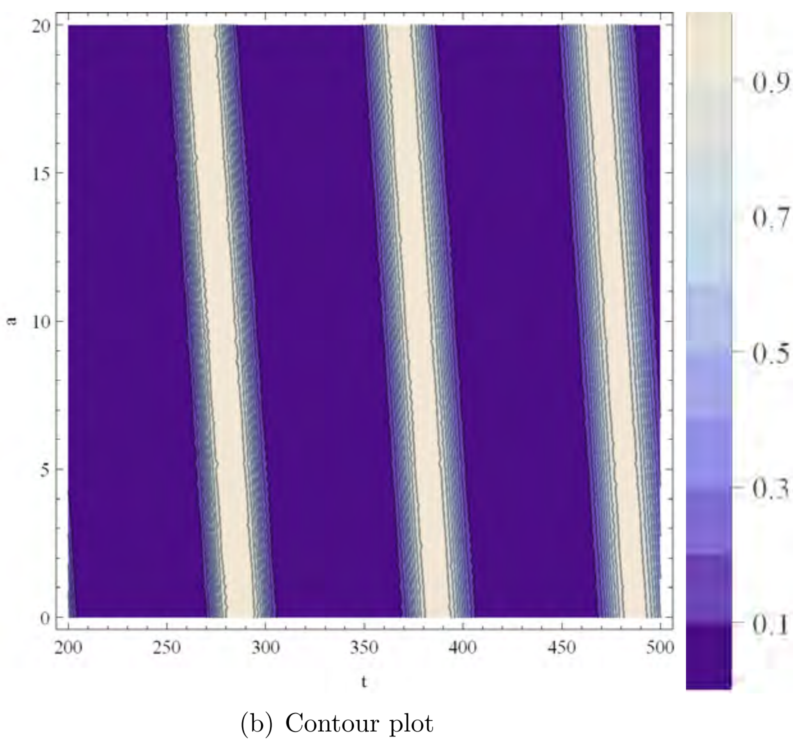


(b) Contour plot

Figure 9: $p(t)$ for various values of t and σ



(a) 3D plot



(b) Contour plot

Figure 10: $p(t)$ for various values of t and a

2.6 Conclusions

If we model the green and red times as independent random variables, we are able to give future predictions based on probability theory. As input for the phase prediction model, we need the distributions of the green and red times and the current state of the traffic light. For each moment in the future, we can calculate the probability that the traffic light will be green. We are also able to give the expected waiting time until green for each arrival time at the traffic light. The traffic light predictions have a harmonic behavior, with the average cycle length as period. When the variance of the green or red times increases, the probabilities will converge faster and the traffic light is less predictable.

3 Network of statically managed traffic lights

In this section, we will analyze the expected travel time of different routes in a network of statically managed traffic lights. The future phases of this kind of traffic lights can be predicted precisely. We start our research with an easy case, to show how traffic light specifications will influence the expected travel time and how possibly faster routes can be found.

3.1 Description of the network in Assen

To study the routes in a network of statically managed traffic lights we use a network in Assen. The center of Assen is located south of this network and the highway is north. In the network two green waves are implemented: one for the west flow and one for the east flow. For both flows the green wave is implemented for each direction. See Figure 11 for an overview of the network.

If a car at the south of this network wants to go to Groningen, it has to drive toward the upper right intersection. To reach this destination the car can go by the left route (intersection 46, 37, 45 and 47) or the right route (intersection 48 and 47). The right route has only two traffic lights and is on average about 48 seconds faster; therefore the existing route planners will always advise to take the right route. The question is: ***if we know the states and cycle plans of the traffic lights, can we find situations where the left route is faster?***

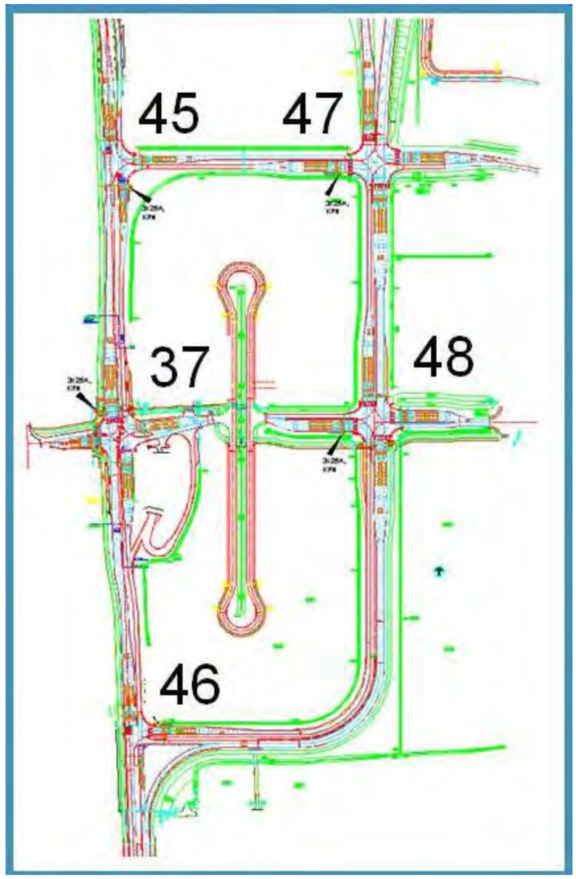


Figure 11: Network in Assen with traffic lights

PeekTraffic has provided us with specifications about the traffic lights. PeekTraffic is a company that does traffic research. The company provides traffic solutions for countries in Europe. The cycle plan depends on time and traffic intensity. The day is divided in morning, day, evening and night. For each time period (except night) there are two cycle plans depending on traffic intensity: normal or rather heavy traffic. The system can also switch to light traffic (night cycle plan) or a special cycle plan for congestion. Intersections 46, 37 and 45 must always have the same cycle to guarantee the green wave, this also holds for intersections 47 and 48. The traffic lights can be considered as very statically managed. Only for some directions it is possible to turn earlier to green if there is no traffic on conflicting signal groups, or the green light can be extended under the same condition. The lights for the green wave cannot turn earlier to green, or else the green wave is disturbed. So if the intensity at the network is high enough we can assume that the traffic lights are statically managed. Especially during Heavy Traffic this will be the case.

3.2 Modeling the route

If we know the route and all required information from the traffic lights, we can give a main model for the route. It takes T_1 seconds to drive to intersection 1, T_2 seconds to get from intersection 1 to intersection 2, etc. At each intersection the car has to wait some time until the next green time, which can be calculated from function (6). If a route has I signalized intersections, it will have $I+1$ sub routes as illustrated in Figure 12:

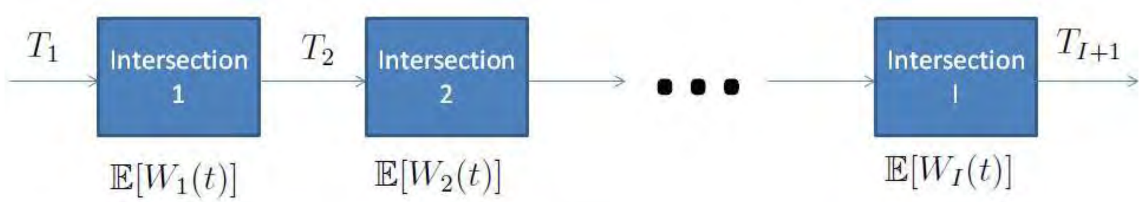


Figure 12: Abstract view of a route

Define $\mathbb{E}[S(i)]$ as the expected travel time after i sub routes, just after T_i (for $i = 1, 2, ..., I+1$). The expected total travel time from Figure 12 (also called the expected sojourn time) can now be computed with the following recursion:

$$\begin{aligned}
\mathbb{E}[S(1)] &= T_1, \\
\mathbb{E}[S(2)] &= \mathbb{E}[S(1)] + \mathbb{E}[W_1(\mathbb{E}[S(1)])] + T_2, \\
\mathbb{E}[S(3)] &= \mathbb{E}[S(2)] + \mathbb{E}[W_2(\mathbb{E}[S(2)])] + T_3, \\
&\vdots \\
\mathbb{E}[S(I+1)] &= \mathbb{E}[S(I)] + \mathbb{E}[W_I(\mathbb{E}[S(I)])] + T_{I+1}.
\end{aligned}$$

So the expected sojourn time of the route is $\mathbb{E}[S(I+1)]$.

3.3 Assumptions

Under the following assumptions the sojourn times are deterministic:

- All traffic lights are statically managed.
- The active cycle plan is known.
- The travel times between traffic lights are deterministic (speed of the vehicle is constant and there are no queues).

3.4 Calculation of the optimal route

To analyze the sojourn times of the two routes we take the normal morning cycle plan. In Table 1 and 2 the green times, red times, start of green times in the cycle, cycle times and travel times between intersections are given. (In both routes we take $T_{I+1} = 0$ for convenience).

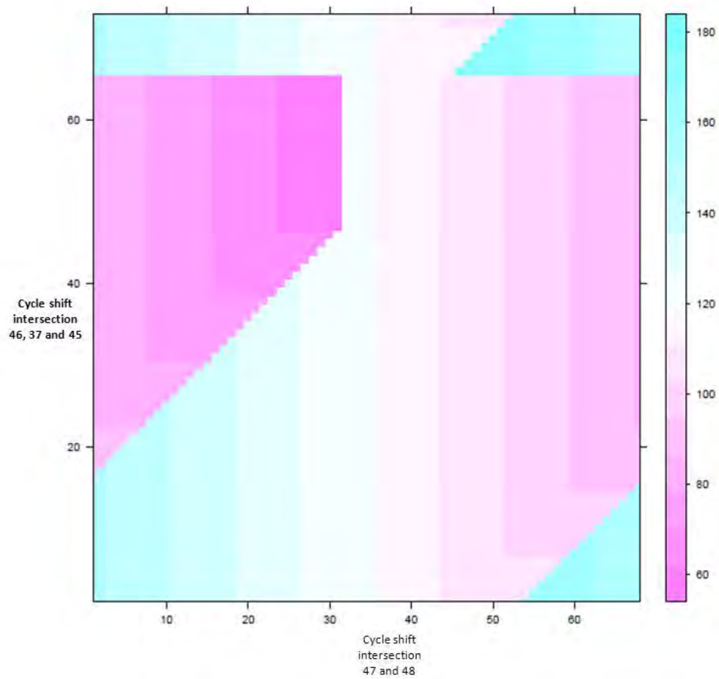
i	Green start	Green time	Red time	Cycle length	T
46	54	21	52	73	10
37	1	24	49	73	20
45	20	40	33	73	17
47	17	8	60	68	15

Table 1: Left route, normal morning cycle plan

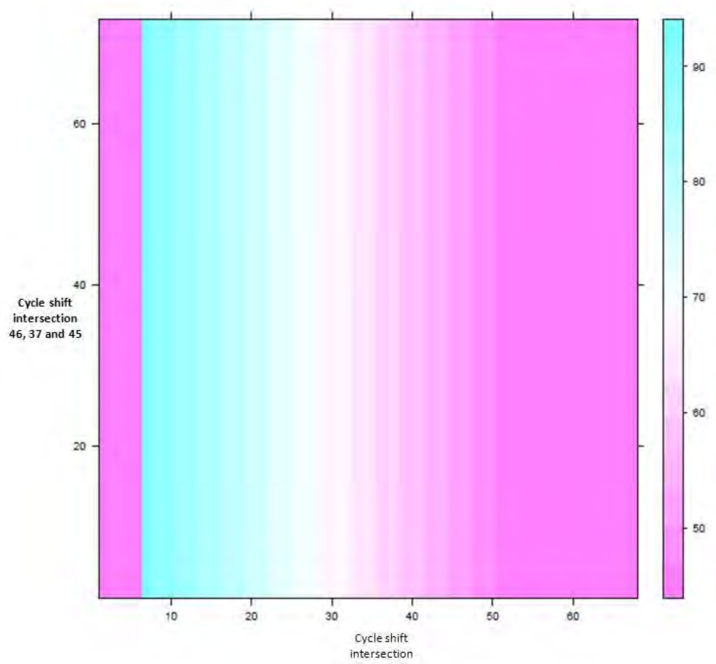
i	Green start	Green time	Red time	Cycle length	T
48	12	24	44	68	30
47	29	27	41	68	17

Table 2: Right route, normal morning cycle plan

If we discretize the cycles in seconds, there are 73 starting possibilities for the traffic lights of intersections 46, 37 and 45 and there are 68 starting possibilities for intersections 47 and 48. When for all $73 \times 68 = 4964$ starting combinations of the traffic lights the routes are calculated, then the left route is on average 109.3 seconds and the right route 61.6 seconds. However, in 15.7% of the cases the left route is faster (with 10.5 seconds on average). These results are plotted in Figure 13. From this example we can conclude that: if we exactly know the cycle plan, status of the traffic lights and the travel times between intersections, it is sometimes possible to advise a faster route than usual.



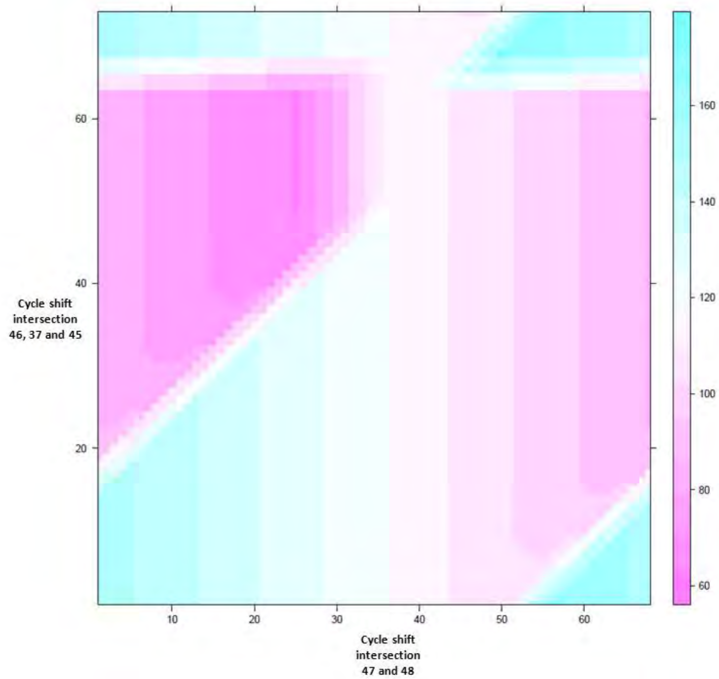
(a) Left route



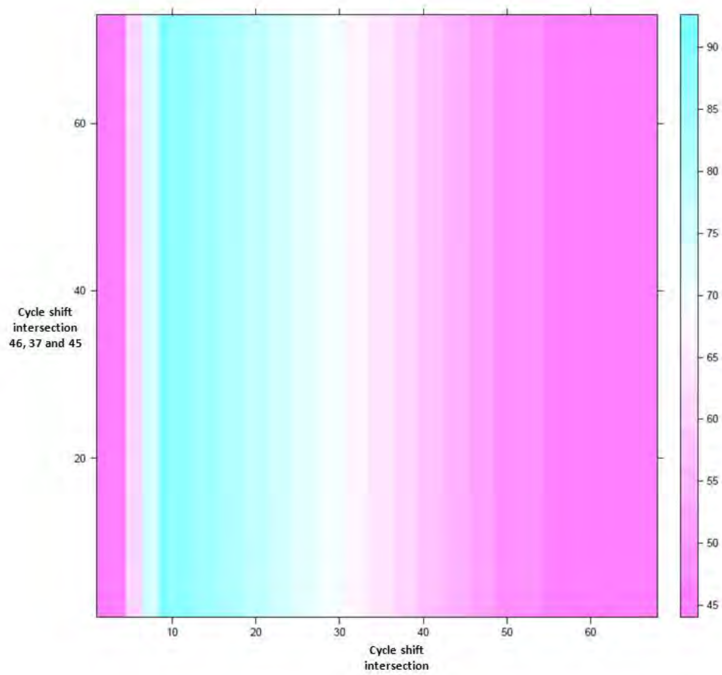
(b) Right route

Figure 13: Sojourn times Assen, all parameters deterministic

But the assumption that travel times between intersections are deterministic is not very realistic due to disturbances from other traffic. In Table 1 we assumed that it takes 17 seconds to drive from intersection 37 to intersection 45. Let this travel time now be a uniform random variable between 15 and 19 seconds, then the sojourn time can be approximated by computing the average of the routes for 15, 17 and 19 seconds. Doing this for all 3^I combinations of T_i, leads to an average sojourn time of 109.4 seconds for the left route and 62.0 seconds for the right route. Note that these averages are slightly bigger than in the completely deterministic case. Like in queueing theory more variation leads to larger sojourn times. This uncertainty is especially fatal at the boundaries, where a probability of arriving just a bit too late for the green light results in an extra waiting time for red light. Under these assumptions the left route is only 10.7% of the cases faster (with 11.2 seconds on average). See Figure 14 for the results.



(a) Left route



(b) Right route

Figure 14: Sojourn times Assen, T is random variable

3.5 Conclusions

At a network in the north of Assen, the driver has two possible routes (left and right). The left route contains four traffic lights and the right route only two traffic lights. The right route will always be advised by the TomTom, because it is 47 seconds faster on average. But our traffic light predictions show that in 15.7% of the cases the left route will be faster (with 10.5 seconds on average). If we allow more variation in traffic speeds between intersections, the left route will still be faster in 10.7% of the cases (with 11.2 seconds on average). **So if all the parameters are approximated accurately enough, we are sometimes able to suggest a faster route than current methods**.

4 Determining the cycle plan using TomTom traces

Traffic lights operate with a cycle of green and red times, these times depend on its cycle plan. The cycle plan of an intersection can change, depending on time and traffic intensity. To be able to give a prediction of the traffic light, we must know the current cycle plan and the current position of the traffic light in the green/red cycle. In Section 2 we indicated the recent status of the traffic light by a. This was the elapsed time that the traffic light was green or red. Oftentimes we do not know the current status, so we must use some data to determine the green/red cycle. If the traffic light is live communicating we can receive its information directly, else we have to use recent data gathered by TomTom. Vehicles with a live TomTom device send their traces anonymously, these traces are lists with GPS coordinates where the car has driven. Via data about stopping and driving of vehicles at an intersection we want to determine the moments in time where the light was green/red. This data can be inaccurate or limited, so the goal is: ***determine the best fitting cycle plan and give a good approximation of the current elapsed green/red time.***

In this section we first explain the main idea to determine the cycle by TomTom traces. For the main idea we analyze the expected time needed to approximate the cycle with a given accuracy. To be able to do these probability calculations, some assumptions have to been made. The probabilities are repeatedly calculated for two extreme cases: Light Traffic and Heavy Traffic. In Heavy Traffic, we likely have more TomTom traces and the cycle can be determined much faster. After the theoretical part, we present the algorithm that can determine which cycle plan is occurring and what the current state of the traffic light is. The algorithm is extensively tested by simulations. The section ends with a real implementation in Portland.

4.1 The main method for determining the cycle

For the traces we consider the time when a car just passed the stop line. At this time it is certain that the corresponding signal group has green light, assuming that vehicles don't drive through a red light. For example consider signal group 2 in Figure 15: we take the times that a car passes square 2, given that this vehicle first passes square 1 and finally passes square 3a, 3b or 3c. Describe these times when the traffic light was surely green as "green observations".

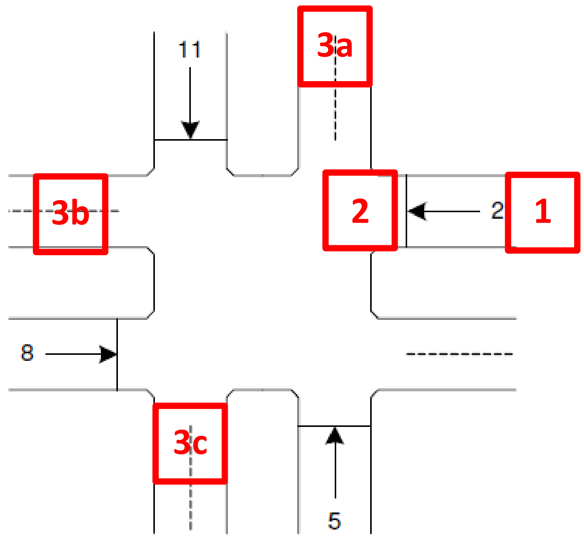


Figure 15: Intersection with 4 signal groups

Again we see the traffic light as a cycle of green and red times. The green observations can be visualized by red crosses as in Figure 16. After putting several crosses we shift the known green/red cycle such that all crosses lie in the green times. When we have at least one green observation at the beginning of a green time and we have at least one green observation at the end of a green time it is not longer possible to shift the cycle to the left or the right. Now we have matched the cycle successfully and we can determine the actual status of the traffic light.

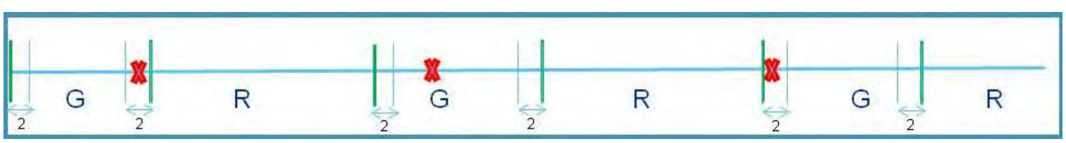


Figure 16: Mapping the cycle with observations from traces using one signal group

In practice we will do approximations to determine the cycle faster. An obvious approximation is to divide the green time into time slots of two seconds, the reason for this size being that at a dissolving lane the mean time between two successive cars is approximately two seconds. So if we observe at least one TomTom vehicle during the first time slot and at least one during the last time slot we have an approximation of maximum two seconds for the status of a traffic light.

Define the probability that a random car has a live communication TomTom device by p_{TomTom}. The fraction of vehicles with a TomTom is a very important factor for the accuracy of the model. In 2013 about 2% of the cars in the Netherlands has a live communicating TomTom device. In this section we will derive that if this percentage increases, the method will become much faster.

4.2 Assumptions

In order to do mathematical probability calculations on the explained method above, we must make some assumptions:

1. **The orange light is considered as green light.**
   We see orange light as allowed time to drive by the traffic light, because we give the predictions when arriving at the stop line.

2. **The traffic light is statically managed.**
   This also means that the green times and red times are deterministic.

3. **Green times are smaller than red times.**
   Without this assumption it is not directly possible to determine if two green observations are part of the same green time. See also Section 4.6.

4. **No drivers through red light.**
   Drivers through a red light give false green observations.

5. **Vehicles arrive according to a Poisson process with the same intensity during all cycles.**
   The Poisson process indicates that the cars arrive independently, which is realistic for stand alone intersections.

6. **If according to the Poisson process more than one car arrives during a slot of two seconds, we consider this as an arrival of precisely one car.**
   This assumption only has a very small effect for the last time slot in the Light Traffic case; the probability that according to the Poisson process more than one car arrives in two seconds is negligible for the model.

7. **The queue has a constant service rate of 0.5 vehicles/second.**
   Because we take time slots of 2 seconds this service rate implies that in each slot 0 or 1 vehicles depart from the traffic light.

8. **Traces of TomTom devices are sufficiently accurate to calculate the crossing over time of the vehicle correctly.**
   Due to GPS inaccuracy, not all TomTom traces are usable/correct for our analysis. So in practice we need more traces or the traces lead to wrong conclusions.

4.3 Light Traffic

In light traffic the arrival intensity is low enough such that the traffic light is always easily able to serve all the cars during its green time. The probability that we observe a TomTom car during the first time slot is the probability that at least one car has arrived during the red time plus 2 seconds from the first slot and the first arriving car has a live TomTom device. The probability that we see a TomTom car during the last time slot is the probability that a car arrives during this slot and this car has a TomTom device. If we analyze signal group 2, we denote the probability of a TomTom car arriving in the first time slot as p_{21} and in the last slot as p_{23}. Denote the number of arrivals for signal group 2 during t seconds by $N_2(t)$. The probabilities can be easily computed:

$$\begin{aligned} p_{21} &= \mathbb{P}\{N_2(R_2+2) > 0\}\, p_{\text{TomTom}} = (1 - \mathbb{P}\{N_2(R_2+2) = 0\})\, p_{\text{TomTom}} \\ &= \left(1 - e^{-(R_2+2)\lambda_2}\right) p_{\text{TomTom}}, \\ p_{23} &= \mathbb{P}\{N(2) > 0\}\, p_{\text{TomTom}} = (1 - \mathbb{P}\{N_2(2) = 0\})\, p_{\text{TomTom}} \\ &= \left(1 - e^{-2\lambda_2}\right) p_{\text{TomTom}}. \end{aligned}$$

Now we want to calculate the probability to have an approximation that is accurate up to 2 seconds, after looking N cycles back (call this probability p_{success}). To achieve a two second approximation, we must observe at least one TomTom car during the first time slot and at least one TomTom car during the last time slot. This probability is calculated below:

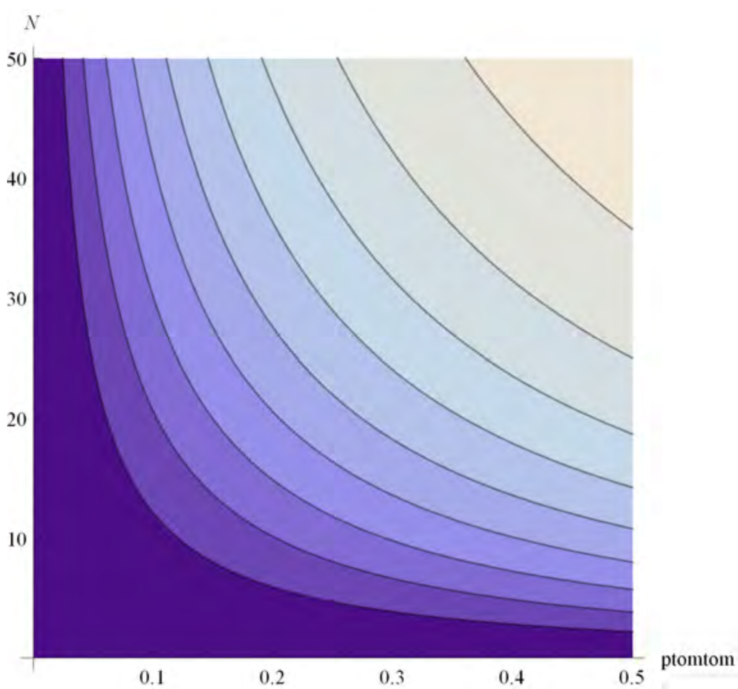
$$\begin{aligned} p_{\text{success}} &= \mathbb{P}\{\text{TomTom car during first and last slot for } N \text{ cycles}\} \\ &= (1 - (1 - p_{21})^N)(1 - (1 - p_{23})^N). \end{aligned}$$

For the intersection of Figure 15 we can take the following parameters (which belong to Light Traffic):

$$\lambda_2 = \lambda_8 = \frac{1}{15},$$
$$\lambda_5 = \lambda_{11} = \frac{1}{30},$$

$$G_2 = G_8 = 20,$$
$$R_2 = R_8 = 50,$$
$$G_5 = G_{11} = 15,$$
$$R_5 = R_{11} = 55.$$

Now we can plot the effect of p_{TomTom} and N to p_{success}, see Figure 17. Here we see that p_{TomTom} has a huge effect on the amount of cycles that we have to look back to successfully map the cycle.



(a) p_{success} at N cycles and p_{TomTom}

(b) p_{succes}

Figure 17: Contour plot Light Traffic, 1 signal group

Only considering signal group 2 will not be fast enough to map the cycle. Thus we also use the green observations of the other signal groups. For example in Figure 18 signal group 2 and 8 have a green light at the same time, directly followed by signal group 5 and 11 (hereafter bicycles and pedestrians can have green). These group numbers are chosen according to Dutch standards. Now all these combinations of time slots with green observations give the desired approximation:

- First slot group 2 and last slot group 2, 8, 5 or 11.
- First slot group 8 and last slot group 2, 8, 5 or 11.
- First slot group 5 and last slot group 2, 8, 5 or 11.
- First slot group 11 and last slot group 2, 8, 5 or 11.

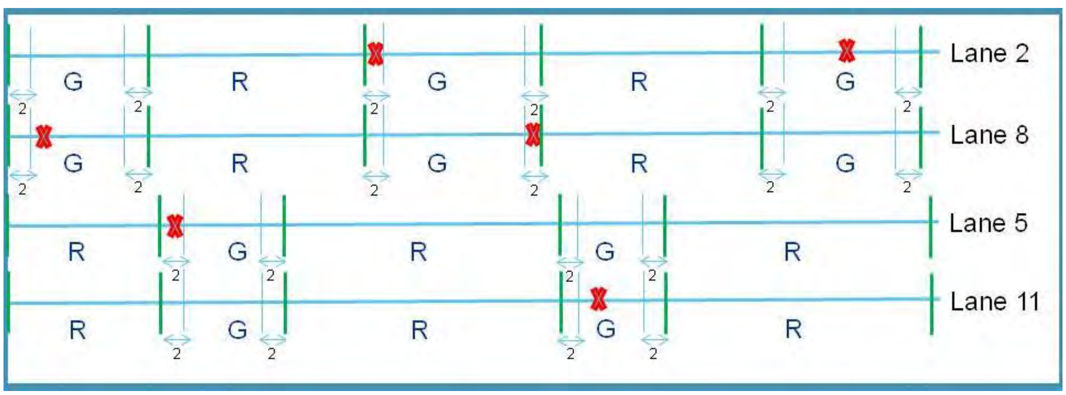
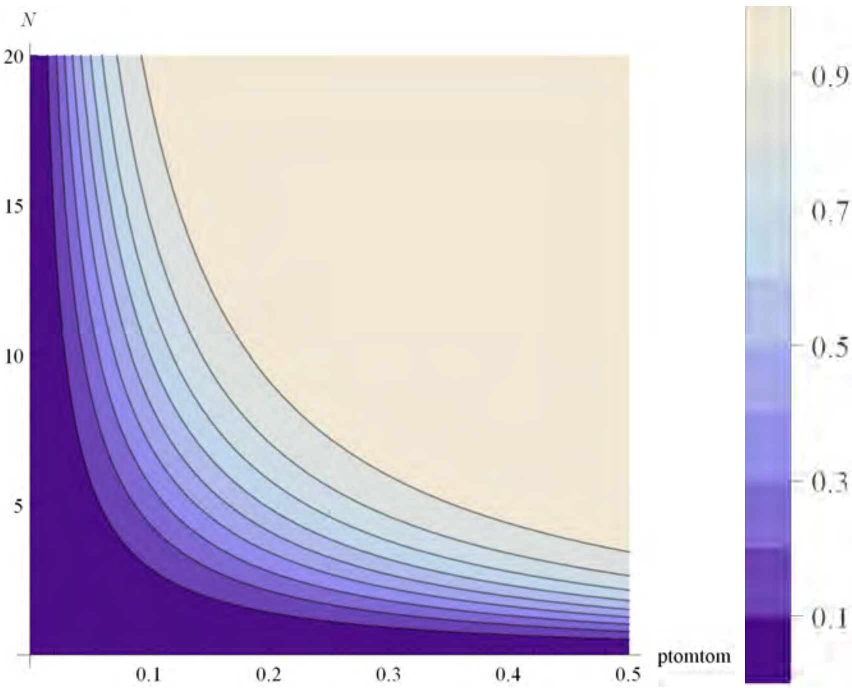


Figure 18: Mapping the cycle with observations from traces using four signal groups

If any of the combinations above is occurring, we also have an approximation that is accurate up to two seconds. When we again plot the effect of p_{TomTom} and N to p_{success} (see Figure 19), the improvement is huge.



(a) p_{success} at N cycles and p_{TomTom} (b) p_{succes}

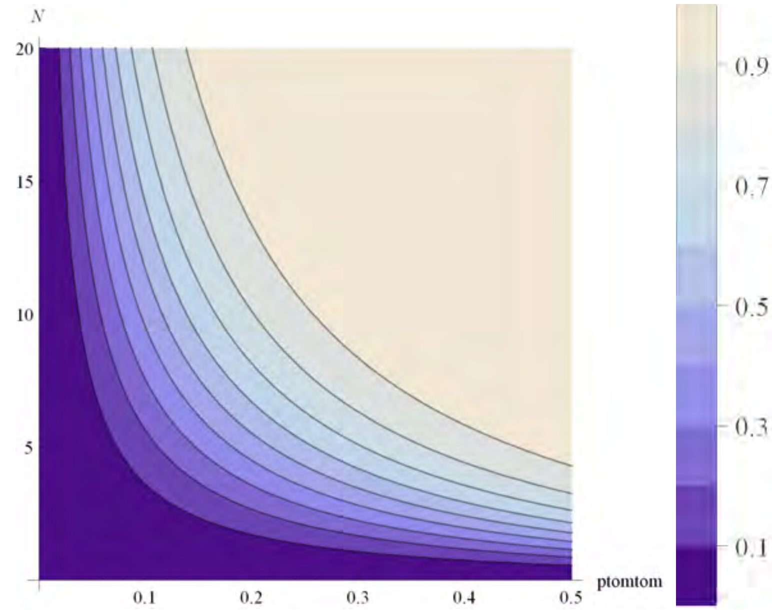
Figure 19: Contour plot Light Traffic, combinations of 4 signal groups

4.4 Heavy Traffic

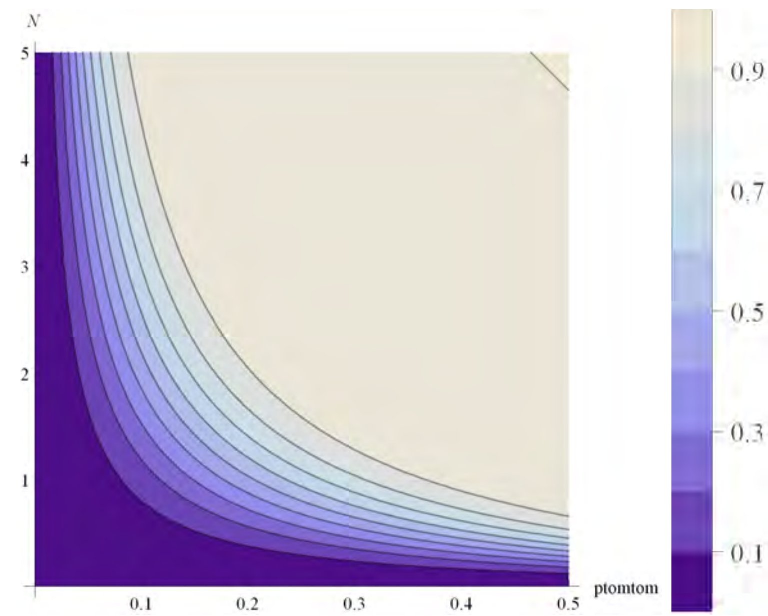
During heavy traffic, the traffic intensity is sufficiently large such that the intersection is saturated and the queue will never become empty. This implies that during green light, the probability that a car departs is 1 for each time slot. So the probability that we see this car is p_{TomTom}:

$$\begin{aligned}
p_{21} &= p_{\text{TomTom}}, \\
p_{23} &= p_{\text{TomTom}}, \\
p_{\text{success}} &= \mathbb{P}\{\text{TomTom car during first and last slot for } N \text{ cycles}\} \\
&= (1-(1-p_{21})^N)(1-(1-p_{23})^N).
\end{aligned}$$

Like in the previous section we combine all four signal groups. These results are shown in Figure 20. Of course the probabilities for success are much larger than for Light Traffic.



(a) 1 signal group (b) p_{succes}



(c) combinations of 4 signal groups (d) p_{succes}

Figure 20: Contour plots Heavy Traffic, p_{success} at N cycles and p_{TomTom}

4.5 Some extensions

4.5.1 Signal group with two lanes

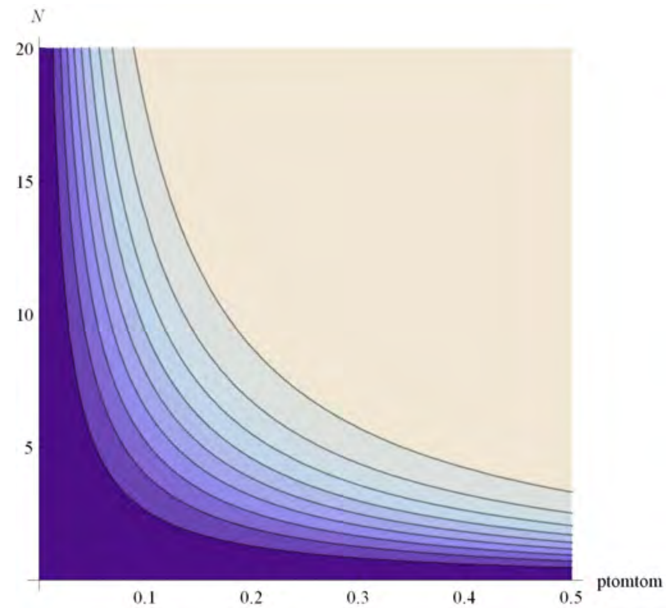
Many large intersections have signal groups consisting of two lanes. In this case two cars can queue next to each other in front of the traffic light. To observe a TomTom car in the first time slot, the first or the second arriving car must have a live TomTom device. So either at least two cars arrive and the second has a TomTom or at least one car arrives and the first has a TomTom. If for example signal group 2 has two lanes, in Light Traffic the probabilities that we observe a TomTom car during the first time slot (p_{21}) and during the last time slot (p_{23}) are:

$$
\begin{aligned}
p_{21} &= 1 - [1 - \mathbb{P}\{\text{min 2 cars arrive, 2nd has TomTom}\}]\,[1 - \mathbb{P}\{\text{min 1 car arrives, 1st has TomTom}\}] \\
&= 1 - \left[1 - \left(1 - e^{-(R_2+2)\lambda_2} - (R_2+2)\lambda_2 e^{-(R_2+2)\lambda_2}\right) p_{\text{TomTom}}\right] \left[1 - \left(1 - e^{-(R_2+2)\lambda_2}\right) p_{\text{TomTom}}\right], \\
p_{23} &= 1 - \left[1 - \left(1 - e^{-2\lambda_2} - 2\lambda_2 e^{-2\lambda_2}\right) p_{\text{TomTom}}\right] \left[1 - \left(1 - e^{-2\lambda_2}\right) p_{\text{TomTom}}\right].
\end{aligned}
$$

In Heavy Traffic the probability is 1 that in each time slot two vehicles depart, thus at least one of these two cars must have a live TomTom device:

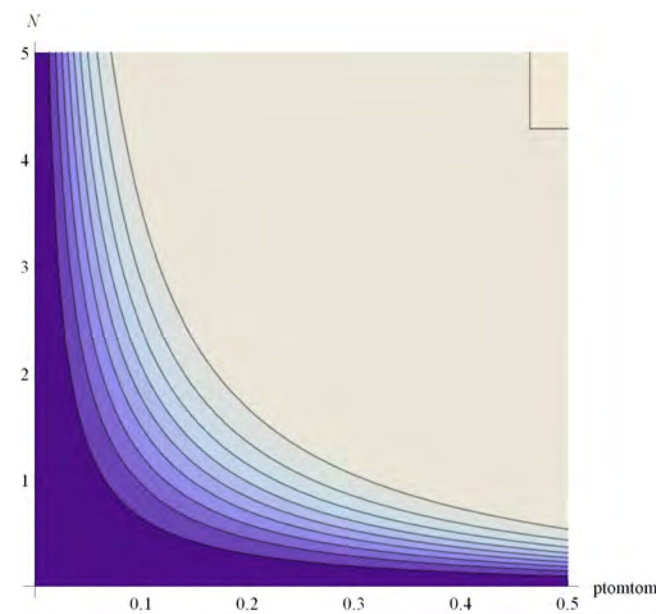
$$
\begin{aligned}
p_{21} &= 1 - \left(1 - p_{\text{TomTom}}\right)^2, \\
p_{23} &= 1 - \left(1 - p_{\text{TomTom}}\right)^2.
\end{aligned}
$$

The probabilities to have an approximation that is accurate up to 2 seconds, are shown in Figure 21. If we compare this with the one lane intersection the results are slightly better. We see for example that in this case the line $p_{\text{success}} = 0.9$ is shifted downward, which indicates that we expect to determine the cycle faster.

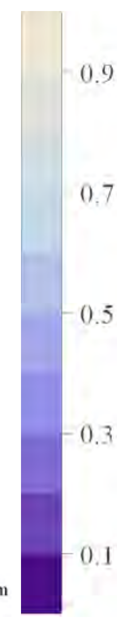


(a) Light Traffic

(b) p_{succes}



(c) Heavy Traffic



(d) p_{succes}

Figure 21: Contour plots, signal group 2 with 2 lanes, p_{succes} at N cycles and p_{TomTom}

4.5.2 4 second approximation

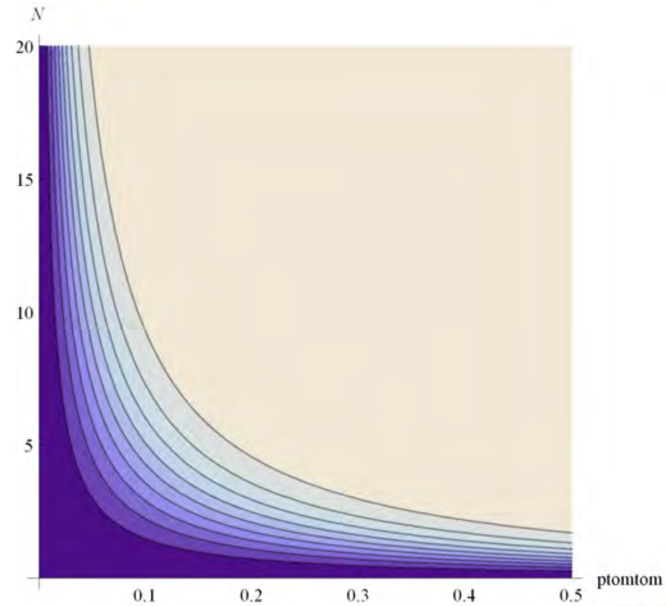
If we increase the time slots to 4 seconds, the probability to observe a live TomTom car in the first or last slot will increase. Now during all time slots, two cars can be observed. For example in Light Traffic, the probabilities of signal group 2 are:

$$\begin{aligned}
p_{21} &= 1-\left[1-\mathbb{P}\left\{\text{min 2 cars arrive, 2nd has TomTom}\right\}\right]\left[1-\mathbb{P}\left\{\text{min 1 car arrives, 1st has TomTom}\right\}\right] \\
&= 1-\left[1-\left(1-e^{-(R_2+4)\lambda_2}-(R_2+4)\lambda_2 e^{-(R_2+4)\lambda_2}\right)p_{\text{TomTom}}\right]\left[1-\left(1-e^{-(R_2+4)\lambda_2}\right)p_{\text{TomTom}}\right], \\
p_{23} &= 1-\left[1-\left(1-e^{-4\lambda_2}-4\lambda_2 e^{-4\lambda_2}\right)p_{\text{TomTom}}\right]\left[1-\left(1-e^{-4\lambda_2}\right)p_{\text{TomTom}}\right].
\end{aligned}$$

For Heavy Traffic the probabilities become (just like the case with two lanes):

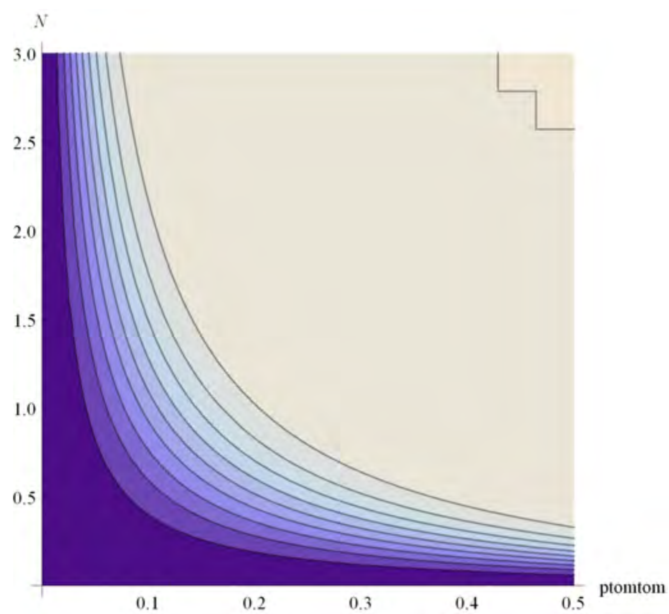
$$\begin{aligned}
p_{21} &= 1-\left(1-p_{\text{TomTom}}\right)^2, \\
p_{23} &= 1-\left(1-p_{\text{TomTom}}\right)^2.
\end{aligned}$$

See Figure 22 for the results. So for an approximation that is accurate up to four seconds, the cycle plan can be determined much faster.



(a) Light Traffic

(b) p_{succes}



(c) Heavy Traffic

(d) p_{succes}

Figure 22: Contour plots, 4 second approximation, p_{succes} at N cycles and p_{TomTom}

4.6 The algorithm of modulo calculation for determining the cycle

Unfortunately, the reality is not as ideal as described in the assumptions. Sometimes people drive through a red light, which makes the analysis impossible. In some cases the green times could be bigger than red times and we can have multiple candidate cycle plans. These items will give some fitting issues.

To be able to implement the described model, we need to use an algorithm. The algorithm should be able to choose the best fitting cycle plan, filter red drivers and make green times possible which are larger than red times. To make these exceptions possible, the algorithm likely needs more traces than analyzed earlier in this section.

If we set the current time to $t = 0$, then our green observations will have negative values. When all green observations are done in the same cycle plan, then after modulo calculation with cycle length C each green observation still lies in a green time. So if during time t_g the light is green, then during $t_g + k \cdot C$ with $k \in \mathbb{Z}$, the light is also green. Calculating modulo C will give values between $-C$ and 0. Define the resulting values as $x_1, ..., x_n$. This can lead to two cases:

1. Exactly one green time lies completely in the interval $(-C, 0]$. So all green observations belong to the same green time. No action needed.
2. Two green times partly lie in the interval $(-C, 0]$. So we see green observations at the beginning of the interval and observations at the ending of the interval which belong to other green times. Now we add C to the green observations which belong to the first green time, such that all green observations belong to the second green time.

To determine which case is occurring, we calculate the distance between all successive green observations. Define the distance between x_{i+1} and x_i as B_i. If $\max_i \{B_i\} > R$, then a red time lies between the green observations and case 2 is occurring. When all B_i are smaller than R, case 1 is occurring. Note that if $G > R$, we must have sufficiently many green observations to determine if two green observations belong to the same green time.

We can combine all green observations of signal groups which have green simultaneously. In the example below (Figure 23) signal group 2 and 8 have green simultaneously:

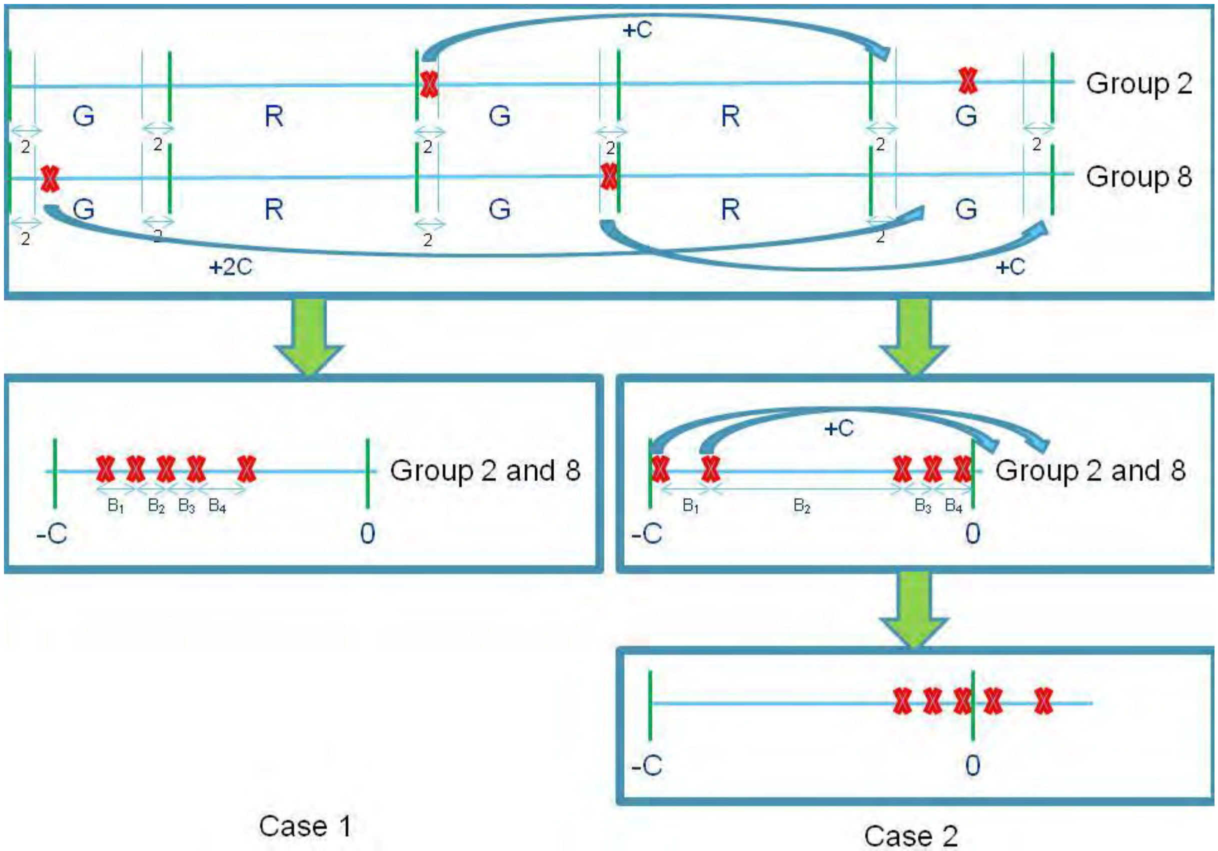
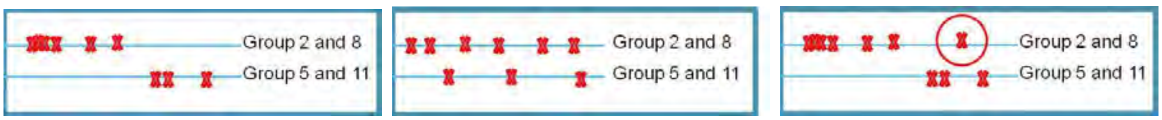


Figure 23: Modulo calculation

We apply modulo calculation for each conflicting group in a cycle plan. If the correct cycle plan is used, all green observations in a conflicting group are nicely clustered. Also the green observations are not conflicting with other groups (like group 5 and 11 in the example). If necessary, we add C to all green observations of group 5 and 11 such that they always start after the green time of group 2 and 8. This situation is shown in Figure 24a. In this example define the green observations of group 2 and 8 by $x_1, ..., x_n$ and the green observations of group 5 and 11 by $y_1, ..., y_m$. If the wrong cycle plan is applied, then the situation looks like Figure 24b. Here all green observations are spread over the interval and are conflicting with the other group. This analysis makes it also possible to detect drivers through a red light (or other measurement mistakes). In Figure 24c we observe a driver through red light, because one green observation lies in the conflicting green time. If a green observation lies in the green time of a conflicting signal group, we remove this green observation (so in the example remove x_n). So if the wrong cycle plan is chosen, a lot of green observations are removed and if the correct cycle plan is chosen, only wrong measurements are removed.

To choose which signal group has a possible driver through red light, we take the signal group which has the largest distance between successive green observations. If all green observations are correct, they should be nicely clustered and the largest distance is small. First analyze if green observations in this group should be removed, then try to remove green observations in the conflicting group.



(a) Correct cycle length (b) Wrong cycle length (c) Red driver

Figure 24: Possible situations after modulo calculation

Apply the operations above for each candidate cycle plan. The cycle plan with the least amount of removals is chosen as best fitting cycle plan. In case that more cycle plans remain with the minimum amount of removals, try to approximate the green time. Except for heavy traffic, we know that most cars will be served in the beginning of a green time because they arrive during a red light. So we probably observe most cars during the beginning of a green time. If signal group 5 and 11 start directly after signal group 2 and 8, we approximate the green time of group 2 and 8 by $V := y_1 - x_1$ (see Figure 25). In this distance also the clearance time between group 2 and 8 and group 5 and 11 is included, define the clearance time as cl. So choose the cycle plan with the smallest $|(G_2 + cl) - V|$ as the best fitting cycle plan.

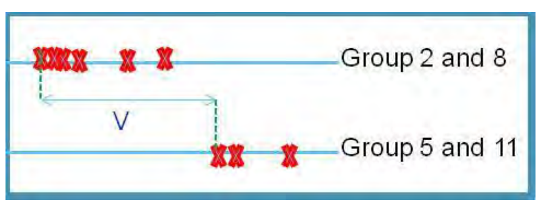


Figure 25: Best approximation for cycle choice

Now we try to approximate the green start of group 2 and 8. Choose the combination (like mentioned in Section 4.3) that approximates the theoretical distance the best. Define the following distances (as illustrated in Figure 26):

$$\begin{aligned} V_{22} &= x_n - x_0, \\ V_{25} &= y_n - x_0, \\ V_{55} &= y_n - y_0, \\ V_{52} &= y_0 - x_n. \end{aligned}$$

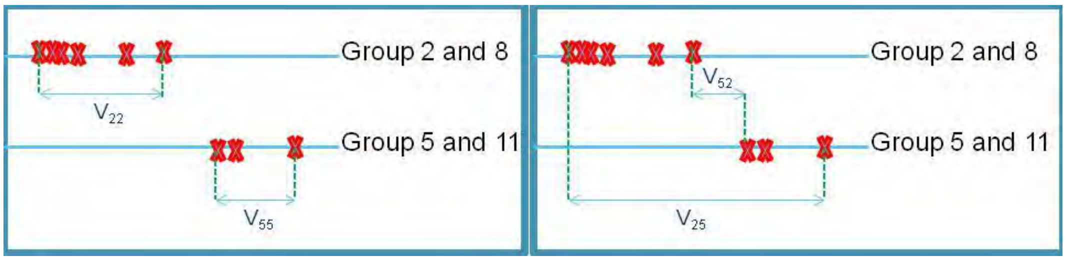


Figure 26: Best approximation for cycle position

So take the combination where corresponding $|W|$ is the smallest:

- Combination 1: $W = \frac{G_2 - V_{22}}{2}$,
- Combination 2: $W = \frac{(G_2 + G_5 + cl) - V_{25}}{2}$,
- Combination 3: $W = \frac{G_5 - V_{55}}{2}$,
- Combination 4: $W = \frac{V_{52} - cl}{2}$.

The green start of signal group 2 and 8 can now be approximated by:

- Combination 1: Green start $= x_0 - W$,
- Combination 2: Green start $= x_0 - W$,
- Combination 3: Green start $= y_0 - W - G_2 - cl$,
- Combination 4: Green start $= y_0 - W - G_2 - cl$.

In Section 4.3 and 4.4 we have calculated how many cycles we probably have to look back to retrieve a two second approximation. The idea is to apply the algorithm for a set of green observations and see if the required approximation is achieved. If the approximation is not as desired: sample more traces and try again.

4.7 Results from simulation

To test how the algorithm performs, we simulate a queue with arriving and departing cars at a traffic light. Cars arrive according to a Poisson Process during the entire cycle and cars can only be served if the light is green. The service rate of the queue is 0.5 and constant such that exactly every two seconds a car may depart at the traffic light. The arrival rate depends on the intensity of the traffic. In case of Light Traffic we use $\lambda_2 = \frac{1}{15}$ and for Heavy Traffic $\lambda_2 = \frac{1}{3}$. In this situation the traffic light is statically managed, so the green and red times are deterministic. These assumptions are also described in Section 4.2. But in the algorithm of modulo calculation we are able to filter drivers through red light and we allow that green times can be bigger than red times. In the simulation we use the following values for the cycle plan:

$$
\begin{aligned}
G_2 = G_8 &= 20, \\
R_2 = R_8 &= 50, \\
G_5 = G_{11} &= 16, \\
R_5 = R_{11} &= 54.
\end{aligned}
$$

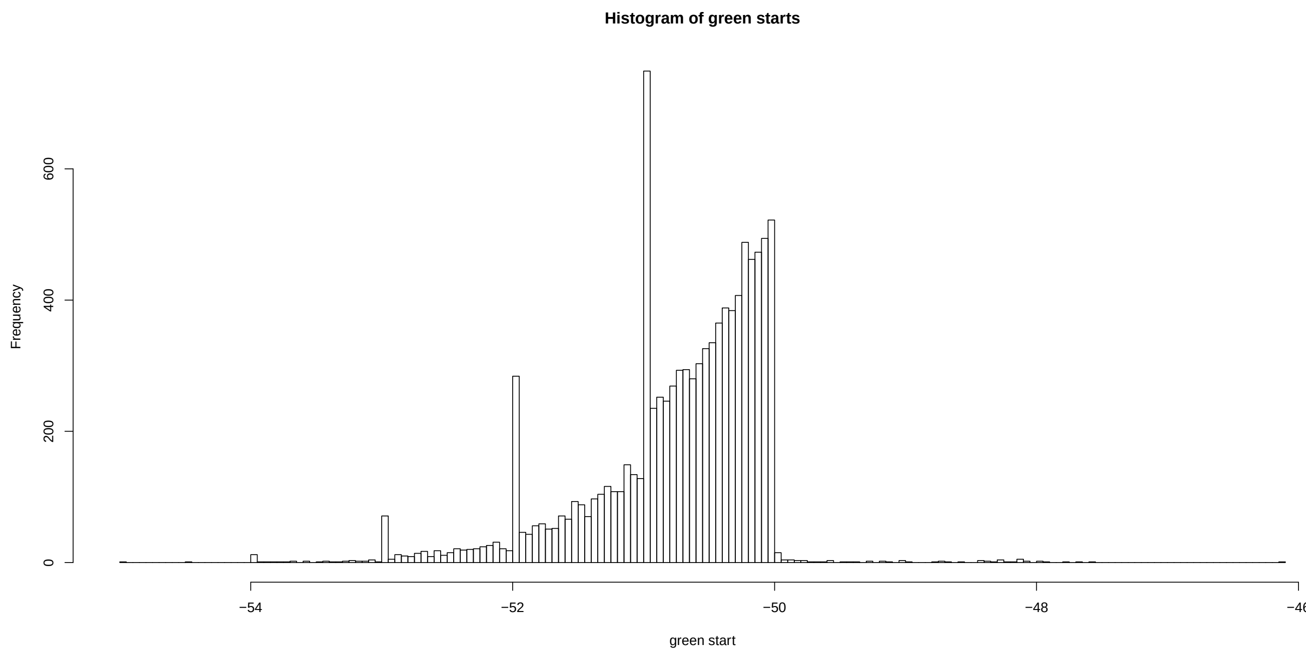
At this intersection signal group 2 and 8 have green light simultaneously, directly followed by signal group 5 and 11. The probability that we see the car depart from the traffic light is $p_{\text{TomTom}} = 0.02$. Each time we take a sample for the algorithm, the sample time for the simulation is 700 seconds (equal to ten cycles). While the approximation is not 2 seconds accurate, we sample additional green observations for 700 seconds and apply the algorithm again. We also sample additional green observations if more cycle plans have the same minimum amount of removals (more cycle plans fit equally well). Due to cycle plan transitions we can only look back a finite amount of time, so let the total time allowed be three hours. If after three hours more candidate cycle plans remain, we use the approximation in Figure 25. Take the following candidate cycle plans:

Cycle plan	G_2	R_2	G_5	R_5
1	25	50	20	55
2	15	50	10	55
3	20	50	16	54
4	22	50	18	54
5	18	52	14	56

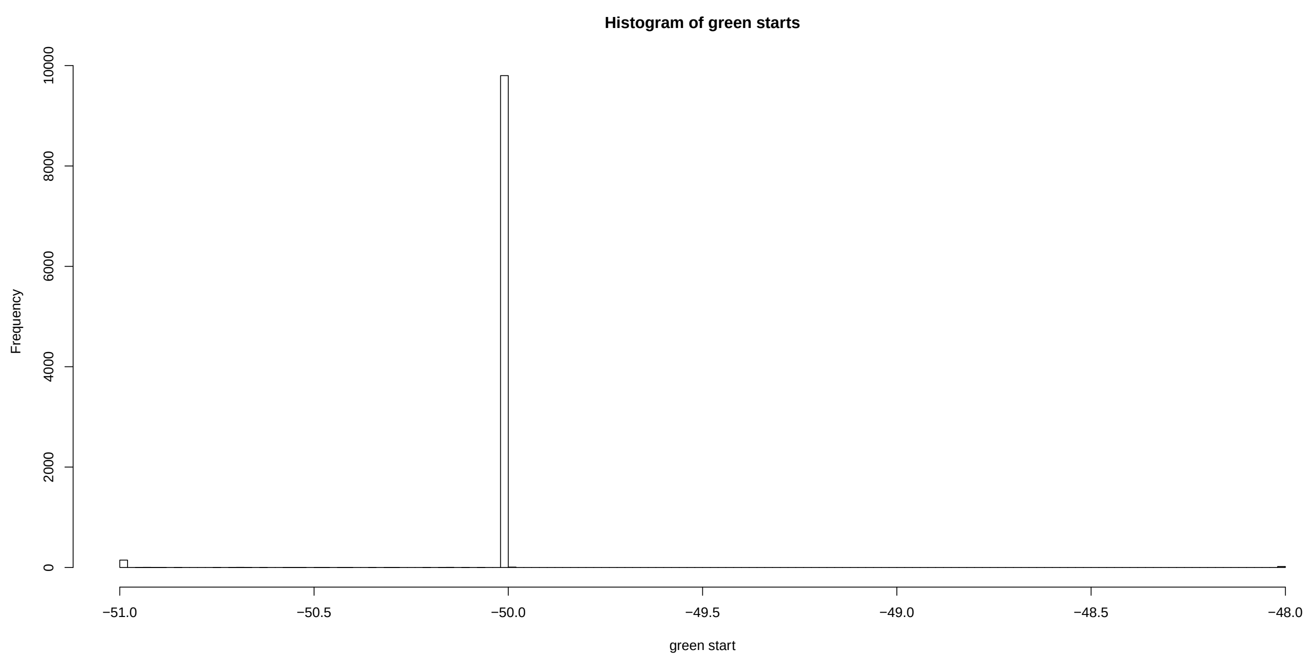
Table 3: The candidate cycle plans

Note that cycle plan 3 is the correct plan and cycle plan 5 is chosen to make it difficult for the algorithm because it has the same cycle length. We start with the situation that no drivers drive through a red light. In Figure 27, the results for the green start estimations of signal group 2 and 8 are shown for 10000 independent simulations. In this situation the correct time for the green start is -50 seconds. In case of Light Traffic we see that most green start estimations are 2 seconds accurate. In some simulations we didn't see enough green observations in three hours, but the error of the estimation was smaller than 4 seconds. In case of Heavy Traffic, almost all estimations are completely correct and a few of the 10000 are approximated within 2 seconds.

To successfully detect and filter drivers through red light, we must have enough green observations. In this simulation we first sample green observations until we have at least 6 green observations for signal group 2 and 8 and at least 5 green observations for signal group 5 and 11. These numbers are based on insight, but more mathematical research is perhaps needed for more efficiency. In each of the 10000 simulations, one green observation for signal group 2 is added at a random time. So with probability $\frac{R_2}{R_2+G_2} = \frac{5}{7}$ this green observation is a driver through red light, with probability $\frac{G_2}{G_2+R_2} = \frac{2}{7}$ this green observation is correct. The algorithm must figure out which case is occurring. In Figure 28 the results are plotted with a possible driver through red light. Here we see that still most green start estimations are correct, but in a few cases the driver through red light is not removed and the conclusion is totally wrong.

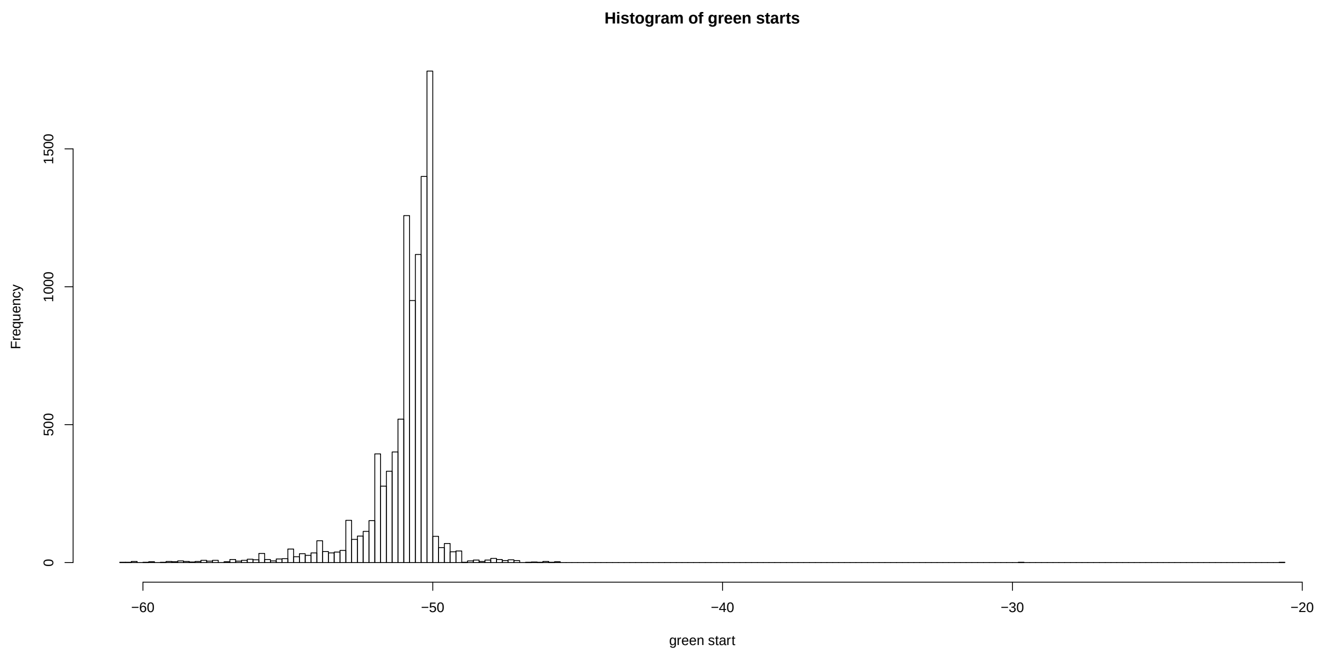


(a) Green starts Light Traffic

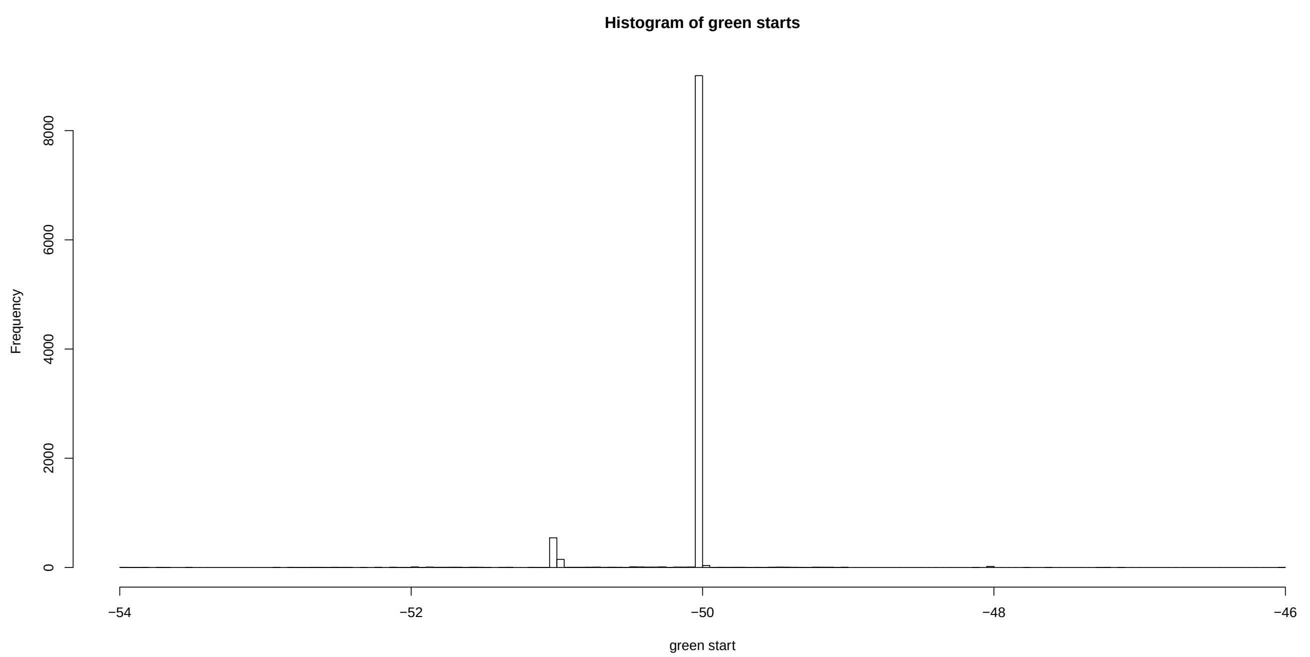


(b) Green starts Heavy Traffic

Figure 27: Green starts



(a) Green Start Light Traffic



(b) Green start Heavy Traffic

Figure 28: Green starts with a driver through red light

The results for the amount of time needed and which cycle plans are chosen are shown in Table 4. In more than 99.5 % of the simulations the correct cycle plan is chosen. For Light Traffic we need about 150 minutes of TomTom traces and in case of Heavy Traffic 60 minutes is enough to filter the driver through red light and to have a 2 second approximation. If a wrong cycle plan is chosen, this is likely cycle plan 5. However, this cycle plan is nearly the same as the correct cycle plan and the green start estimation is still accurate. Also notice that when a possible driver through red light is added, the algorithm needs a few more green observations to give a conclusion.

	LT	HT	LT with red driver	HT with red driver
Average time needed	147 min	52 min	153 min	60 min
Cycle plan 1 chosen	0	0	5	0
Cycle plan 2 chosen	0	0	1	0
Cycle plan 3 chosen	9958	9978	9962	9981
Cycle plan 4 chosen	0	0	1	1
Cycle plan 5 chosen	42	22	31	18

Table 4: Simulation results

4.8 Results of implementation in Portland

Like explained in the beginning of this section, a TomTom trace is a list of GPS coordinates. In Figure 29 we visualize the traces by red dots, so we can see where the cars have driven in the city of Portland. In the figure, the main roads are clearly visible. For our research we zoom in on an intersection with traffic lights.

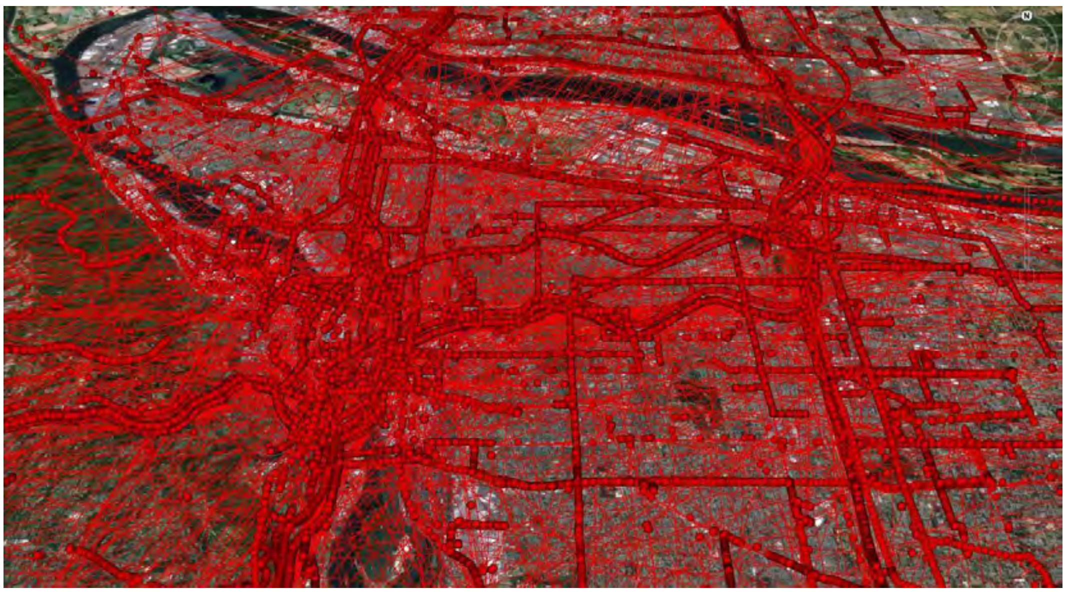


Figure 29: Traces in Portland

4.8.1 Polygons

To analyze if a car has crossed an intersection, we make use of polygons. These are geometric shapes which are closed. We define the positions of the polygons based on GPS coordinates. The longitude describes the east-west position of a point on the Earth's surface and the latitude the north-south position. At each intersection we define several polygons similar to the following rectangle:

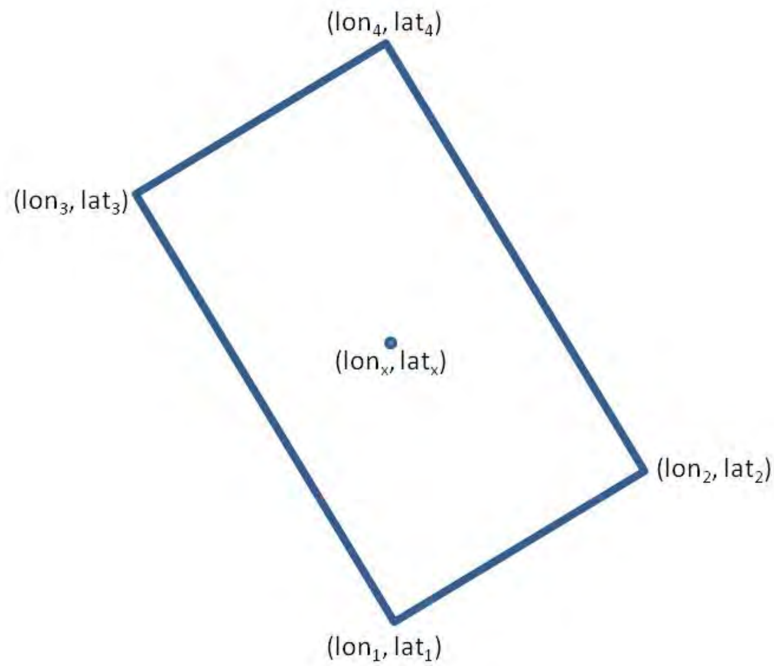


Figure 30: Polygon

Describe point x with the coordinates $(\mathrm{lon}_x, \mathrm{lat}_x)$. The point lies in the polygon if the following inequalities are satisfied:

$$
\begin{aligned}
\mathrm{lat}_1 + \frac{\mathrm{lat}_2 - \mathrm{lat}_1}{\mathrm{lon}_2 - \mathrm{lon}_1}(\mathrm{lon}_x - \mathrm{lon}_1) \leq \quad \mathrm{lat}_x &\leq \mathrm{lat}_3 + \frac{\mathrm{lat}_4 - \mathrm{lat}_3}{\mathrm{lon}_4 - \mathrm{lon}_3}(\mathrm{lon}_x - \mathrm{lon}_3), \\
\mathrm{lon}_1 + \frac{\mathrm{lon}_3 - \mathrm{lon}_1}{\mathrm{lat}_3 - \mathrm{lat}_1}(\mathrm{lat}_x - \mathrm{lat}_1) \leq \quad \mathrm{lon}_x &\leq \mathrm{lon}_2 + \frac{\mathrm{lon}_4 - \mathrm{lon}_2}{\mathrm{lat}_4 - \mathrm{lat}_2}(\mathrm{lat}_x - \mathrm{lat}_2).
\end{aligned}
$$

To fully describe the polygon, we only need the coordinates of three corners. So if we know $(\mathrm{lon}_1, \mathrm{lat}_1)$, $(\mathrm{lon}_2, \mathrm{lat}_2)$ and $(\mathrm{lon}_3, \mathrm{lat}_3)$, the fourth corner can be calculated by: $\mathrm{lon}_4 = \mathrm{lon}_3 + (\mathrm{lon}_2 - \mathrm{lon}_1)$ and $\mathrm{lat}_4 = \mathrm{lat}_3 + (\mathrm{lat}_2 - \mathrm{lat}_1)$.

4.8.2 Implementation of the polygons

Because sometimes traces can be inaccurate, we use bigger polygons than drawn in Figure 15. We define a polygon at each side of the intersection covering all lanes. If for example a car enters at the east polygon and leaves at the west polygon, we know which transition the vehicle has driven. For this car we estimate the time that it has passed the middle polygon. If a trace did not enter and leave at the desired polygons, we do not use the trace. For intersection 2033 (intersection number according to Green Driver) in Portland we draw the polygons as in Figure 31. Green Driver is a company in the United States that provides live updates for traffic light states.

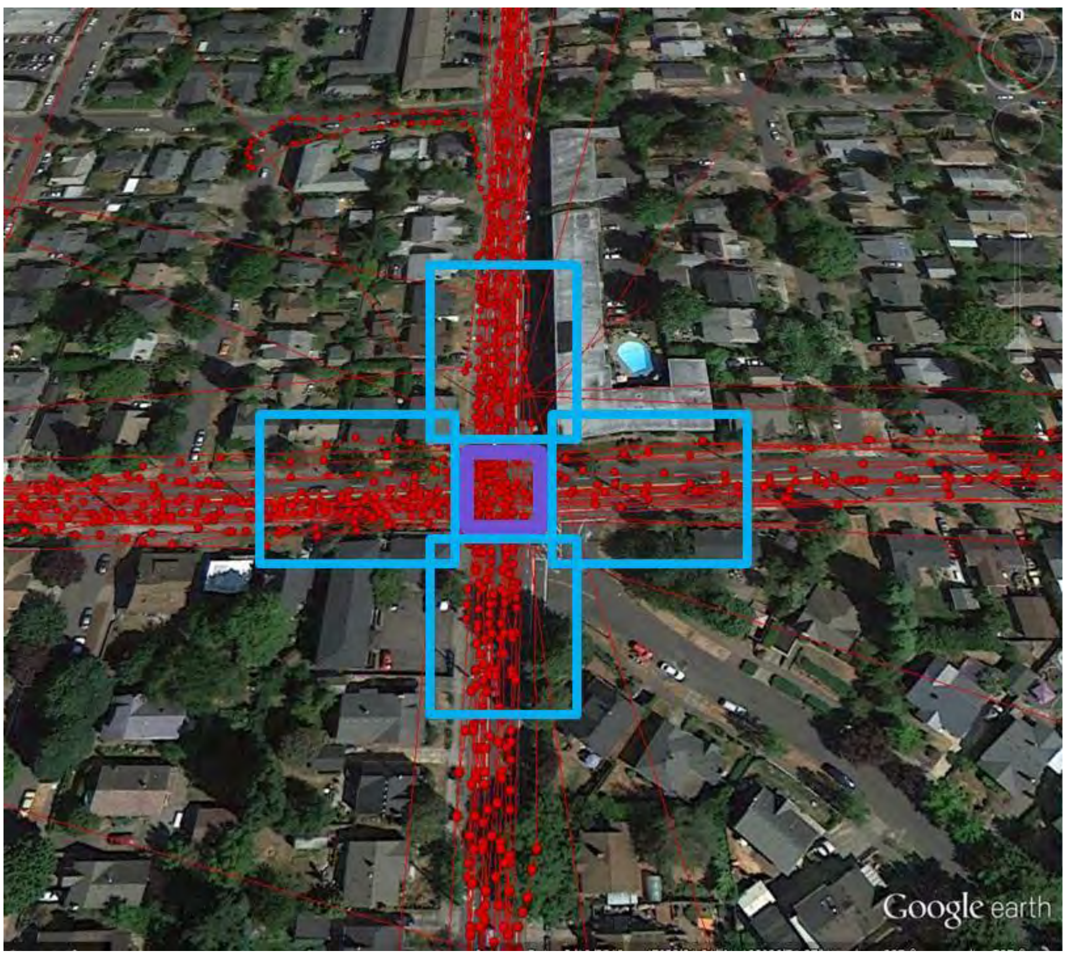


Figure 31: Intersection 2033 with polygons

4.8.3 Cycle plans of intersection 2033

Intersection 2033 is chosen for the implementation, because we observed from the Green Driver data that the traffic lights are statically managed. See Figure 32 for a layout of the intersection.

Figure 32: Phases of intersection 2033

Note that the phase numbers are different than the Dutch standards, which have been used earlier. At this intersection phase 2 and 6 have green light simultaneously, followed by phase 4 and 8. The traffic lights work according to two possible cycle plans. During rush hour the green and red times are (in seconds):

- $G_2 = G_6 = 58$,
- $R_2 = R_6 = 42$,
- $G_4 = G_8 = 38$,
- $R_4 = R_8 = 62$.

Note that within the red times a clearance time of 2 seconds is involved. Outside rush hour the cycle plan is:

- $G_2 = G_6 = 38$,
- $R_2 = R_6 = 32$,

- $G_4 = G_8 = 28$,
- $R_4 = R_8 = 42$.

Moreover, the traffic light has exactly the same states for every single day. So we can use the traces from an entire week to determine the starting time of its cycle. In this example we need the traces from an entire week, because the percentage of live communicating TomTom devices is much lower than in the Netherlands. Also a lot of traces are too inaccurate to use for our implementation.

4.8.4 Results

We will try to approximate the start of the green/red cycle of intersection 2033 in Portland during evening rush hour (3:33:19 PM - 6:45:37 PM, local time) and outside rush hour (8:49:10 AM - 3:31:11 PM, local time). As input for the algorithm we use the green observations analyzed from TomTom traces for 11-03-13 until 17-03-13 within the given time slots. Because the data is restricted we only apply the algorithm once, regardless of the accuracy of the approximation.

See Appendix B.1 and B.2 for the output of the algorithm. If we compare the estimation of the green start of phase 2 and 6 with the real green start from the Green Driver data, the algorithm approximates the green starts in respectively two and three seconds. Also in both implementations we see that the driver through a red light is filtered successfully.

Although the sample of real statically managed traffic lights and TomTom traces is still small, these results seem to indicate that the method is working properly.

4.9 Conclusions

For a statically managed traffic light, we can calculate how many cycles we probably need to approximate the cycle plan with a given accuracy. The expected number of needed cycles strongly depends on the fraction of cars with live communicating TomTom devices and the traffic intensity. An algorithm has been constructed which determines the best fitting cycle plan and approximates the current state of the traffic light. The algorithm is also able to filter drivers through red light. Simulations have verified that the algorithm almost always filters the driver through red light and chooses the correct cycle plan. The algorithm needs on average 60 minutes of TomTom traces to determine the cycle for Heavy Traffic and 150 minutes on average for Light Traffic. Also a real implementation in Portland has determined the cycle correctly and approximated the state of the traffic light within three seconds.

5 Behavior of dynamically managed traffic lights

In this section we focus on the behavior of dynamically managed traffic lights and its impact on the quality of our predictions. For this research, traffic light data in Helmond (the Netherlands) and Portland (Oregon, USA) is available.

First we study the situation in Helmond. These traffic lights are known as very dynamic, which makes it difficult to determine the distributions of the green and red times. In the second part, we study the traffic lights in Portland. For this area, we have data for about 850 signalized intersections. This will be more useful to analyze different types of traffic lights.

5.1 Helmond

First we analyze three intersections in Helmond. Helmond is a city in the province of North Brabant in the southern Netherlands, which has a population of approximately 90000. The data is again provided by PeekTraffic. The intersections are directly connected and lie in the center of the city, as illustrated in Figure 33. The signal group numbers are assigned according to the Dutch standards. The main road is going from east to west (and west to east), which causes the most traffic for signal group 2 and 8. To the west the road goes to the city of Eindhoven, where many people are working. So during morning rush hour most traffic will go to the west and in the evening rush hour most traffic will go to the east.
The traffic lights are perhaps of the most dynamic type in the world. For example, fire trucks from a nearby station can influence the behavior of the traffic lights. In case of emergency, the fire trucks can request a green light immediately. Those kind of exceptions will make it hard to give good predictions.

5.1.1 Description of PeekTraffic data

The data contains observations from 16 January 2012 until 22 January 2012. For the entire week we can see every changing state of the traffic light and all the times when a vehicle has made a request (by an induction loop). From the data we can determine accurately the green, red and orange times of the traffic light. Furthermore we know which approaches have received a request.

PeekTraffic also provided us with the phase diagram of intersection 101. This describes in which sequence the signal groups can have green. The diagram is given in Figure 34. Here P indicates that a group gets green if a request has been done. If the signal group has A, it will have green if the conflicting approaches are empty. The PF are special cases for firetrucks that have priority.

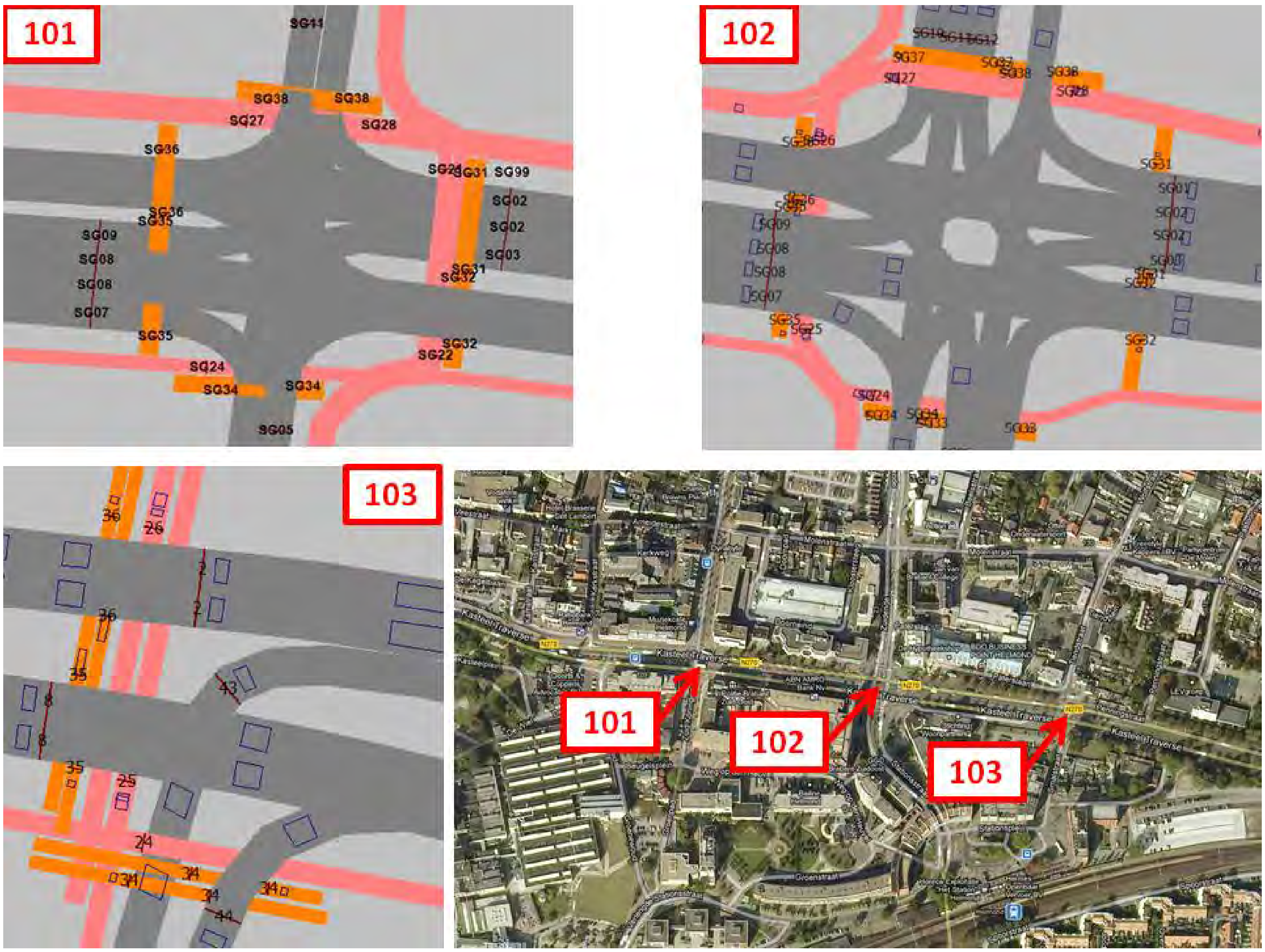


Figure 33: Three signalized intersections in Helmond

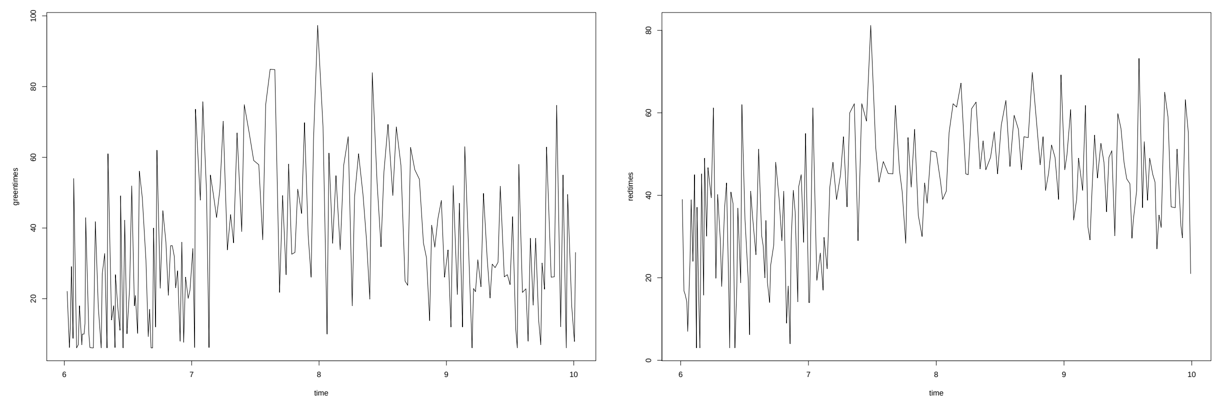
	1	2	3	4	5	6	7	8	9	10	11	12
02		P	P			PF						
03			P			PF						
05					P							
07	A	P	A				PF					
08	P	P					PF					
09	P						PF					
11					P							
21				P								
22				P								
24	P	A										
27		P	P									
28		P	P									
31				P								
32				P								
34	P	A										
35				P								
36				P								
38		P	P									
99		P	P									

Figure 34: Phase diagram of intersection 101

5.1.2 Analysis of the traffic light behavior

For our cause the red and green times describe the behavior of the traffic light. The main questions are: how are the times distributed and how does the time of the day influence the distribution? First we analyze the behavior of signal group 2, which is for the main road. Since the traffic intensity for this signal group is high, these green times will be larger than for other signal groups. Especially during rush hour, we expect that the times have a small variation (because signal groups are rarely skipped and the maximum specified green times occur more often).

In Figure 35, we see how the green and red times evolve from 6 AM till 10 AM on a normal work day. On this day the times vary between 6 and 100 seconds. Around 7 AM the green and red times increase because the traffic intensity gets higher. Also the fluctuation in the times is decreasing. So we may say that the traffic light is more dynamically managed outside the rush hour. Therefore, during rush hour the traffic light will be more predictable.

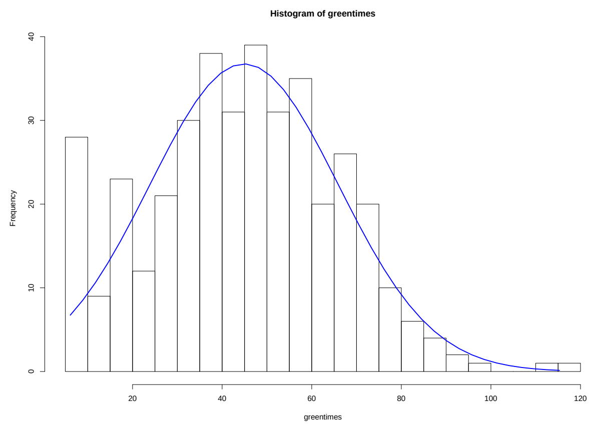


(a) Green times (b) Red times

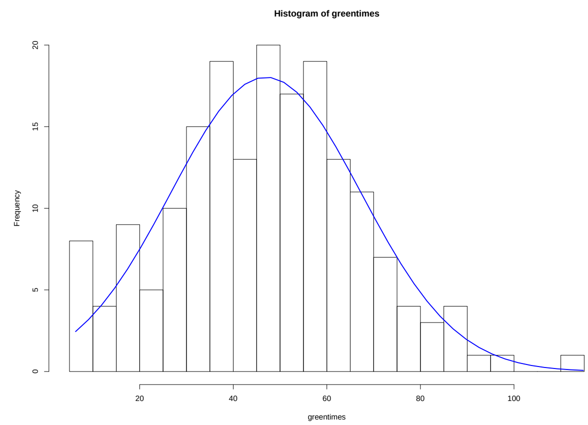
Figure 35: Time line from 6 AM till 10 AM, signal group 2 intersection 101

To see how the green and red times are distributed, we plot histograms. In Figure 36 the green times for the entire work week are plotted. The normal distribution fit is presented by the blue line. We see that during rush hour, the green times appear normally distributed. Especially from 7:30 AM till 8:30 AM, the green times are more normally distributed than from 7:00 AM till 9:00 AM (because the small green times occur more often in the last case and this makes the distribution more asymmetric). So the normal distribution fit is practically usable for the phase prediction model in Section 2 during rush hour.

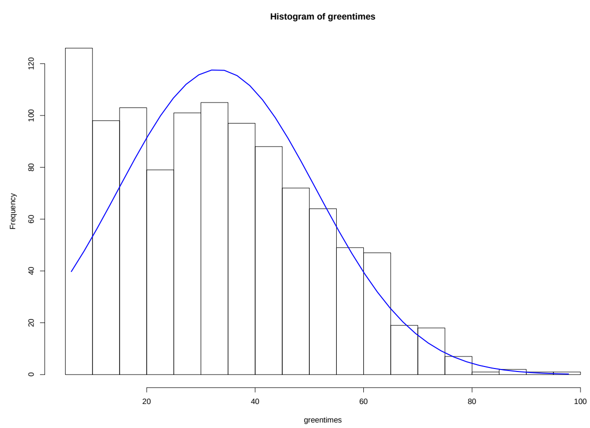
But outside rush hour, small green times occur very often. So the times are clearly not normally distributed, also the normality tests imply this conclusion. If the traffic intensity gets lower, the queues become smaller and the traffic light will decide to switch to red earlier. It is also possible that the traffic light has a maximum green time of 120 seconds. These extreme times make it very difficult to make accurate predictions. Also the time line in Figure 35 indicates that there is no correlation in successive green times.



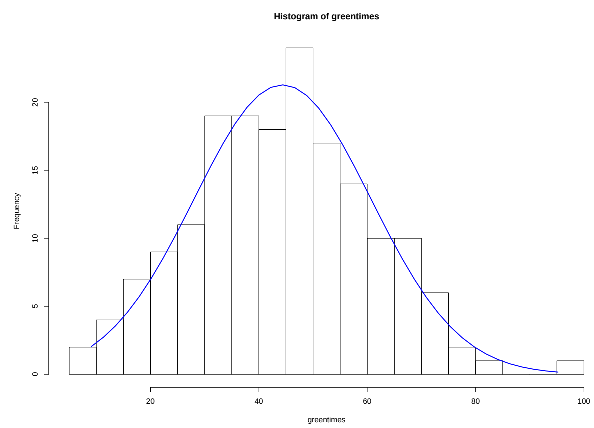
(a) 7:00 AM - 9:00 AM



(b) 7:30 AM - 8:30 AM



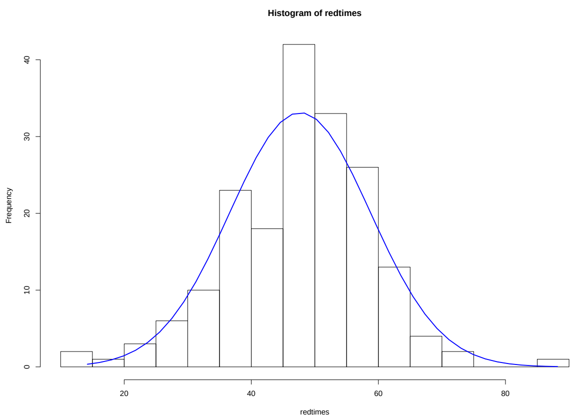
(c) 9:30 AM - 14:30 PM



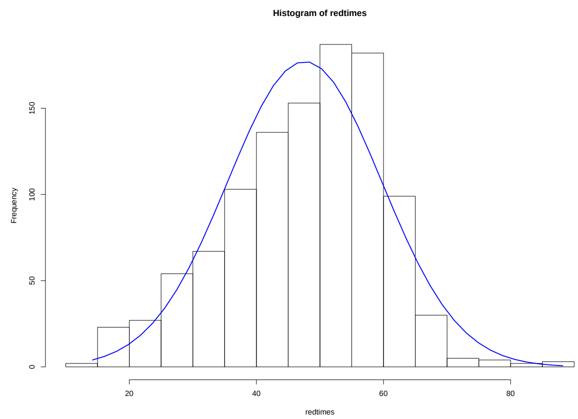
(d) 5:00 PM - 6:00 PM

Figure 36: Green times signal group 2 with normal distribution fit, intersection 101

In Figure 37 we see that the red times seem normally distributed both during rush hour and outside rush hour. This can be expected because the red time of signal group 2 is the sum of green times of the other groups. In probability theory, we know that the sum of independent random variables converges in distribution to a random variable with a normal distribution. Compared to the green times of the traffic light, the variation of the red time is smaller.



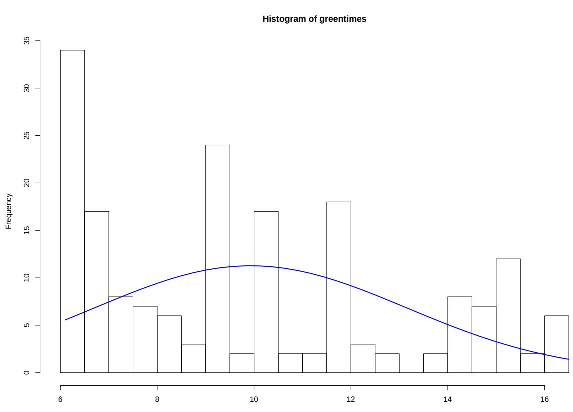
(a) 7:30 AM - 8:30 AM



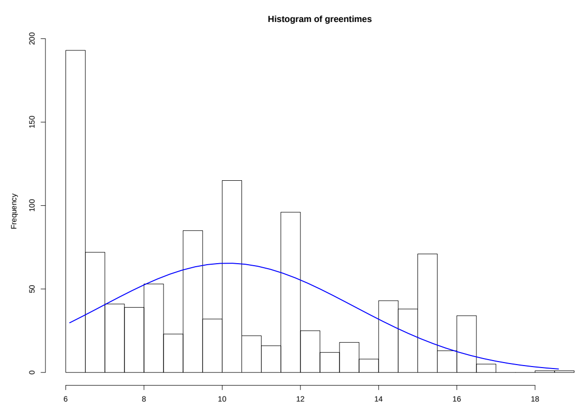
(b) 9:30 AM - 14:30 PM

Figure 37: Red times signal group 2 with normal distribution fit, intersection 101

Signal group 5 is one of the sideways of intersection 101, so the green times will be much lower than for signal group 2. In Figure 38 the green times are plotted. The histograms indicate that the green times of group 5 are not normally distributed. The behavior is more or less the same for different time slots. However, such small green times make it hard to predict when this light will be green.



(a) 7:30 AM - 8:30 AM



(b) 9:30 AM - 14:30 PM

Figure 38: Green times signal group 5 with normal distribution fit, intersection 101

In Figure 39 we analyze the green times of signal group 2 for intersection 102 during morning rush hour. Although the situation seems to be the same as for signal group 2 in intersection 101, the distribution of the green times is totally different. Here it seems that around 60/70 seconds some maximum green time is specified, such that we see a big peak in this area.

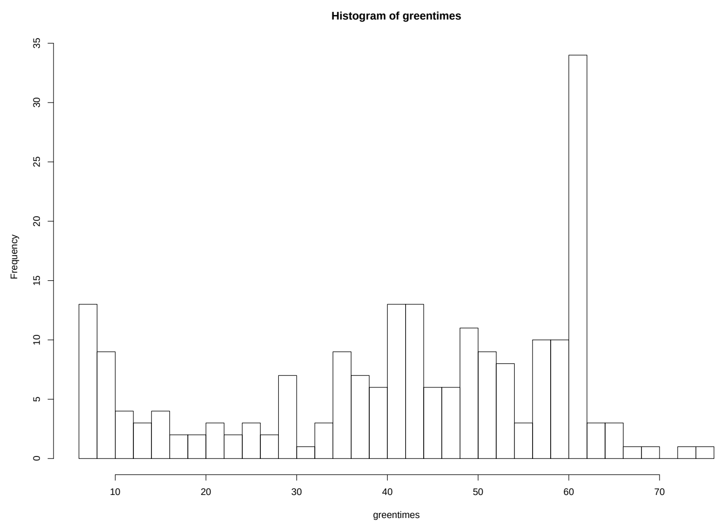


Figure 39: Green times signal group 2 during morning rush hour, intersection 102

So we conclude that the distributions of green and red times can be different for each signal group, each intersection and each time of the day. This makes it hard to fit appropriate distributions and parameters, especially if we do not have all specifications of the traffic lights. It is also not desirable to do detailed analysis for each single intersection. So in this project, we want to make a model which is implementable for all possible traffic lights. In Section 2 we used the normal distribution as example, but the PeekTraffic data shows that the normal distribution is perhaps only usable for signal group 2 and 8 during rush hour.

That is why a more empirical approach is easier to implement for the TomTom device. Also a discretization of the green and red times will make the calculations easier and faster. The idea is to sample the green and red times and use their empirical distributions for the necessary probability calculations. See Section 7 for more explanation about the implementation decisions.

5.1.3 Traffic light predictions

In this example we make predictions for intersection 101 during evening rush hour. In the previous subsection we concluded that for this intersection during rush hour, the green

and red times look normally distributed. So we apply the theory of Section 2.5 to make predictions.
Based on our normal distribution fit for the green and red times, take the following input for the prediction model:

$a = 10$,
$\mu_G = 41.6$,
$\sigma_G = 17.2$,
$\mu_R = 54.1$,
$\sigma_R = 9.4$,
$r = 0$.

In Figure 40 we can see the probability that the light is green for $0 \leq t \leq 200$.

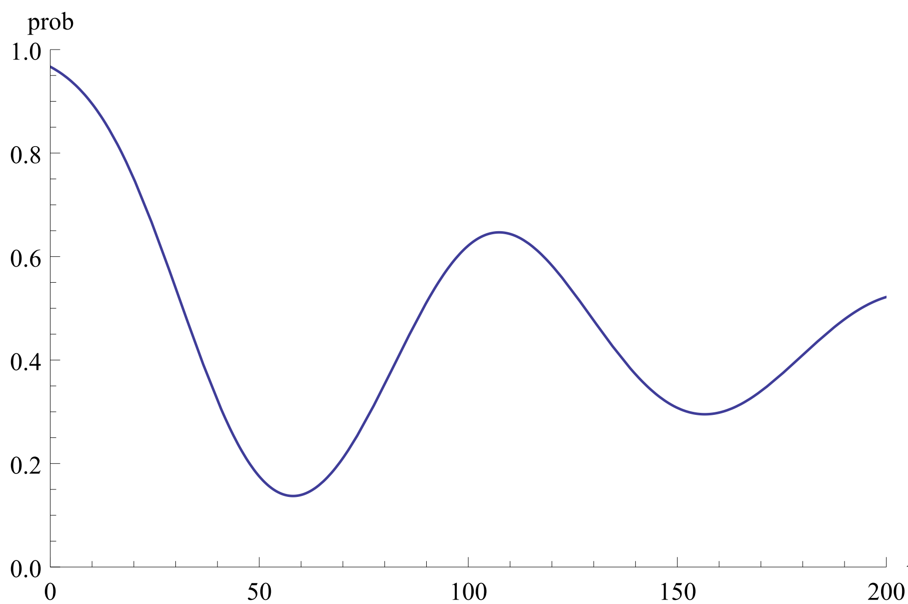


Figure 40: Traffic light prediction intersection 101 for signal group 2, evening rush hour

Due to the high variance, the phase and cycle length of the traffic light is unpredictable. This leads to a fast convergence of the probability for green light. The predictions are perhaps only useful for the first minute. Because these traffic lights are one of the most dynamically managed, we can see these predictions as a worst case scenario. Hopefully other traffic lights are less dynamic and the predictions are more reliable, which would make it possible to do distant future predictions for networks of traffic lights.

5.2 Portland

Portland (Oregon) is a large city in the northeast of the United States of America. The city has a population of approximately 600000 and the metropolitan area has about 2.3 million inhabitants. For this city, we received live data from Green Driver for a lot of traffic lights. This was very useful to study the behavior of dynamically managed traffic lights.

5.2.1 Description of Green Driver data

Green Driver is a company in the United States that provides live updates for traffic light states. In Portland we know of 1750 existing signalized intersections, 850 of which are connected to Green Driver. The traffic lights can be statically or dynamically managed. On the web server of Green Driver, we can find the locations and phases of the traffic lights. A phase consists of one or more signal groups. The phase numbers are different than the Dutch standards we studied in Helmond. The locations are given by the red dots in Figure 41. Each update we receive, describes which phases have green at the moment and which phases have done a request. Green Driver also provides predictions for the remaining time of the current phase. These are likely based on the distributions of the first green or red time (see Section 2.4).

So we are able to see how 850 traffic lights in Portland are behaving and we can test our prediction model on these intersections. Also networks of traffic lights exist to study the expected travel time through a network. We will study networks of traffic lights in Section 6.

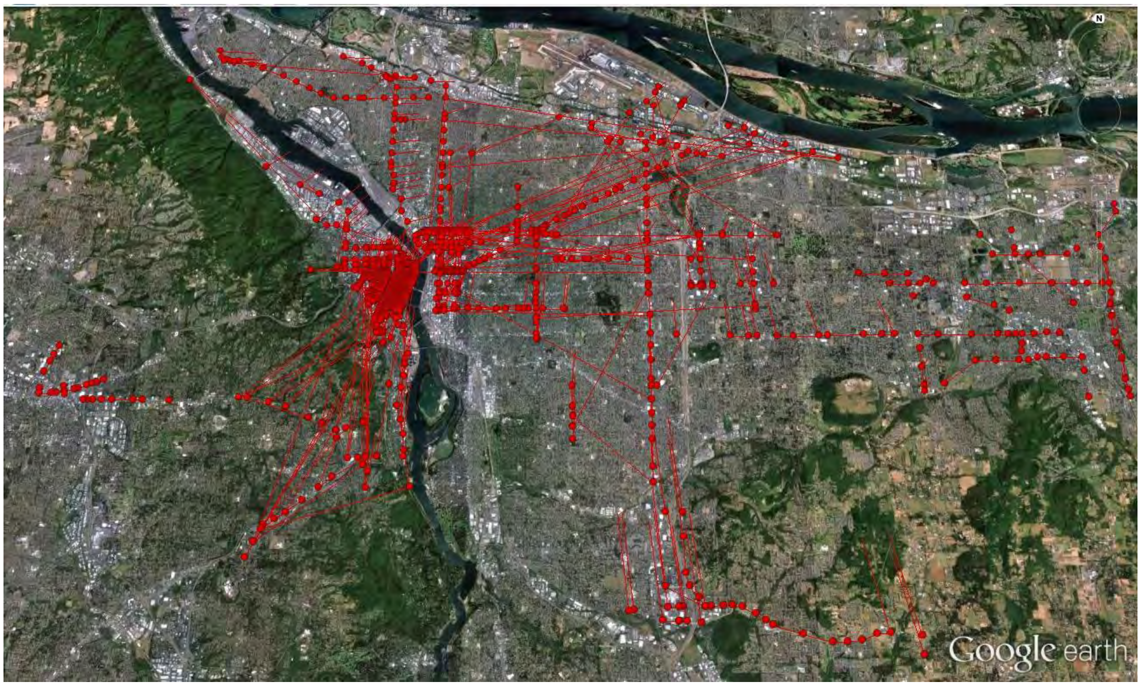


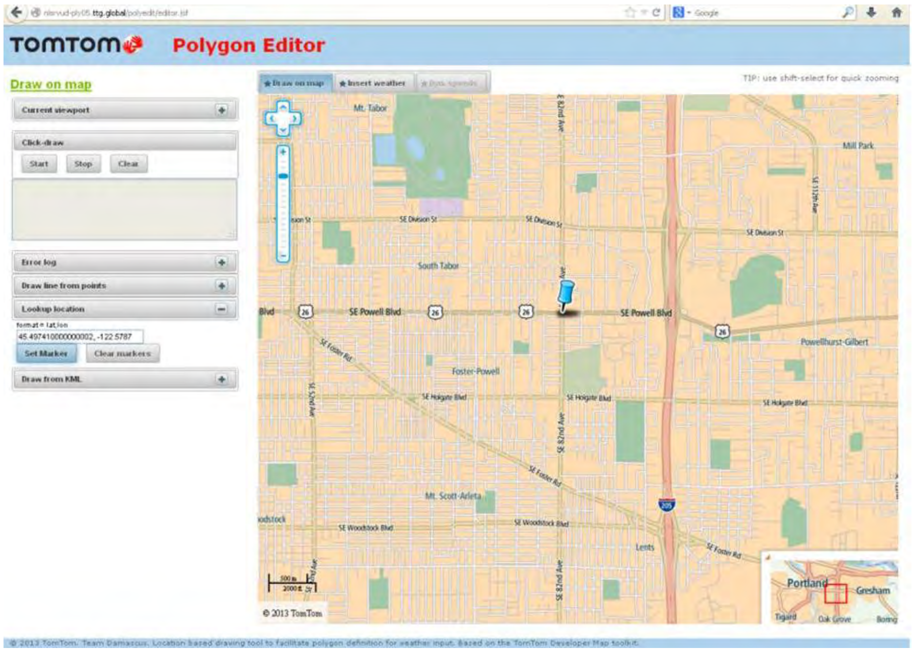
Figure 41: The online traffic lights in Portland

5.2.2 Analysis of the traffic light behavior

In downtown Portland, a lot of one-way roads are constructed. For these roads the traffic lights seem statically managed. Also in Section 4 we analyzed a statically managed traffic light. On the bigger roads, the traffic lights are more dynamically managed. Especially traffic lights which involve a passing tram have more variations in green times, because public transportation receives priority.

We start our research with intersection 4110. The intersection is illustrated in Figure 42. The traffic lights are dynamically managed, located in the outskirt of Portland and the traffic intensity is high. Many companies and restaurants are located in this area. If we want to go from the east to the west, we need phase 6 if we cross the intersection.

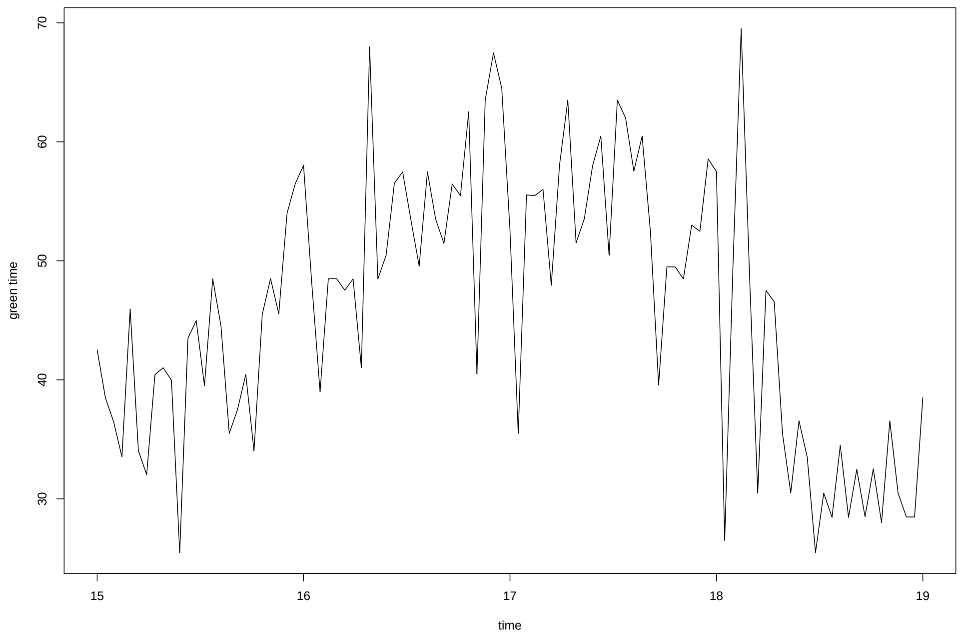
The time line of phase 6 for 3 PM till 7 PM is given in Figure 43. Here we see that during rush hour (4 PM till 6 PM), the green and red times are clearly larger. Moreover, the fluctuation is much lower than for the intersections we analyzed in Helmond. Also in the histograms of Figure 44 we see that the variation is much smaller. Unfortunately here the red times are not normally distributed. Outside rush hour we see that the times look normally distributed and the variations give us hope that the predictions are still reliable (see Figure 45).



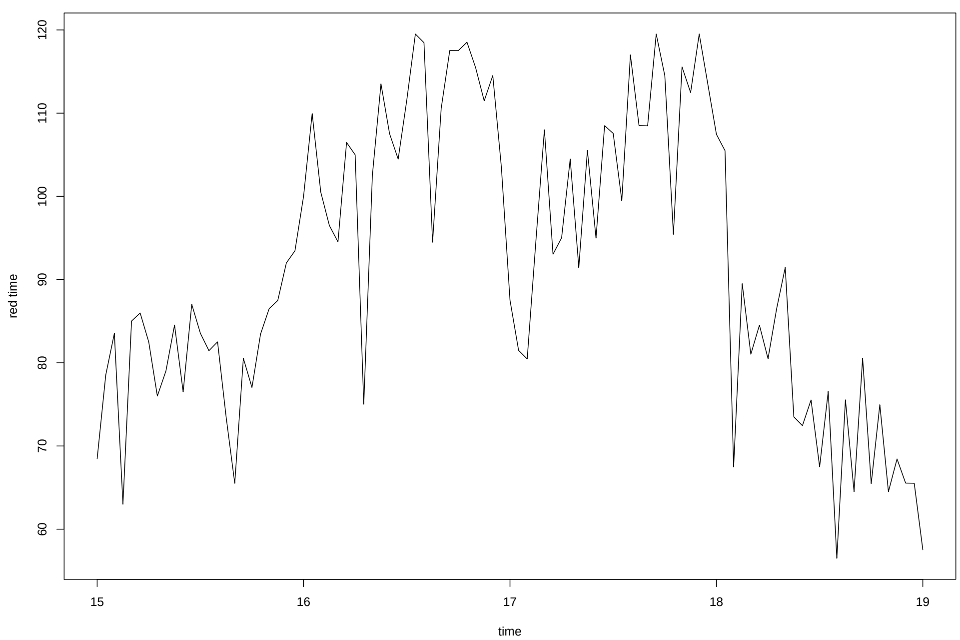
(a) Location of intersection 4110

(b) Phase 6

Figure 42: Intersection 4110, Portland

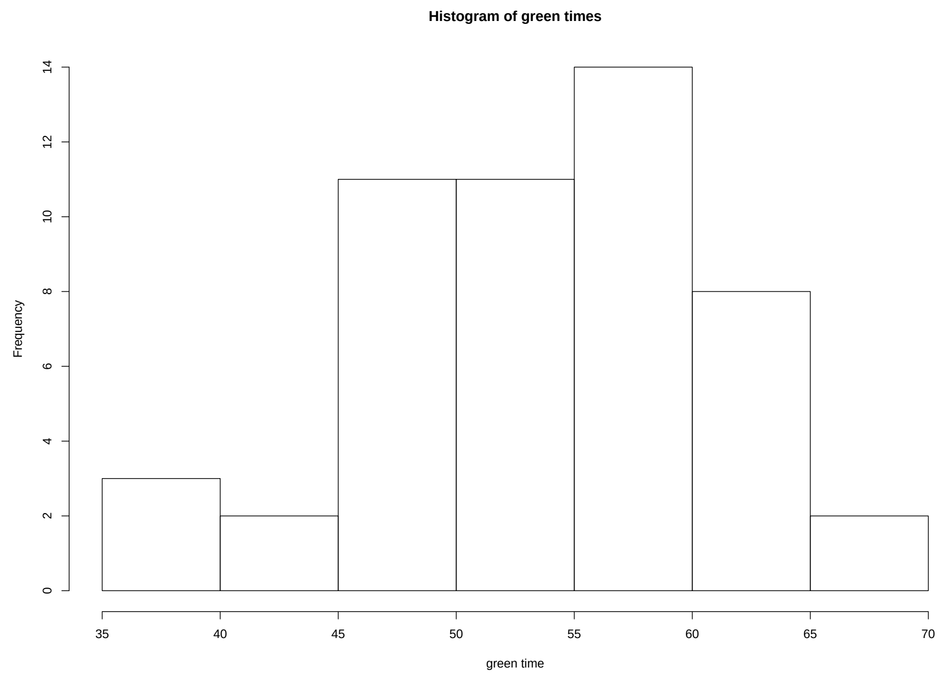


(a) Green times

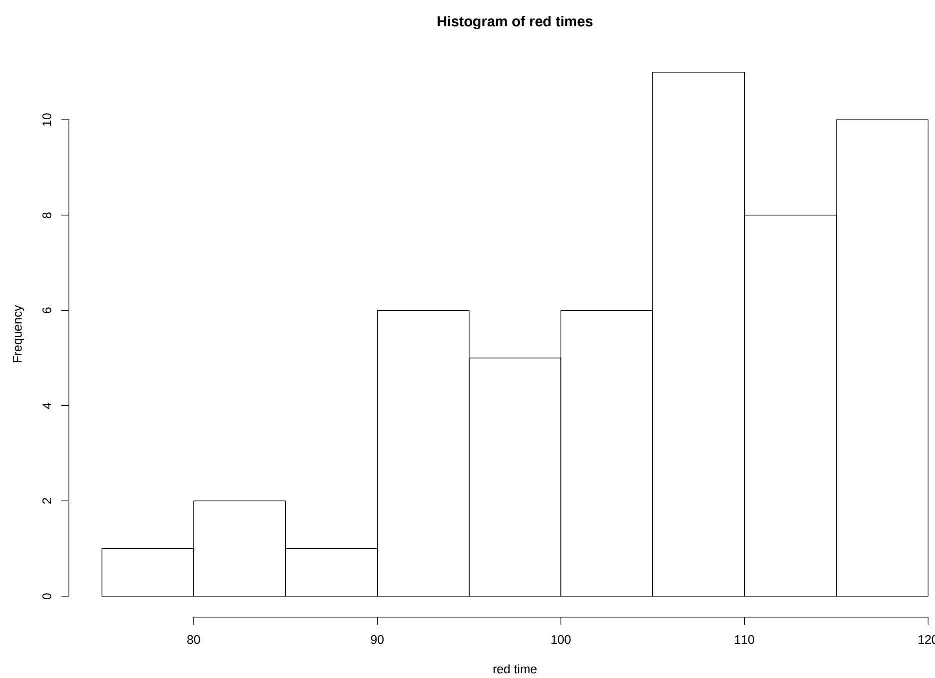


(b) Red times

Figure 43: Time line from 3 PM till 7 PM, signal group 6, intersection 4110

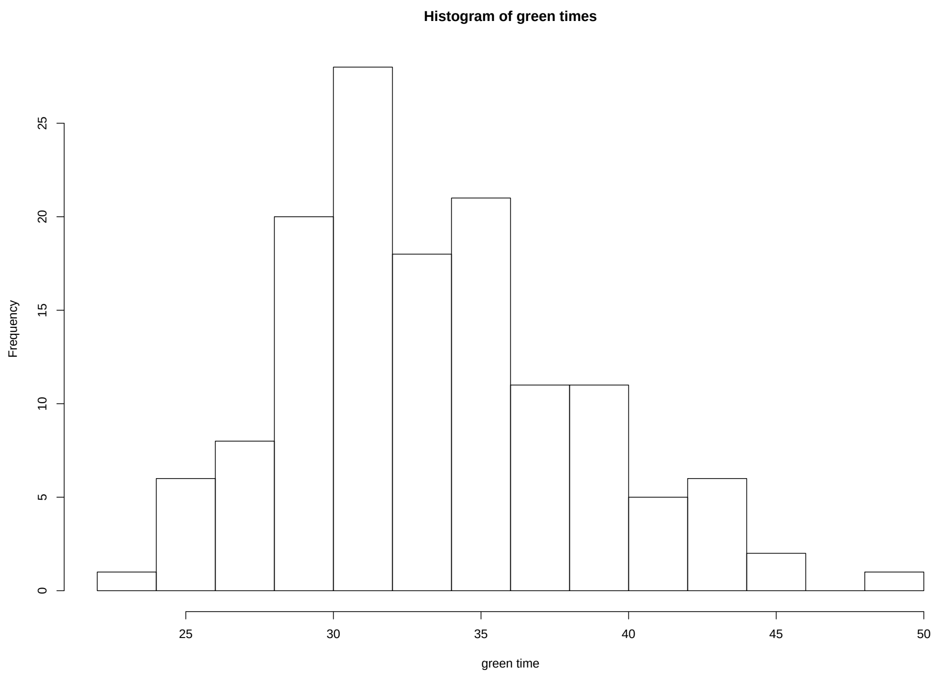


(a) Green times

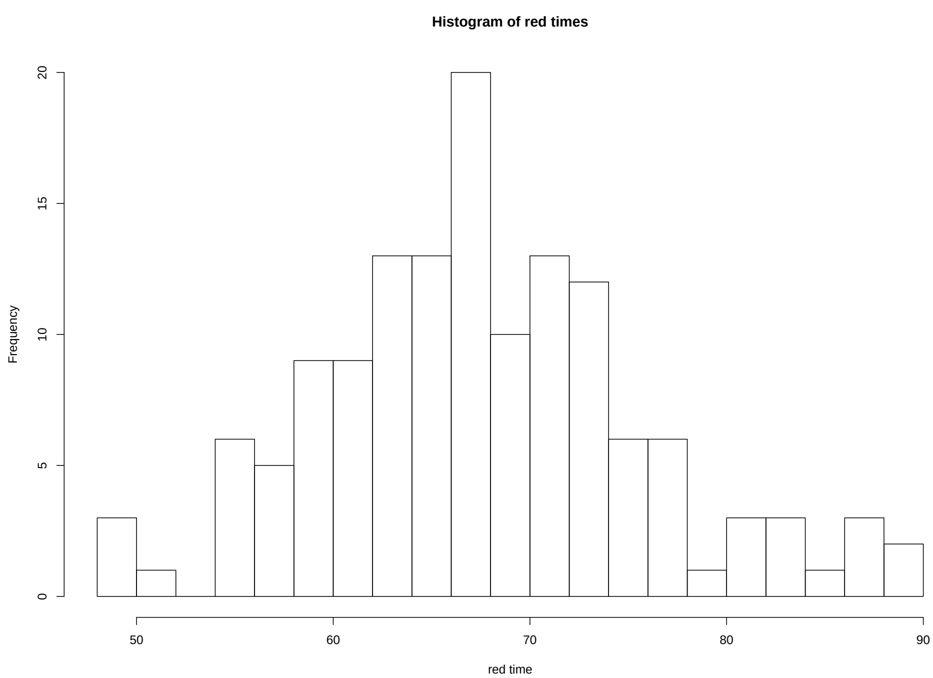


(b) Red times

Figure 44: Evening rush hour, intersection 4110



(a) Green times



(b) Red times

Figure 45: 9:30 AM till 14:30 PM, intersection 4110

5.2.3 Traffic light predictions

First we want to give predictions for intersection 4110 during morning rush hour. As input for the phase prediction model, we need a profile of the traffic light (the way the traffic light behaves at the moment). For this cause, we can give a list of the green and red times of the traffic light. Again we model orange light as green. For example, during morning rush hour (starting at 7:31:41 AM) the light had the following green times:

43, 34, 27, 33, 26, 31, 31, 35, 40, 36, 44, 30, 33, 29, 26,
31, 32, 46, 43, 28, 40, 26, 38, 28, 37, 37, 35, 30, 33, 36.

Between these green times, the following red times have occurred:

57, 63, 59, 65, 46, 65, 70, 66, 59, 69, 69, 72, 71, 69, 51,
68, 58, 72, 64, 60, 60, 57, 66, 58, 57, 52, 62, 70, 65, 64.

Note that the traffic light in this situation is less dynamic than outside rush hour and more dynamic than during evening rush hour (see Figures 44 and 45). In Figure 46, the predictions are plotted which used the lists of green and red times above. In this situation the light for phase 6 was green for 6 seconds at $t = 0$ (or in the context of Section 2: $r = 0$ and $a = 6$). We see that the plot starts with an interval where the light surely stays green according to our prediction model. After this interval the probability drops which indicates that the light can turn red any moment. Hereafter we know for 100% that the light is red (according to our prediction model), etcetera.

For the visualization, we can translate the probabilities to a lemon green, brown and pink red time window. We make the window lemon green where we predict that the light will be green, the lemon green intervals are smaller than the average green time of the traffic light. The lemon green intervals are located at the peaks of the probability curve (or the local maxima). At the local minima, we predict that the light will be red and visualize this by pink red. In the areas in between, the light is unpredictable and we visualize this by brown. We do not want to make the lemon green intervals too large, such that we try to send the vehicle toward the middle of the green time. We hope that at this time the queue at the traffic light will be dissolved and the light will not turn red any moment.

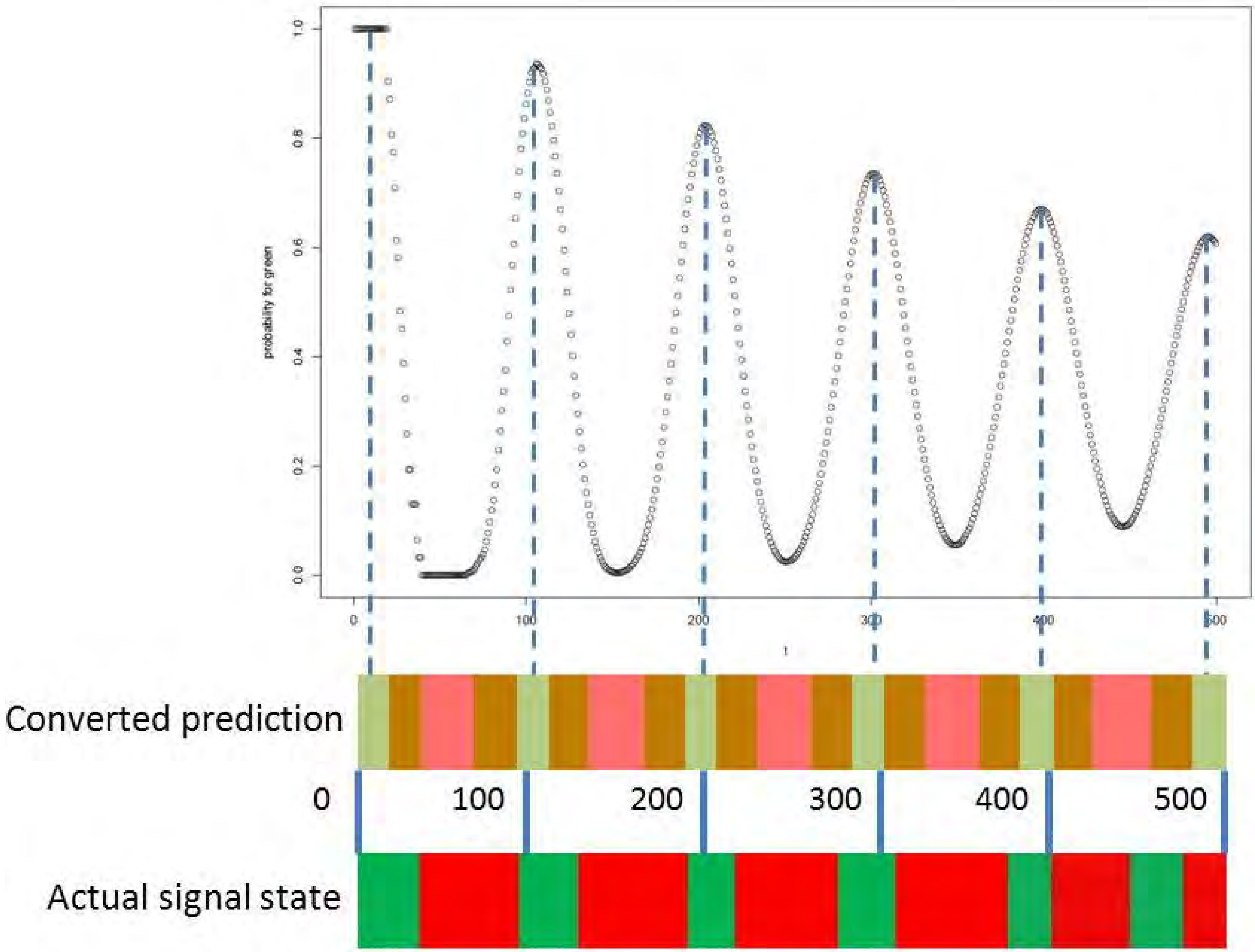


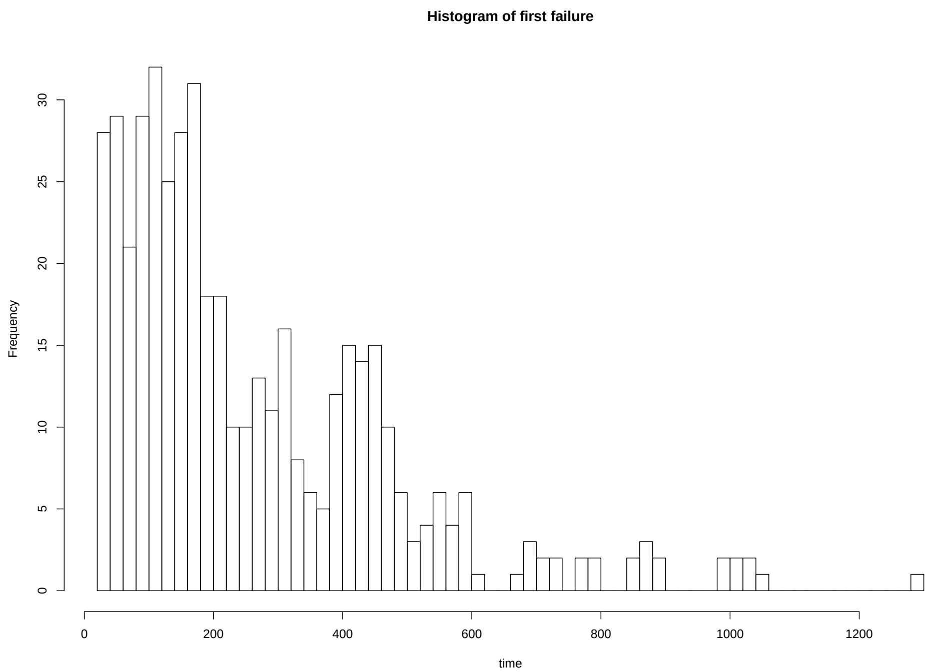
Figure 46: Visualization of the predictions and comparison with reality

The converted prediction in Figure 46 gives us a way to measure how our predictions perform. We can compare the predictions with the real green and red time window (the actual signal states). In this case we see that up to 400 seconds, the predictions are correct. Hereafter, a shorter cycle occurs and prediction becomes wrong.

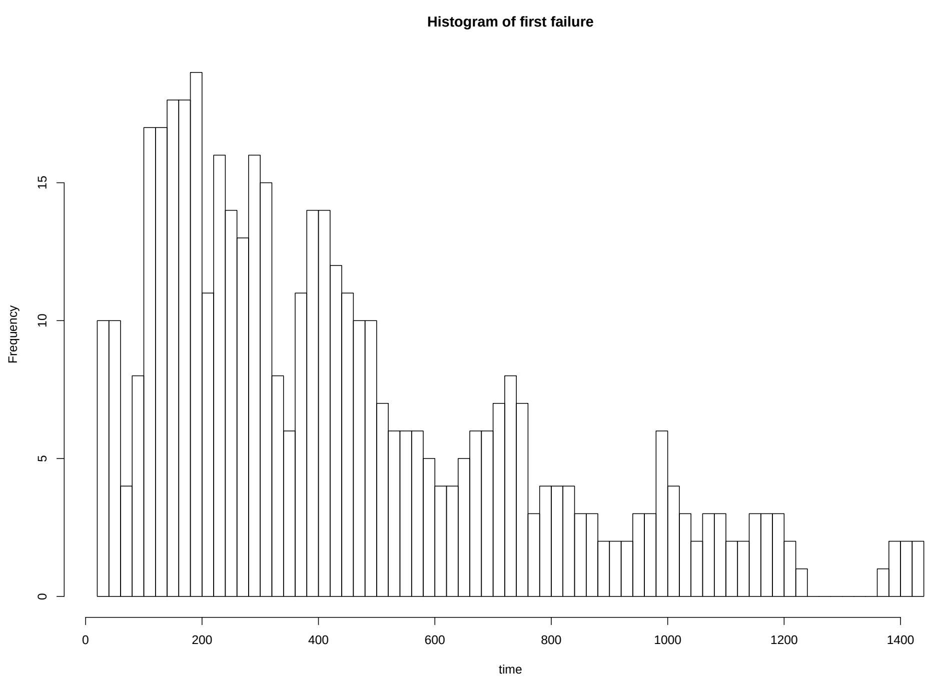
We want to indicate how long the predictions run parallel. When the lemon green intervals do not match the real green times for the first time, we indicate this by the first failure. It is possible to define a match in multiple ways (or in other words: we can define different performance measures). We can demand that during 100% of the lemon green interval the actual signal state is green, or we can also be satisfied if the actual signal state is green during the majority of the lemon green interval. Also the size of the lemon green intervals is important. If the size of the lemon green intervals increases, the probability that the light is actually red during this interval also increases. So the first failure will occur earlier. If we make the lemon green interval too small, it will be difficult for the driver to arrive at the traffic light during this interval.

First we analyze how the predictions are performing for intersection 4110 during morning rush hour (7:35:41 AM till 9:22:21 AM). For this cause, we made a time line of around 7000 seconds for the real green and red phases. For times 0, 10, 20, ..., 5000 we did our predictions and compared them with the real traffic light states. With this method, we consider all possible scenarios for the traffic light states. For each prediction we indicate when it fails and plot the first failures in a histogram.

We start with a lemon green interval which is 50% of the average green time. In Figure 47, the histograms are plotted of the first failures. In (a), the first failure is given if during 100% of the lemon green interval the actual signal state has to be green and in (b) we are satisfied if at least 80% of the lemon green interval is correct. The 100% case is the most strict, such that the first failure will occur often at the beginning. Accidentally, the first failure can occur around 20 seconds. But it is also possible that the predictions are correct for about 1300 seconds. The average first failure occurs around 250 seconds. If the lemon green interval should be at least 80% correct, the average is about 400 seconds. If we are lucky, the predictions can be correct up to 1400 seconds and in the worst case after 23 seconds the predictions can already fail.



(a) 100% of interval has to be correct



(b) 80% of interval has to be correct

Figure 47: Histograms of first failure for 50% of the average green time, intersection 4110

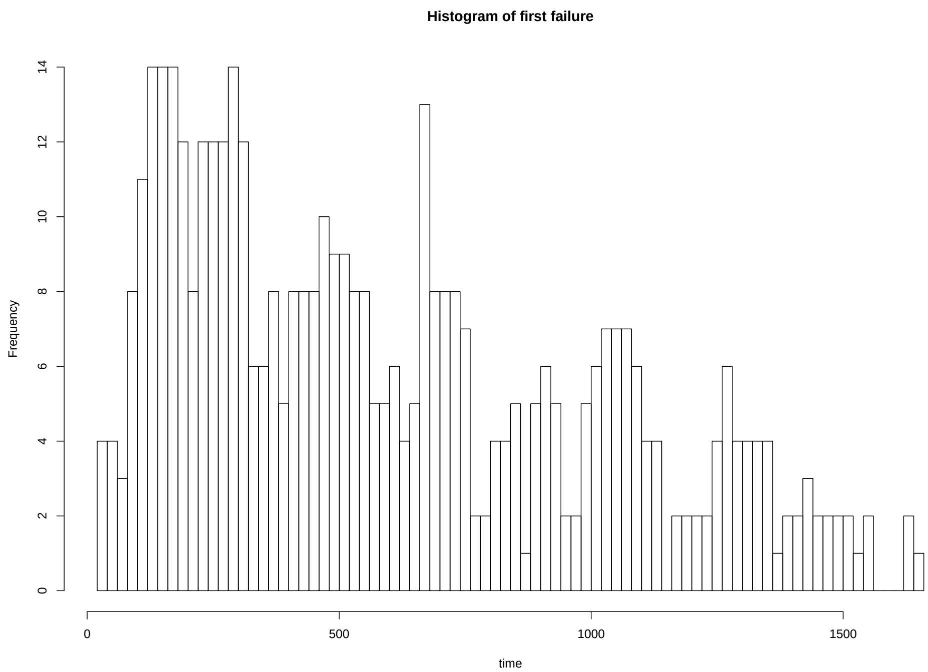
When we change the lemon green interval to 25% of the average green time, the predictions will be better. See Figure 48 for the results. If 100% should be correct, the average first failure time is about 500 seconds. In the best case the predictions will be correct up to 1600 seconds and if we are unlucky, the first failure is after about 25 seconds. If at least 80% of the lemon green interval has to be correct, the first failure occurs around 27 seconds in the worst case. If we are lucky the predictions are correct for 1600 seconds and the average performance is around 700 seconds.

To indicate that the predictions run parallel to the reality (the predictions are not shifting largely beside the actual signal states), a size of 1 second for the lemon green can be used. This performance measure will only indicate whether the signal state in the middle of the predicted green interval (the local maxima of the probability curve) is indeed green. We see the results in Figure 49. The average is about 750 seconds. If we are lucky the predictions will be correct up to 1700 seconds and in the worst case after 80 seconds the predictions will fail.

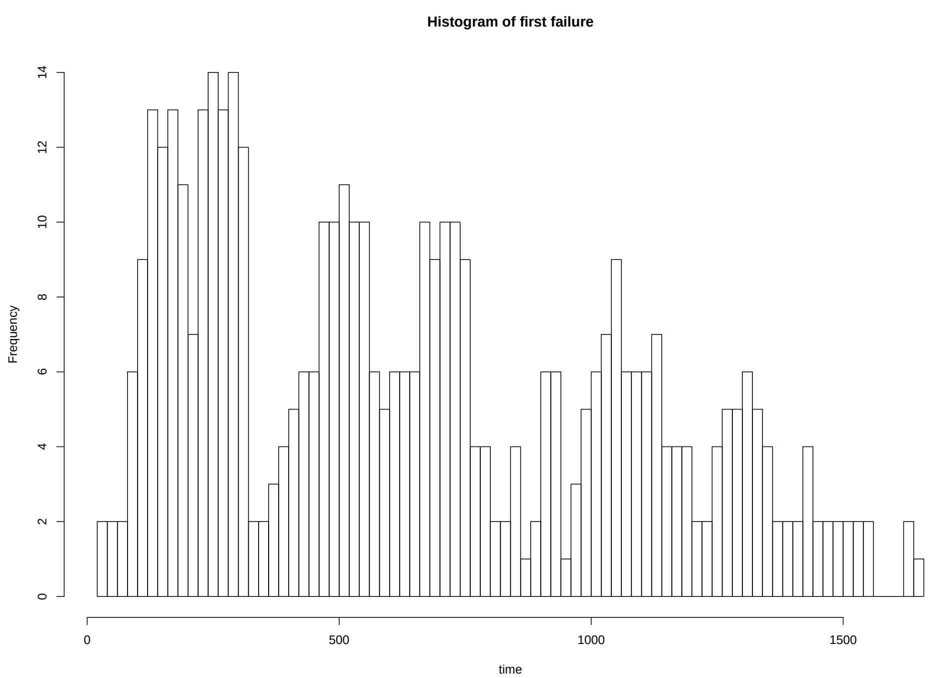
For the TomTom implementation, we are mainly interested in the tail on the left side of the histograms (also called the starting tail). The tail tells us how often we should update the predictions to achieve a desired confidence. Denote by $F(t)$ the probability that the predictions will fail before time t. To have a 100(1-α)% confidence, we need a time t_α such that $F(t_\alpha) = \alpha$. For the different performance measures, Table 5 shows the required update periods to achieve 99.9% and 95% confidence. To guarantee the stated performances, it is essential that we receive all updates correctly. So it may not happen that a large gap occurs between two successive updates of a traffic light. Hence, every changing state of the traffic light should be received within the chosen update period.

Performance measure	99.9% confidence	95%confidence
50% of average green and 100% correct	20 sec	35 sec
50% of average green and 80% correct	23 sec	65 sec
25% of average green and 100% correct	25 sec	107 sec
25% of average green and 80% correct	27 sec	121 sec
1 second interval	81 sec	136 sec

Table 5: Updating period to reach given confidence, intersection 4110



(a) 100% of interval has to be correct



(b) 80% of interval has to be correct

Figure 48: Histograms of first failure for 25% of the average green time, intersection 4110

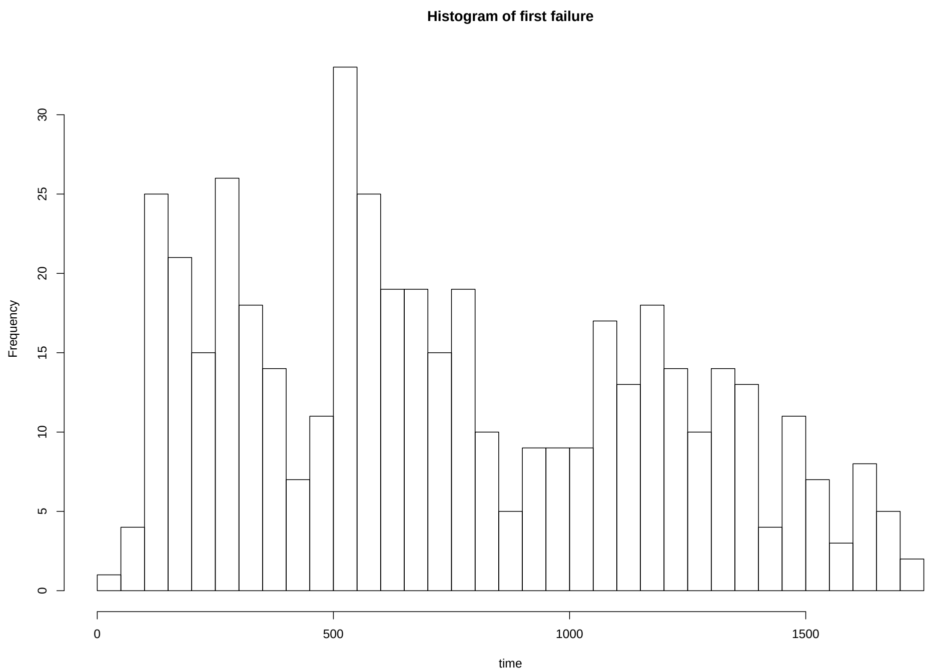


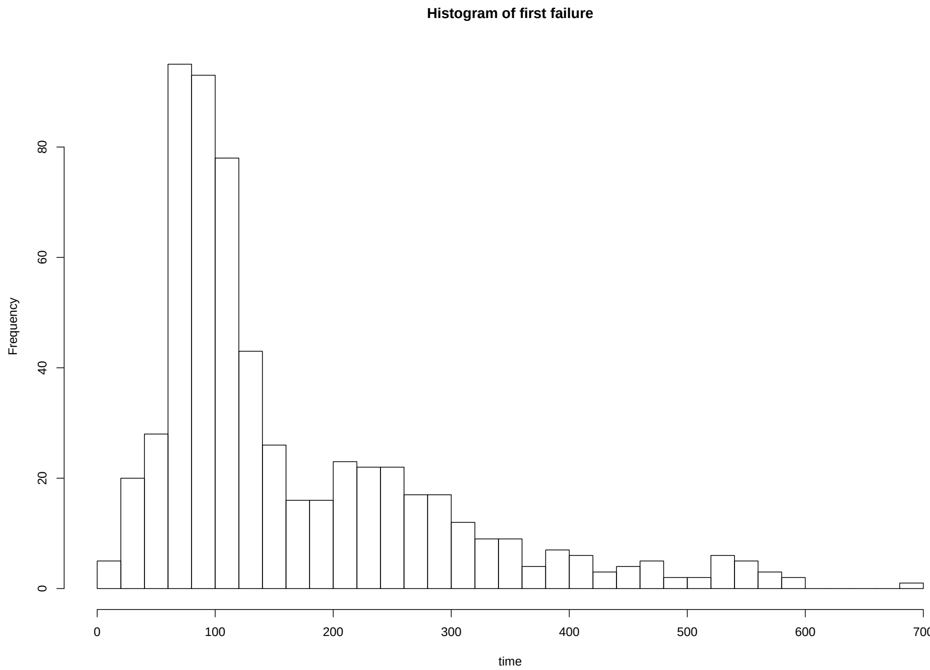
Figure 49: Histogram of first failure for 1 second interval, intersection 4110

In Figure 50 we see how the predictions are performing during the night (1:38:21 AM till 4:45:01 AM). Because the traffic light is less predictable during the night, we see that the first failures occur earlier. Most predictions are already wrong after 100 seconds. Perhaps only the situation where the network was empty, has resulted in good performances. Table 6 shows the analysis of the starting tail of the histograms. So the traffic light phase predictions should be updated very frequently to guarantee good performances.

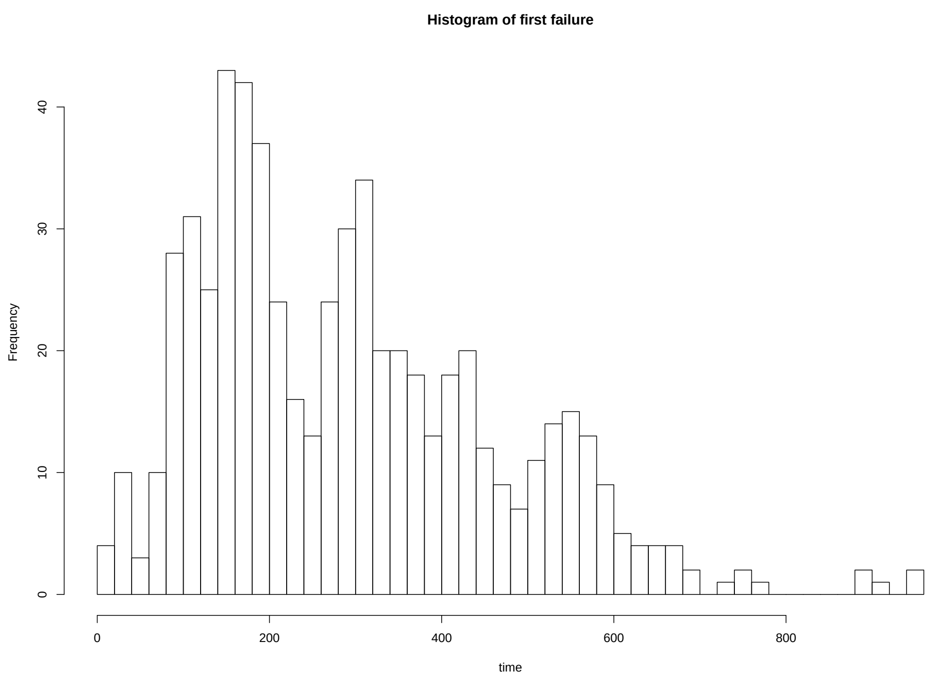
In practical situations this may pose no problem, because when arriving at an empty intersection, the traffic light will very likely switch to green immediately. Perhaps it is wise to give no predictions during the night for dynamically managed lights.

Performance measure	99.9% confidence	95%confidence
50% of average green and 100% correct	15 sec	48 sec
25% of average green and 100% correct	18 sec	81 sec

Table 6: Updating period to reach given confidence during night, intersection 4110



(a) Prediction interval is 50% of the average green time, 100% has to be correct



(b) Prediction interval is 25% of the average green time, 100% has to be correct

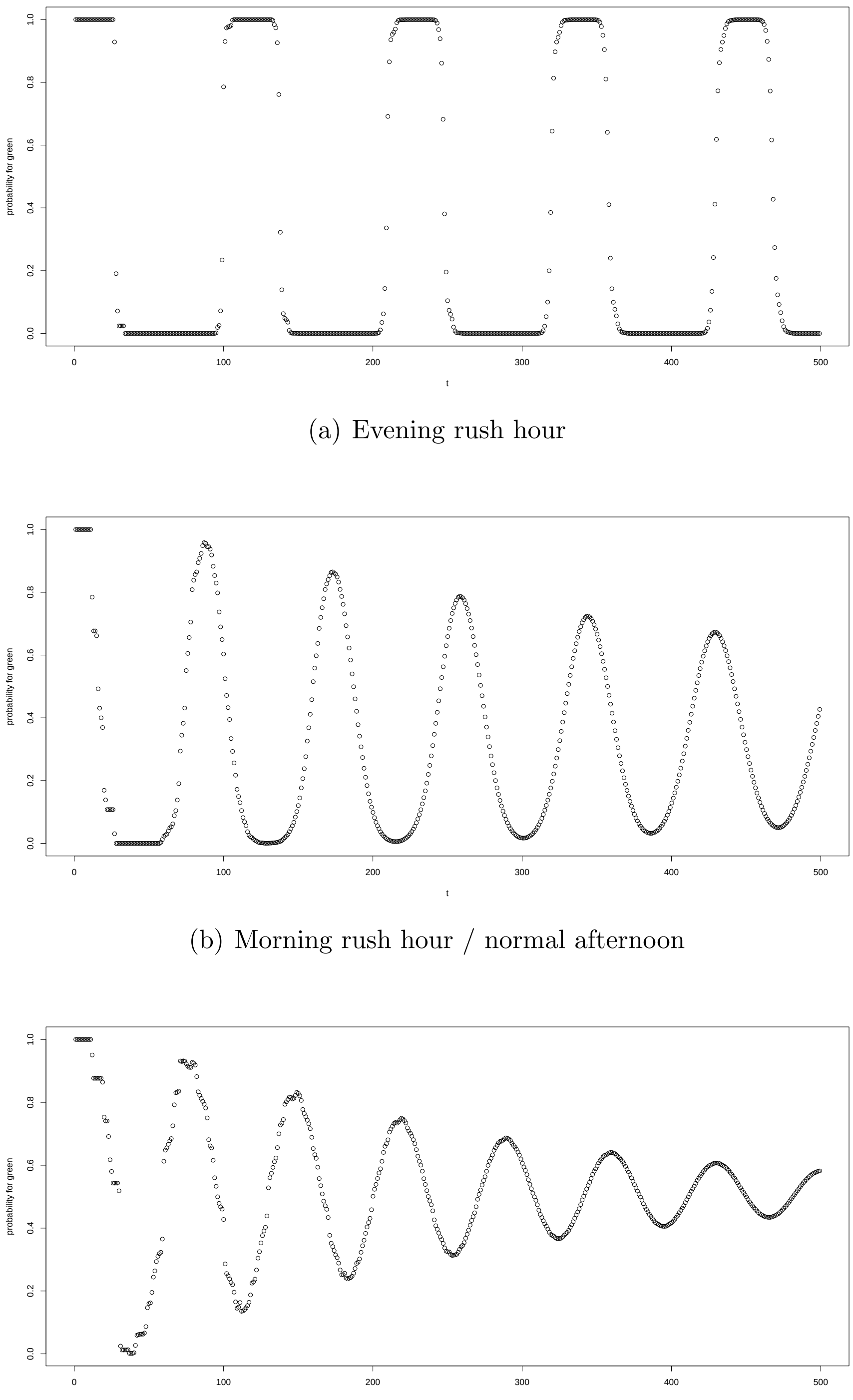
Figure 50: Histograms of first failure in the night, intersection 4110

Intersection 4113 is located in the neighborhood of intersection 4110 (see Figure 53 for the exact location of this intersection). The predictions of phase 6 (from south to north) illustrate nicely how the traffic light behaves differently during a day. The traffic light behaves almost statically during evening rush hour. At this moment the traffic intensity was sufficiently high such that the maximum specified times almost always occur. During a normal afternoon and morning rush hour, the traffic intensity was lower and the traffic light is more dynamically managed. In the night, the traffic light is very dynamic because the roads are almost empty and the traffic light immediately reacts to upcoming vehicles.

We see in Figure 52 how the predictions perform for intersection 4113. If we compare intersection 4113 with intersection 4110, intersection 4113 is better predictable during morning rush hour and less predictable in the night. But the main behavior of the performances is similar. The results of the updating periods can be seen in Table 7.

Performance measure	99.9% confidence	95%confidence
25% of average green and 100% correct, morning rush hour	66 sec	92 sec
25% of average green and 100% correct, night	16 sec	39 sec

Table 7: Updating period to reach given confidence, intersection 4113

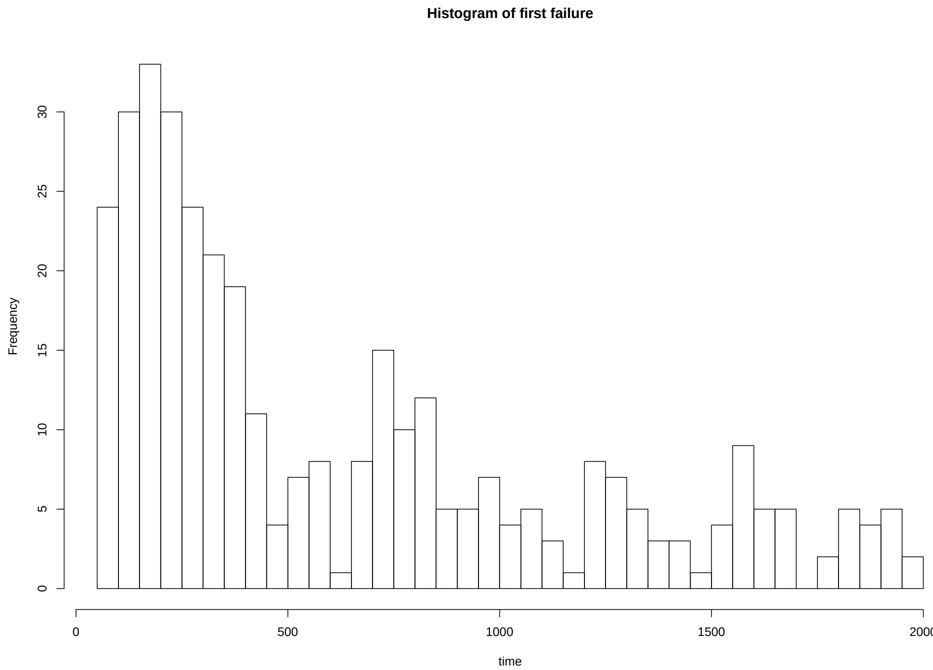


(a) Evening rush hour

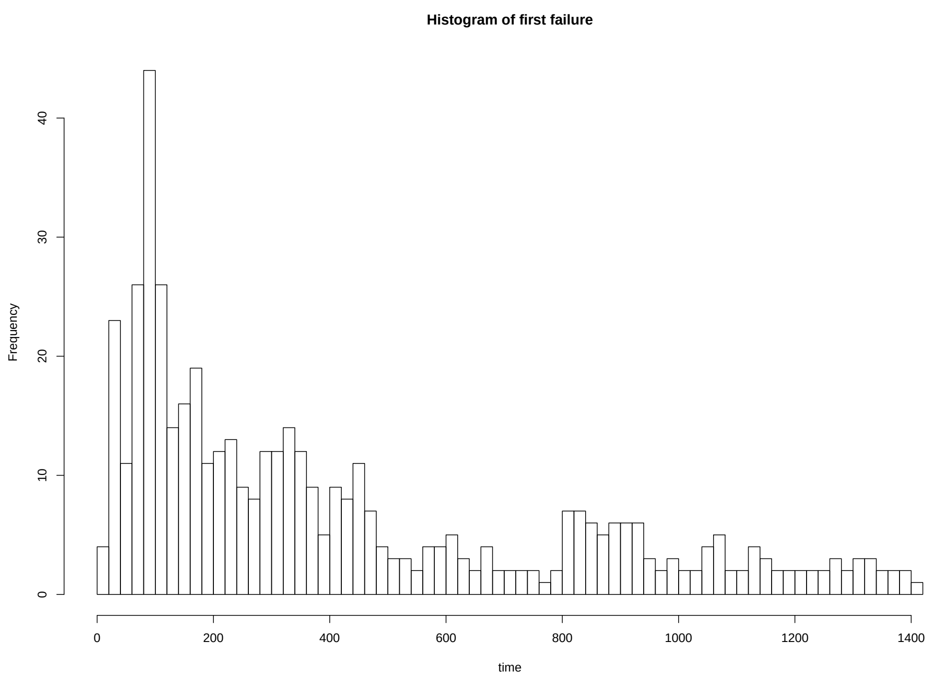
(b) Morning rush hour / normal afternoon

(c) Night

Figure 51: Predictions for intersection 4113



(a) Morning rush hour, 100% of interval has to be correct



(b) Night, 100% of interval has to be correct

Figure 52: Histograms of first failure for 25% of the average green time, intersection 4113

5.3 Conclusions

The traffic lights in Helmond can be considered as one of the most dynamically managed. For these lights it is perhaps only useful to give predictions for the first minute. The distributions of the green and red times can be different for each approach, intersection and hour of the day. So fitting distributions and corresponding parameters will not be desirable for the implementation and an empirical approach will be easier and more efficient (see Section 7 for more information about the implementation issues).

The analysis of the traffic lights in Portland is very hopeful, because compared to Helmond we see less variation in the green and red times. During morning rush hour, the predictions for dynamically managed traffic lights can be correct for about 400 seconds on average. In the night, the predictions are correct for around 100 seconds. We also observe that some intersections look statically managed during rush hour and very dynamic in the night. Using first failure analysis for the predictions, we can provide update periods to achieve a desired confidence for the predictions.

During the night the unpredictability of the traffic lights may not be a problem, because when arriving at a signalized intersection the light will likely turn green immediately.

The predictions in Portland are performing very well and they seem to be usable to improve travel time estimations in networks with multiple signalized intersections.

6 Network of dynamically managed traffic lights

As indicated in Section 5, we have received data for about 850 signalized intersections in Portland. This data is very useful to test our travel time predictions for a network containing dynamically managed traffic lights. First we explain how the travel time estimations for a route can be calculated based on our traffic light predictions. We conclude this section with an example of a network in Portland with various types of traffic lights, where we will try to find faster routes.

6.1 Travel time predictions

In Figure 53 we see a small route in Portland which contains three successive dynamically managed traffic lights. Our goal is to give better travel time predictions for this route using traffic light predictions. At the moment, only *average* delays at traffic lights are used in car navigation. We want to predict the delay at an intersection, which strongly depends on current traffic light states. The travel times between intersections can be estimated by the TomTom device. This idea improves the existing route planning and can lead to faster routes and green wave advices.

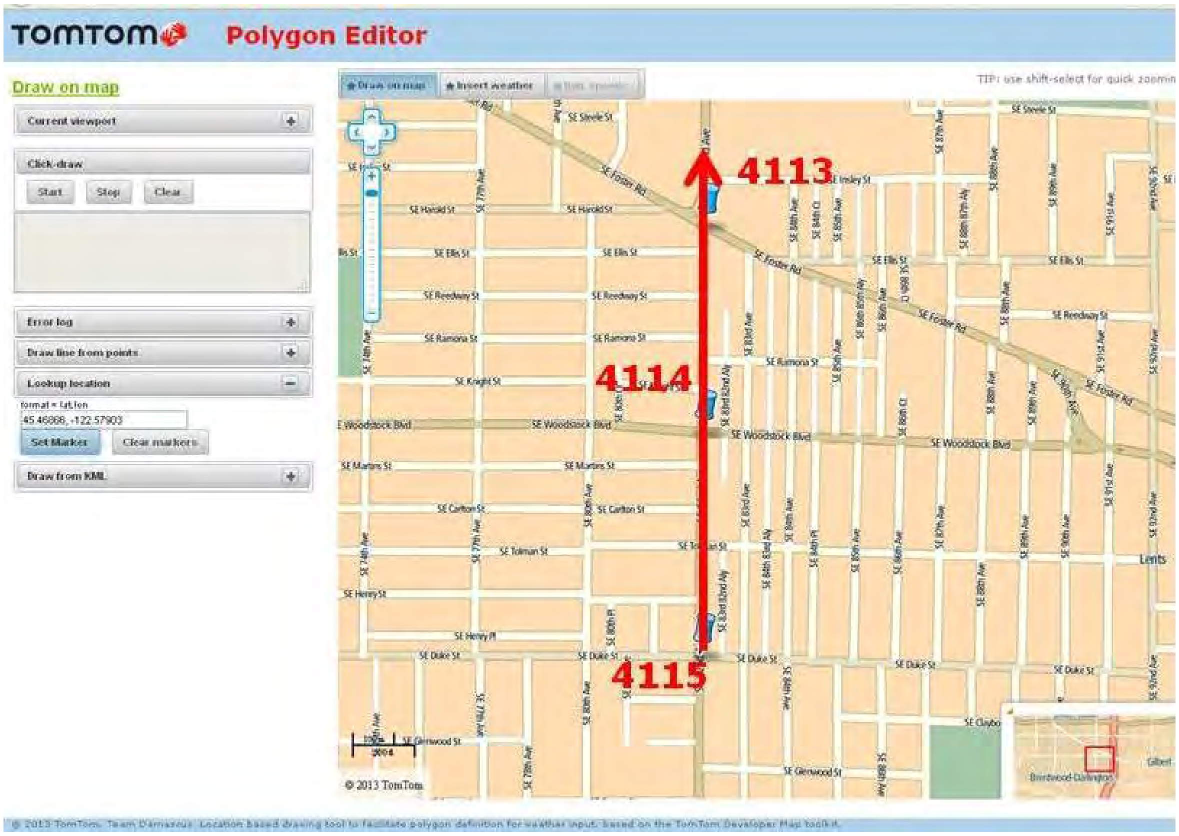


Figure 53: Route with three dynamically managed traffic lights

The model for the route is the same as explained in Section 3. It takes T_1 seconds to drive to intersection 1, T_2 seconds to get from intersection 1 to intersection 2, etc. At each intersection the car has to wait some time until the next green time. The expected waiting time ($\mathbb{E}[W_i(t)]$) of intersection i can be calculated by Equation (11). If a route has I intersections with traffic lights, it will have $I + 1$ sub routes as illustrated in Figure 54:

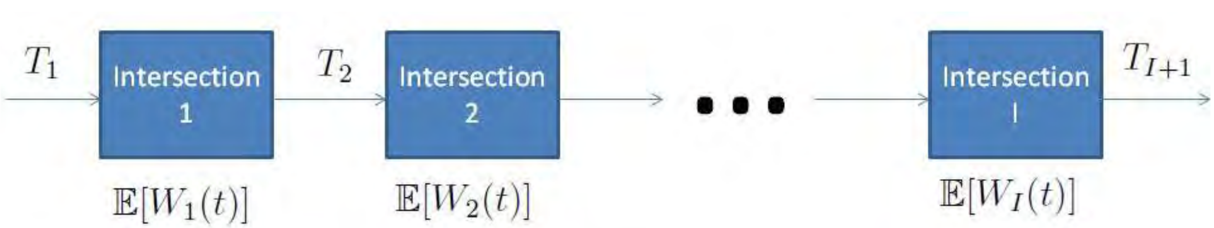


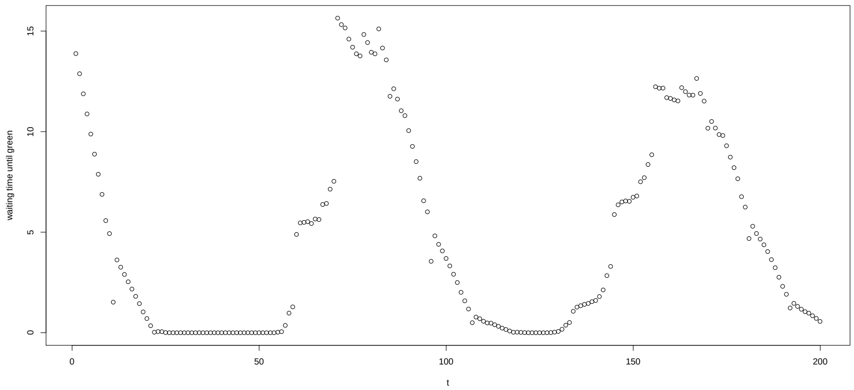
Figure 54: Abstract view of a route

Define $\mathbb{E}[S(i)]$ as the expected travel time after i sub routes, just after T_i (for $i = 1, 2, ..., I+1$). The expected total travel time from Figure 54 (also called the expected sojourn time) can now be computed with the following recursion:

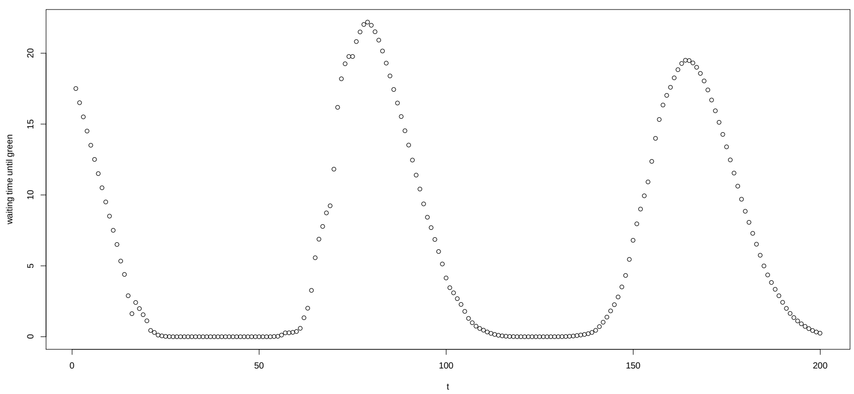
$$
\begin{aligned}
\mathbb{E}[S(1)] &= T_1, \\
\mathbb{E}[S(2)] &= \mathbb{E}[S(1)] + \mathbb{E}[W_1(\mathbb{E}[S(1)])] + T_2, \\
\mathbb{E}[S(3)] &= \mathbb{E}[S(2)] + \mathbb{E}[W_2(\mathbb{E}[S(2)])] + T_3, \\
&\vdots \\
\mathbb{E}[S(I+1)] &= \mathbb{E}[S(I)] + \mathbb{E}[W_I(\mathbb{E}[S(I)])] + T_{I+1}.
\end{aligned}
$$

The expected sojourn time of the route is $\mathbb{E}[S(I+1)]$. If at least one of the traffic lights is dynamically managed, the sojourn times will be stochastic. Like in Section 3, we can allow variation in the travel times between intersections. We will introduce a heuristic to approximate the expected sojourn time of the route. If we assume that $T_i \sim \mathrm{U}(a_i, b_i)$, then we calculate the expected travel times of the route for $T_i = a_i, T_i = \frac{a_i+b_i}{2}$ and $T_i = b_i$. If the route has I intersections, the approximation of the expected sojourn time is the average of all 3^{I+1} combinations. If the travel times between intersections are distributed differently, we can add more weight to the travel times that are more likely. Research on actual travel time data is needed to derive how the times are distributed, which strongly depends on traffic intensity. The idea of allowing variation in the travel times between intersections, is to recognize the situation where it becomes uncertain whether the car will catch the green light.

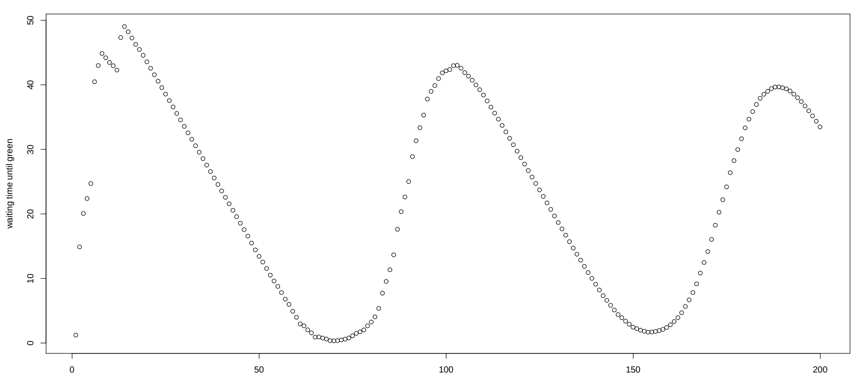
In the example of Figure 53, we want to calculate the expected sojourn time of the network with three signalized intersections (the moment we arrive at intersection 4115 till we leave intersection 4113). So we calculate $\mathbb{E}[S(4)] - \mathbb{E}[S(1)]$, where $T_4 = 0$ and $T_2 \sim T_3 \sim \mathrm{U}(21, 25)$. Now we want to calculate the expected sojourn time for each arrival time at the first intersection. Therefore, we need the expected waiting time until next green light functions for the three intersections. These functions are plotted in Figure 55. The plots indicate the expected waiting time until next green light for $t = 1, 2, ..., 200$, which are based on lists of green and red times.



(a) Waiting time until green, intersection 4115



(b) Waiting time until green, intersection 4114



(c) Waiting time until green, intersection 4113

Figure 55: Waiting time until next green light

Let for example the arrival time at intersection 4115 be 50 seconds and $T_2 = T_3 = 23$. The expected time when the vehicle leaves intersection 4113 will be calculated by:

$$
\begin{aligned}
\mathbb{E}[S(1)] &= 50, \\
\mathbb{E}[S(2)] &= 50 + \mathbb{E}[W_1(50)] + 23 = 73, \\
\mathbb{E}[S(3)] &= 73 + \mathbb{E}[W_2(73)] + 23 = 109, \\
\mathbb{E}[S(4)] &= 109 + \mathbb{E}[W_3(109)] + 0 = 150.
\end{aligned}
$$

Thus the expected sojourn time of the network is: $\mathbb{E}[S(4)] - \mathbb{E}[S(1)] = 150 - 50 = 100$, which is very high because the vehicle has to wait long at the last intersection.

For the second example, let the arrival time at intersection 4115 be 25 seconds, $T_2 = 21$ and $T_3 = 25$. The expected time when the vehicle leaves intersection 4113 will be calculated by:

$$
\begin{aligned}
\mathbb{E}[S(1)] &= 25, \\
\mathbb{E}[S(2)] &= 25 + \mathbb{E}[W_1(25)] + 21 = 46, \\
\mathbb{E}[S(3)] &= 46 + \mathbb{E}[W_2(46)] + 25 = 71, \\
\mathbb{E}[S(4)] &= 71 + \mathbb{E}[W_3(71)] + 0 = 71.
\end{aligned}
$$

Hence, the expected sojourn time of the network is: $\mathbb{E}[S(4)] - \mathbb{E}[S(1)] = 71 - 25 = 46$. In this case, the car does not have to wait at any intersection. This indicates that the driver will get a green wave. If a green wave is not possible, we can also try to minimize the amount of red lights during a journey. If one of the expected waiting times is sufficiently small, our traffic light phase predictions can still guide the driver through a green wave without giving too small speed advices (see Section 7 for more details about the speed advices). In our research, the goal is to minimize the travel time.

Doing the expected sojourn time calculations for all combinations and all arrival times at the first intersection, results in Figure 56. We see that the expected sojourn time fluctuates between 46 and 100 seconds, which makes a huge difference. At the local minima, the driver will likely have a green wave.

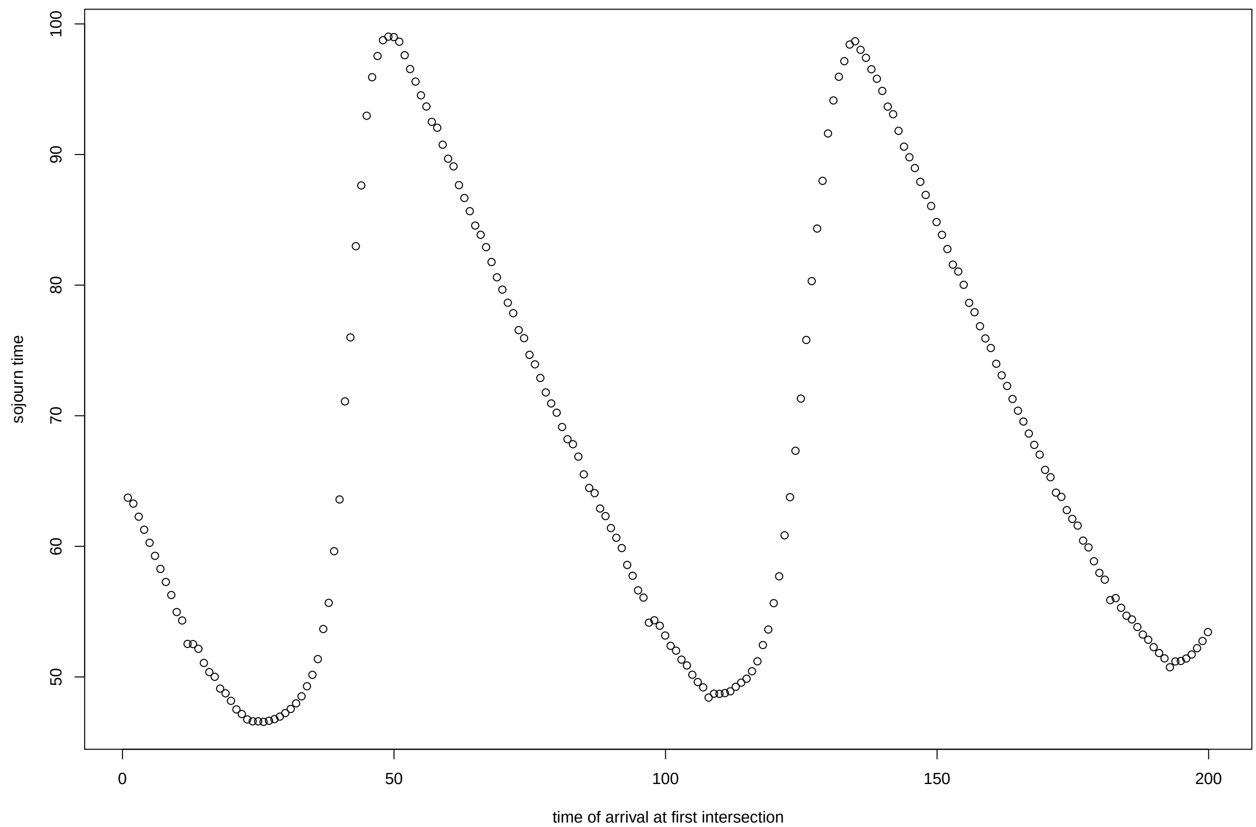


Figure 56: Sojourn time

To obtain a feeling how the expected sojourn times perform, we can compare the predictions with the real traffic light states like in Figure 57. On the left side of the figure the real green and red times are visualized, included with maximum speed lines to determine how fast a vehicle can drive through the network. On the right side of the figure some predictions are visualized by arrows, with the time when the vehicle arrives at intersection 4115 and the predicted time when the vehicle leaves the network at intersection 4113.

Our travel time estimations predict that if we arrive at time 1, we leave the network at 64. The real green and red time windows tell us that if we arrive at time 1, we first have to wait 11 seconds until we can leave intersection 4115. With the maximum speed, we catch green light at intersection 4114 and arrive during a red light at intersection 4113. We can leave intersection 4113 around time 61, which is close to our prediction. The predictions say that if we arrive around time 32, we will have a green wave and the real time window confirms this. The real time window shows that if the car arrives at time 48, it will just catch the green lights at the next intersections if the maximum speed can be driven. If the speed is slightly lower, the driver will miss a green light and this results in a delay of 54 seconds. The travel time predictions recognize that this situation is risky and the probability that we miss a green light results in a high estimated sojourn time. For car navigation this behavior is desirable, because we do not want to advise high risk routes.

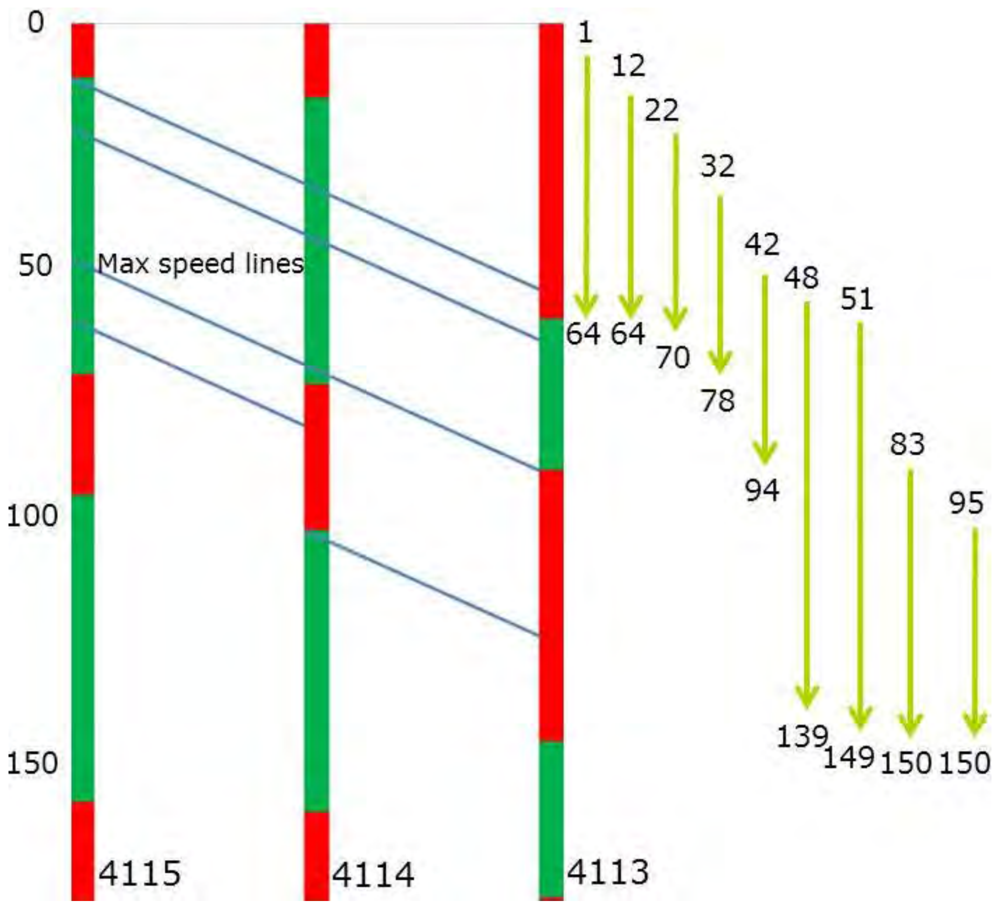


Figure 57: Comparing the expected travel times with real traffic light states

We can also analyze the limiting behavior of the expected sojourn time. Because all three traffic lights are dynamically managed, the expected sojourn time will converge to a fixed value. This limit should be the *average* travel time of the route (which is currently used in car navigation). By using the limit Equation (13), we can calculate the theoretical limit of the expected sojourn time:

$$
\begin{aligned}
\lim_{t\to+\infty} \mathbb{E}[W_1(t)] &= \frac{\mathbb{E}[R_j^2]}{2(\mathbb{E}[R_j]+\mathbb{E}[G_j])} = 5.10, \\
\lim_{t\to+\infty} \mathbb{E}[W_2(t)] &= 6.25, \\
\lim_{t\to+\infty} \mathbb{E}[W_3(t)] &= 19.70, \\
\mathbb{E}[T_2] &= 23, \\
\mathbb{E}[T_3] &= 23, \\
\mathbb{E}[S] &= 5.10 + 23 + 6.25 + 23 + 19.70 = 77.05.
\end{aligned}
$$

If we compare the theoretical value of the limit with Figure 58, it corresponds.

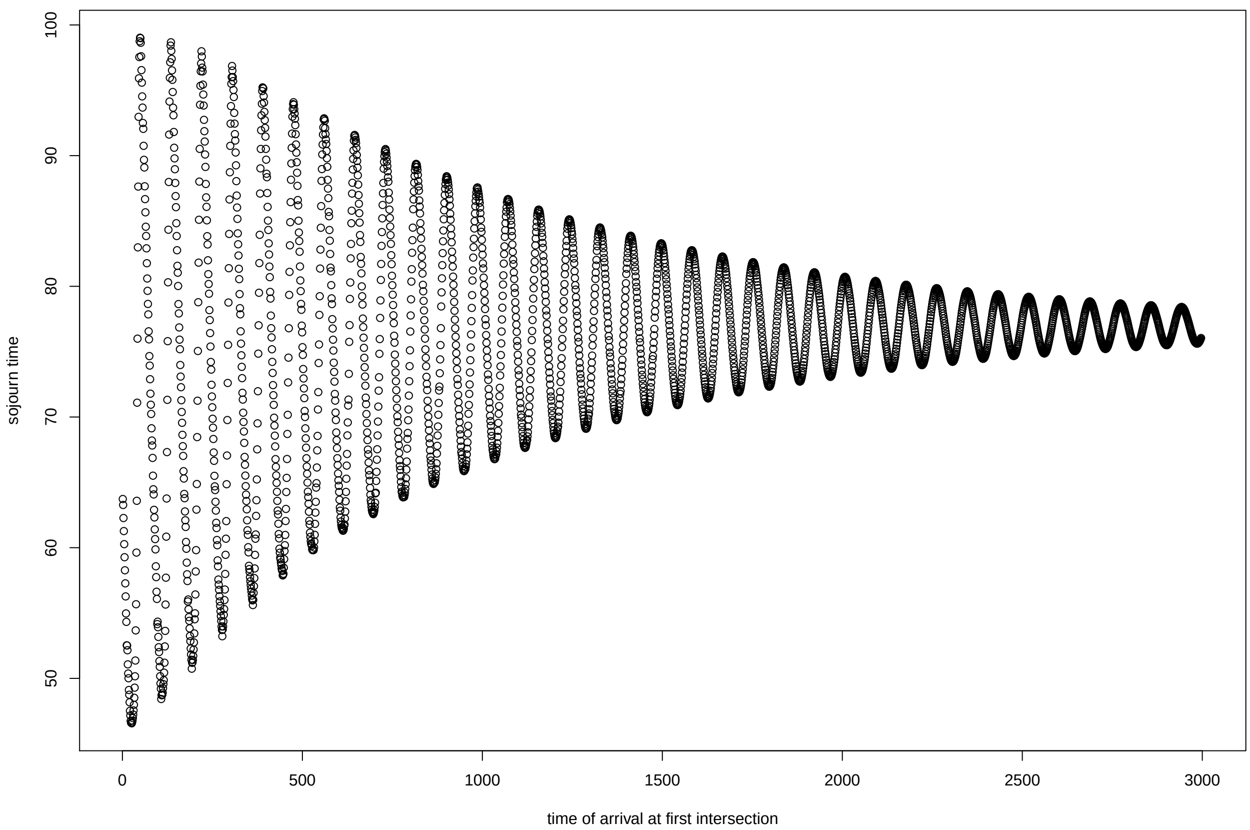


Figure 58: Limit of sojourn time

6.2 Example of a network in Portland

If we have a network of signalized intersections with different routes, we can apply our improved travel time predictions to find faster routes. For this purpose we use a network in the center of Portland. This network contains both statically and dynamically managed traffic lights. Also a tram passes the network, causing large variations for green times of some intersections. The network is illustrated in Figure 59. In Table 10, we give an indication how the traffic lights are behaving. We see that the traffic lights at the bottom of the network are very dynamic, due to the passing tram. The upper left route is on average the fastest and will always be advised by the TomTom. For this route the traffic lights are also better specified to give green waves. But if we know the profiles and current states of all traffic lights, can we find a faster route? The possible routes are given in Tables 8 and 9. Note that route 6 is the upper left. The travel times between the intersections are estimated from an online route planner and the combination of distances and speed limits. For the actual TomTom implementation more research is needed how to derive the required distributions, but for our purpose we make assumptions to illustrate how the model works. We distinguish two cases for the distribution of the travel time between two intersections:

$$T_i \sim \begin{cases} \mathrm{U}(\mathbb{E}[T_i]-1, \mathbb{E}[T_i]+1) & \text{if } \mathbb{E}[T_i] \leq 30, \\ \mathrm{U}(\mathbb{E}[T_i]-3, \mathbb{E}[T_i]+3) & \text{if } \mathbb{E}[T_i] > 30. \end{cases}$$

For all routes described in Tables 8 and 9, we calculate the expected sojourn time for each arrival time at intersection 2028. These results are plotted in Appendix B.3. In the plots we can conclude that for most routes one traffic light is dominating the behavior of the expected travel time. The traffic light with the largest red time is the dominant factor. For route 6 we see that two traffic lights are mainly influencing the expected sojourn time. If we take the limit of the sojourn time, only statically managed traffic lights on the route will dominate the behavior. The results show that in 16.7% of the cases, another route is faster. The distribution of the saved time for the faster route, is plotted in Figure 71. The average time we save when taking a faster route is 14.5 seconds.

By following these faster route advices, we will probably also improve the traffic flow in the entire network. In the old scenario, all vehicles are led over route 6. But including our advices, 16.7% will be guided over a different route. So the predictions will also divide the traffic better over the network and not only the individual driver is faster, but the entire network is improved.

Figure 59: Network in Portland

	route 1	route 2	route 3	route 4	route 5	route 6
intersection, phase	2028, 4	2028, 4	2028, 4	2028, 4	2028, 4	2028, 4
$\mathbb{E}[T_1]$	8	8	8	12	12	12
intersection, phase	2161, 6	2161, 6	2161, 6	2029, 2	2029, 2	2029, 2
$\mathbb{E}[T_2]$	8	8	8	16	16	12
intersection, phase	2037, 2	2037, 2	2037, 2	2044, 2	2044, 5	2150, 2
$\mathbb{E}[T_3]$	17	17	17	15	40	7
intersection, phase	2038, 1	2044, 4	2044, 4	2045, 5	2092, 8	2224, 2
$\mathbb{E}[T_4]$	17	15	40	20	14	7
intersection, phase	2045, 8	2045, 5	2092, 8	2127, 4	2097, 2	2030, 2
$\mathbb{E}[T_5]$	20	20	14	7		15
intersection, phase	2127, 4	2127, 4	2097, 2	2097, 4		2092, 2
$\mathbb{E}[T_6]$	7	7				14
intersection, phase	2097, 4	2097, 4				2097, 2
average travel time	152	129	145	113	175	97

Table 8: Possible routes through network in Portland

	route 21	route 61	route 62	route 63
intersection, phase	2028, 4	2028, 4	2028, 4	2028, 4
$\mathbb{E}[T_1]$	8	12	12	12
intersection, phase	2161, 6	2029, 2	2029, 2	2029, 2
$\mathbb{E}[T_2]$	8	60	12	60
intersection, phase	2037, 2	2092, 8	2150, 2	2127, 2
$\mathbb{E}[T_3]$	17	13	7	7
intersection, phase	2044, 4	2097, 2	2224, 2	2097, 4
$\mathbb{E}[T_4]$	35		35	
intersection, phase	2127, 2		2127, 2	
$\mathbb{E}[T_5]$	7		7	
intersection, phase	2097, 4		2097, 4	
average travel time	159	127	114	127

Table 9: Some alternative routes through network in Portland

behavior	intersections
statically managed	2150, 2224, 2030, 2097, 2127
bit dynamically managed	2029, 2092, 2045, 2044
very dynamically managed	2028, 2161, 2037, 2038

Table 10: Behavior of the traffic lights

6.3 Conclusions

Using traffic light phase predictions, we are able to improve the travel time estimations for a route containing multiple dynamically managed traffic lights. We observe that the expected travel times behave harmonic and the traffic light with the largest red time is most dominant. In the center of Portland, there is a network with various situations. The network has both dynamically and statically managed traffic lights. The traffic light phase predictions show that for the upper left route, a green wave has been implemented and this route has the lowest average travel time. Also the TomTom devices will always advise this route. **An implementation for nine possible routes in this network, shows that we can find faster routes in 16.7% of the cases (with 14.5 seconds on average)**.

7 TomTom implementation

In this section we will explain how the ideas can be implemented and visualized in the TomTom device and which issues are involved. The End-to-end design is illustrated in Figure 60. We use traffic light data in our prediction model. The outcome of the prediction model can be used to give better route advices and speed advices to catch green light. In this section we will explain the design and discuss the corresponding issues.

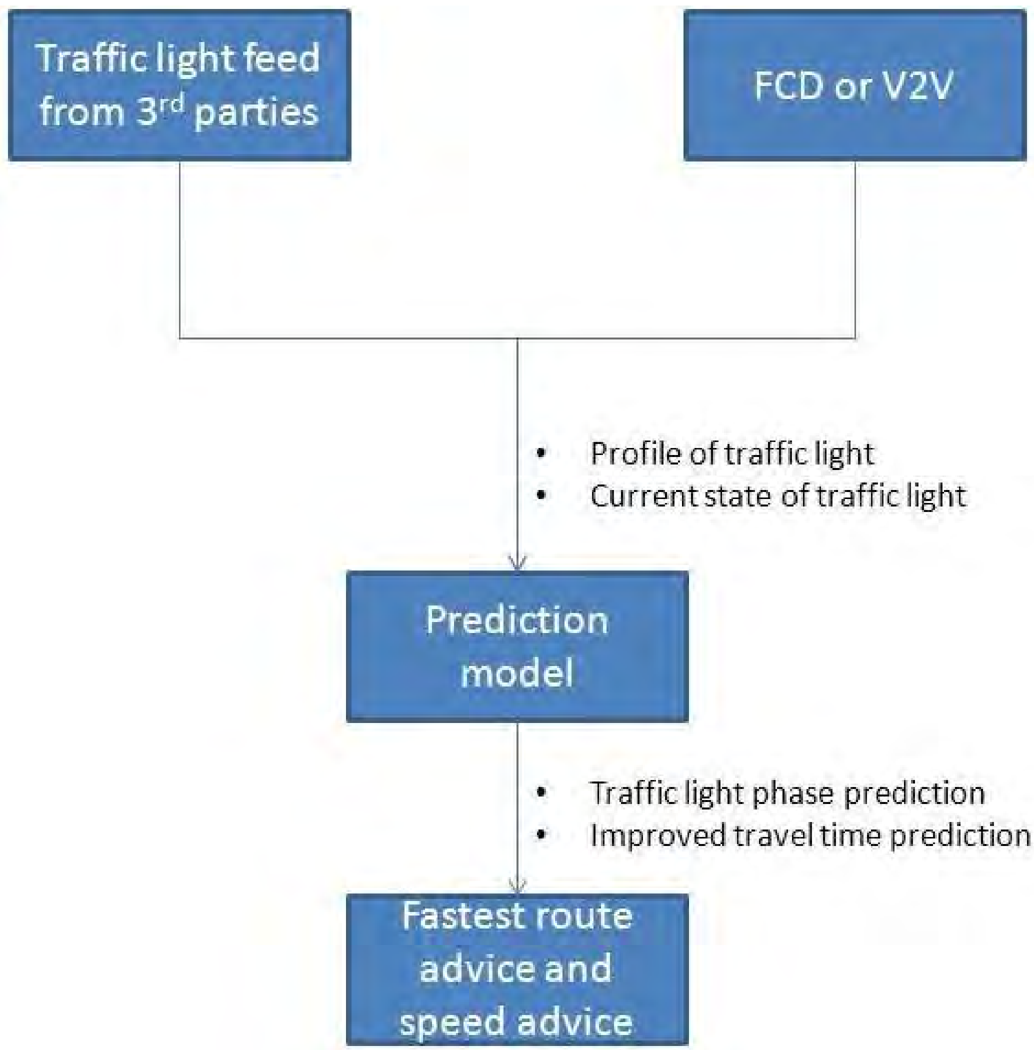


Figure 60: End-to-end design

7.1 Issues with the traffic light phase predictions

As input for the prediction model we need the profile of the traffic light and a current state. The profile of the traffic light describes how the traffic light is behaving, in our model a list of green and red times is sufficient (as used in Section 5.2.3). The current state is the amount of time the light is green or red. In Section 2, we modeled the current state as a and r.

The lists of green and red times can be collected from historical analysis. In this application, we should store lists for the following time slots: night, morning rush hour, normal morning/afternoon, evening rush hour and normal evening. Tentatively, the lists are stored for each possible cycle plan. In the last case before each prediction, we first have to determine which cycle plan is currently active (if more are possible). The database of the profiles can also be more dynamic. So for example, we could choose to store the lists of green and red times of the last 30 minutes and update the list every 5 minutes. The last option is likely better for the performance of the predictions, because it reacts to current developments of the traffic lights. But it will use more CPU power of the server/device. It is still desirable to analyze how large the lists of green and red times should be to give good predictions. If the list is too small, perhaps the behavior is not described correctly. If we increase the sizes of the lists, then we have to store/send more data.

If the traffic intensity is low for a dynamically managed traffic light, it is possible that a signal group will be skipped during the cycle. So in our list of red times, we do not want to include red times which are significantly larger than the cycle length. When we do add the large red times, it will lead to a wrong cycle length prediction. Also when the car arrives at the intersection, the corresponding signal group will always receive green within the next cycle. When we remove red times that are larger than the cycle length, the predictions will converge to the correct limit. Since in this situation the traffic light behaves very dynamic, the predictions will converge fast.

The traffic light data can be provided by third parties and/or FCD/V2V. Floating Car Data and Vehicle to Vehicle information is continuously gathered by live communicating TomTom devices. Often it is not possible or desirable to use data from third parties (like Green Driver or PeekTraffic), so TomTom may want to acquire the data with its own resources. We are able to determine the required traffic light data using TomTom traces for statically managed traffic lights (see Section 4). For both traffic light feeds and TomTom traces, the frequency of the traffic light updates is essential to guarantee good predictions.

To increase the calculation speed, we make use of discretization. We discretize the green times, red times and time line in seconds. We use the profile of the traffic light to calculate the empirical distributions of the green and red times. Let $x_1, x_2, ..., x_n$ be the green times of the traffic light and $y_1, y_2, ..., y_m$ the red times (which are all positive integers). Define $G(t)$ as the empirical distribution function of the green time and $R(t)$ of the red time. The empirical distribution functions can be calculated by:

$$G(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \leq t\}, \text{ for } t = 0, 1, ...$$
$$R(t) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{y_i \leq t\}, \text{ for } t = 0, 1, ...$$

Convert the empirical distributions of the green and red times to probability mass functions. The probability mass functions can be derived by:

$$g(t) = \begin{cases} G(t) - G(t-1) & \text{for } t = 1, 2, ... \\ 0 & \text{for } t = 0, -1, -2, ... \end{cases}$$

$$r(t) = \begin{cases} R(t) - R(t-1) & \text{for } t = 1, 2, ... \\ 0 & \text{for } t = 0, -1, -2, ... \end{cases}$$

Using convolution, we can calculate the probability mass function of the sum of two random variables. The probability mass function of the cycle can be calculated by:

$$c(t) = (g * r)(t) = \sum_{i=-\infty}^{\infty} g(i)r(t-i), \text{ for } t = 1, 2, ... \tag{14}$$

Then calculate all probability mass functions of necessary cycle combinations by repeatedly using convolutions. These cycle combinations are given by the sums of random variables in Section 2. Convert these probability mass functions back to distribution functions to calculate the probability of green light for every moment in the future. Let $m(t)$ be a probability mass function, the corresponding distribution function $M(t)$ can be derived by:

$$M(t) = \sum_{i=1}^{t} m(i), \text{ for } t = 1, 2, ..$$

Also use the convolutions to calculate conditional expectations, which are used for the expected waiting time until next green light. For the predictions, also continuous variables can be used. This continuous case is described by the equations in Section 2 and an example is presented for the normal distribution. For the discrete case, the integral in Equation (10) becomes a sum and the probability density function is a probability mass function (like used above).

The bounds of the sums can be replaced by using the minimum and maximum green/red times of the given lists. Let $g_{\min} = \min\limits_{i=1,\dots,n}(x_i)$ and $r_{\min} = \min\limits_{i=1,\dots,m}(y_i)$. Define the maximum times as $g_{\max} = \max\limits_{i=1,\dots,n}(x_i)$ and $r_{\max} = \max\limits_{i=1,\dots,m}(y_i)$. Also use the fact that the green and red times are always positive. So we can bound Equation (14) by:

$$c(t) = \begin{cases} (g * r)(t) = \sum\limits_{i=0}^{t} g(i)r(t-i) & \text{for } g_{\min} + r_{\min} \leq t \leq g_{\max} + r_{\max}, \\ 0 & \text{for } t < g_{\min} + r_{\min} \text{ or } t > g_{\max} + r_{\max}. \end{cases}$$

7.1.1 Special case: normal distribution

In this thesis, we have used normal distribution fits for the traffic light phase predictions. In case of the normal distribution, the possible values of the green and red times are unbounded. So like above, the minimum and maximum green and red times cannot be used. As in Section 2, we start with the case where the green times are normally distributed and the red times are deterministic.

In theory we must take all starting green times, thus we sum from 1 to ∞. But most green times are unlikely to be the next starting green time. So we must choose the boundaries of the summation wisely, or else a computer will have problems calculating the small numbers (in some equations we will possibly divide by values which are nearly zero). When the green times are normally distributed, we can calculate for each G_{is} a confidence interval in which this G_{is} is likely to be the next starting green time. If we want to make predictions for given time t, then take all $i \in \mathbb{N}$ such that:

$$t \quad \in \quad \left(\mathbb{E}[G_{is}] - R - d\sqrt{i\sigma_G^2}, \mathbb{E}[G_{is}] + d\sqrt{i\sigma_G^2}\right).$$

This can be rewritten as:

$$t \quad \in \quad \left((i-1)\mu_G + (i-2+r)R - a - d\sqrt{i\sigma_G^2}, (i-1)\mu_G + (i-1+r)R - a + d\sqrt{i\sigma_G^2}\right),$$

where d is the number of standard deviations away from the expected value. So if $d = 2$ we approximately have a 95% confidence interval and for $d = 3$ the confidence interval is around 99.7%.

When both green and red times are normally distributed, we can calculate similarly for each G_{is} a confidence interval in which this G_{is} is likely to be the next starting green time. If we want to make predictions for given time t, then take all $i \in \mathbb{N}$ such that:

$$t \in \left(\mathbb{E}[G_{is}] - \mathbb{E}[R_2] - d\sqrt{i(\sigma_G^2 + \sigma_R^2)}, \mathbb{E}[G_{is}] + d\sqrt{i(\sigma_G^2 + \sigma_R^2)}\right).$$

This can be rewritten as:

$$t \in \left((i-1)\mu_G + (i-2+r)\mu_R - a - d\sqrt{i(\sigma_G^2 + \sigma_R^2)}, (i-1)\mu_G + (i-1+r)\mu_R - a + d\sqrt{i(\sigma_G^2 + \sigma_R^2)}\right).$$

7.2 Visualizing the predictions

First we visualize the probabilities by a lemon green, pink red and brown time window. This is illustrated in Figure 61. The lemon green intervals indicate that the probability of green light is high, so we try to lead the driver to this region to have maximum probability to catch green light. The pink red area indicates that the probability for red light is high and in the brown intervals the predictions are insecure. The middles of the lemon green intervals lie at the local maxima of the probability plot. The middles of the pink red intervals lie at the local minima. The sizes of the lemon green and pink red intervals can be changed, which influences the performance of the predictions. The issues around the visualization are described in Section 5.2.3.

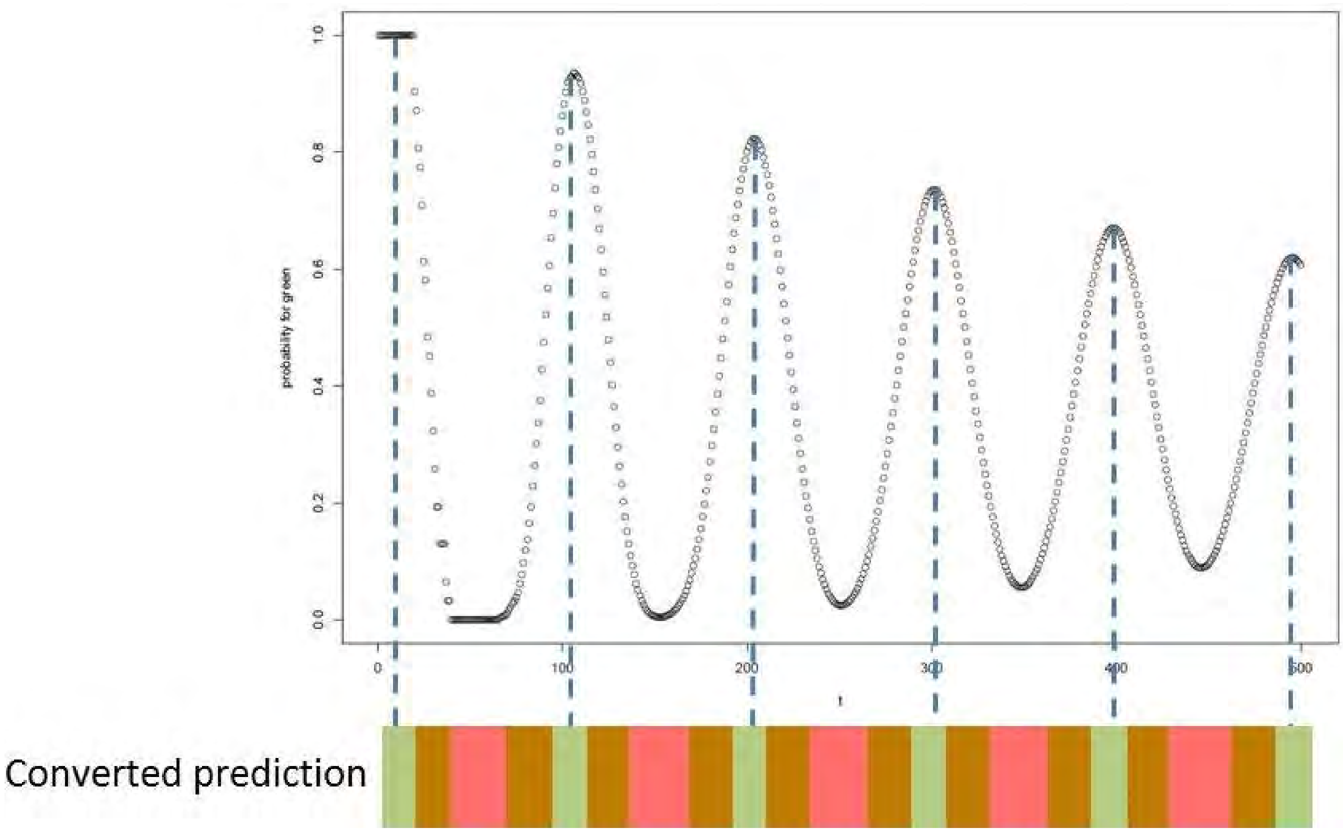


Figure 61: Visualizing the prediction

When approaching a traffic light, the speed advices should be visualized in the car. By using the distance from the vehicle to the traffic light, we convert the time window to a speed advice window. First we calculate at which area in the time prediction window we arrive, if the vehicle can drive with maximum speed. Therefore TomTom has to know the speed limit on the road toward the intersection. Then we take all times where the color changes in the prediction window. Calculate for these times what the corresponding speed should be to arrive exactly at this moment. Use the calculated speeds to visualize the speed advice window. Note that if the time prediction window is infinite, the speed advice intervals convert to zero. It is clearly not desirable to give too small speed advices (like 1 miles/hour), which can lead to irritations toward other road users. Therefore a minimum speed should be implemented and only speed advices between the boundaries shall be visualized like in Figure 62. If desired, we can also tell the driver that it is not possible to catch green light.

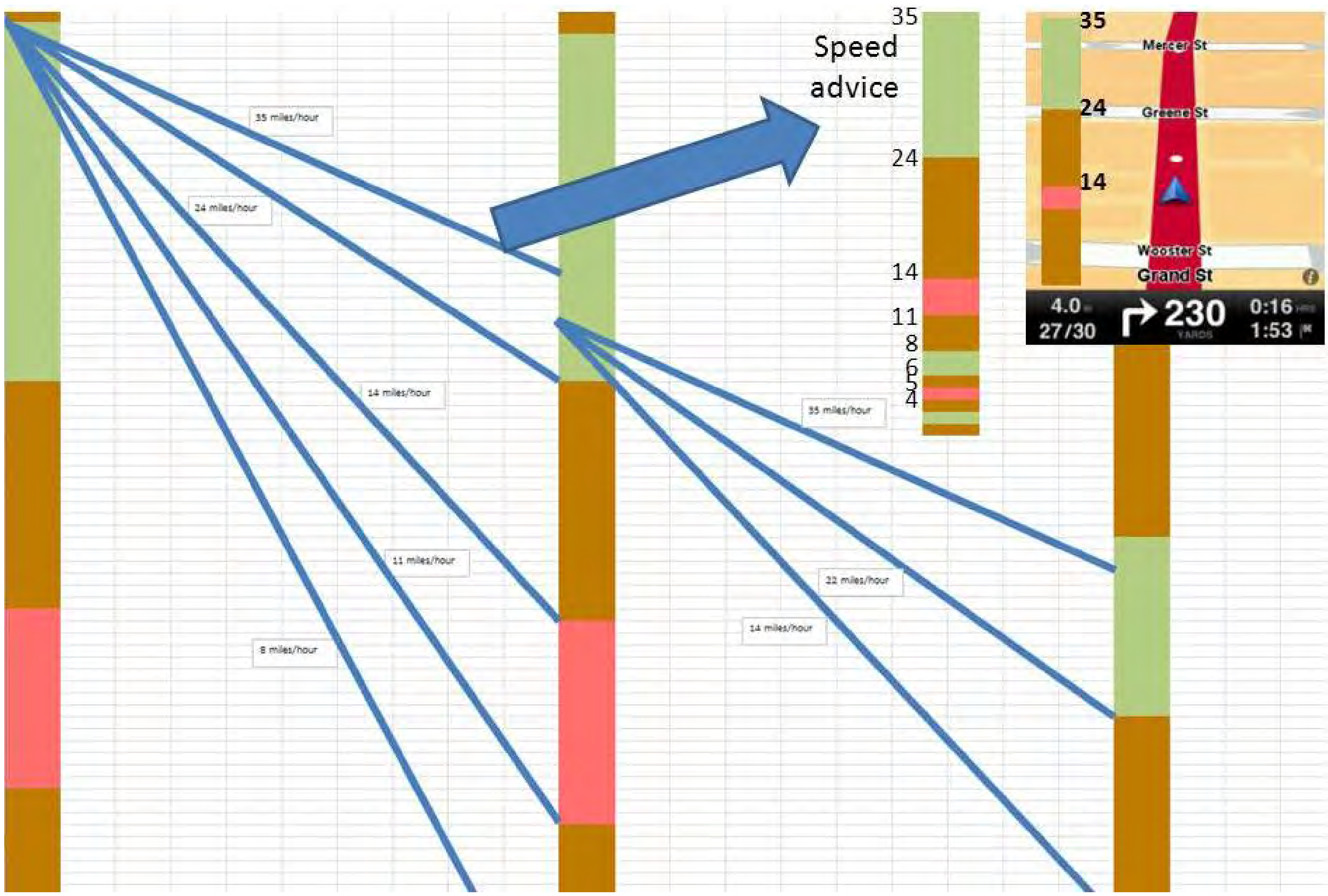


Figure 62: Visualizing speed

7.3 Calculating the expected travel time

We use the expected waiting time until next green functions to predict the travel time for a route containing multiple traffic lights. Combine these expected waiting times with the expected travel times between intersections to give better actual travel time predictions. See Section 6 for more details about the calculation of the expected travel time. The expected travel times between intersections are stated by $T_1, T_2, ...$ and they can be random variables. These travel times can be derived from TomTom's IQ routes and/or HD Traffic. Figure 63 is an example for a route containing three traffic lights. If a car is located before a network of signalized intersections, all possible routes should be calculated and the fastest will be advised to the driver.

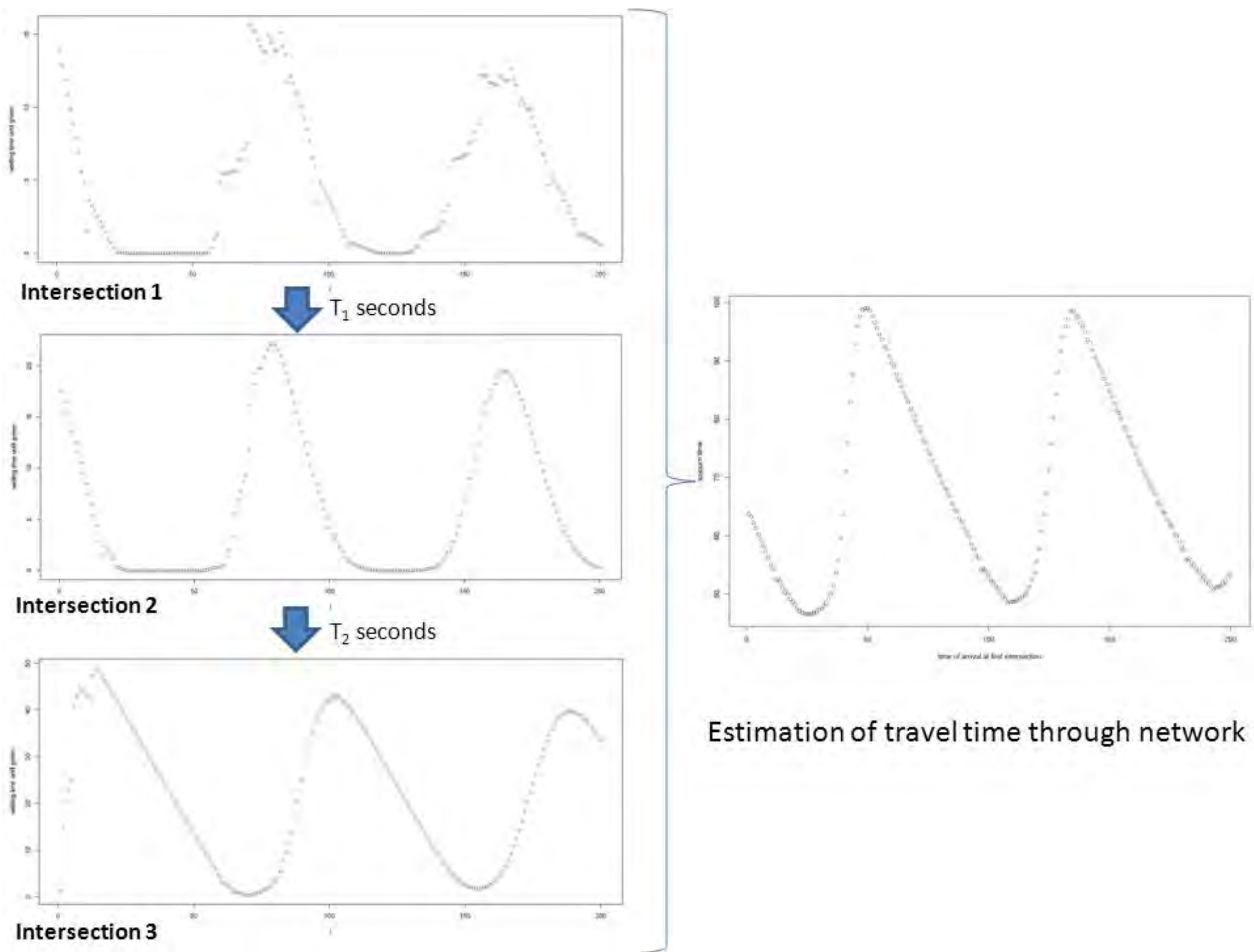


Figure 63: Calculating the expected travel time

7.4 Decisions about the implementation design

Important issues for the TomTom implementation are which calculations have to been done on the device (inside the car) and which in the BO (Back Office, the TomTom server). These decisions determine what data have to be sent between device and BO. Minimizing the amount of transmitted data is more important, than minimizing the CPU and memory usage of the device. Most likely all the traffic light data from third parties and FCD/V2V are firstly sent to the BO. So the BO must send the necessary traffic light data toward the device.

Concerning these three items one has to decide where they should be stored/calculated:

- Database traffic lights: coordinates of traffic lights, possible transitions, profile of traffic lights (the profile data can be static or dynamic)
- Calculation of traffic light phase prediction: only prediction for next approaching traffic light needed
- Calculation of route: predictions needed for multiple traffic lights which are further into the future, this makes these calculations far more heavy than phase predictions

In the following table some possible options are given, with estimated performance measures. A "+" indicates that the decisions are favorable for the performance and a "-" will be bad for the performance.

	Option 1	**Option 2**	**Option 3**	**Option 4**
Database traffic lights	Device	BO	BO	BO
Calculation phase predictions	Device	BO	Device	Device
Calculation route	Device	BO	BO	Device
Needed data transmission	Updates traffic light	Fastest route, phase predictions, updates traffic lights	Fastest route, profiles, updates traffic light	Profiles, updates traffic light
Performance measure				
Mobile data	++	- -	+	+
CPU device	- -	++	+	-
Memory device	- -	++	+	+
CPU BO	++	- -	-	+
Memory BO	++	-	-	-

Table 11: Possible options for implementation

We consider option 3 as best solution. If a car is navigating from A to B, the device sends the corresponding coordinates to the BO. The BO calculates all possible routes and returns the fastest. The location of B should be chosen wisely. If we are driving to Rome for example, we do not want to give predictions for the traffic lights in Rome the moment we leave Eindhoven. So B can be located on the original route and will be bounded by distance or estimated travel time. The device will send the new locations of A and B repeatedly to check if a faster route is possible.

For the fastest route the BO will send the profiles of corresponding traffic lights, current states and predictions for travel times between intersections. The estimated travel time between intersections can also be stored in the device (depending if the data is dynamic; derived from HD Traffic). With the data, the device can calculate the traffic light phase predictions and visualize speed advices. Every x seconds the device has to receive the new current state and update the prediction. If x decreases, the quality of the prediction improves, but more calculation power and data transmissions are needed. See Section 5.2.3 for information about the update frequency. During a trip the car can send the coordinates of A and B more often to see if a new fastest route can be found. In the BO, the database and current states can be verified with FCD/V2V (if traffic light feeds from third parties are used).

In Figure 64, the implementation of option 3 is visualized. On the left side, an example of a network between A and B is drawn. The $K_{n,i}$ represents traffic light i on route n. Define $T_{n,i}$ as the travel time between intersection $i-1$ and i for route n. On the right, we see how option 3 is applied if route n is chosen as fastest route.

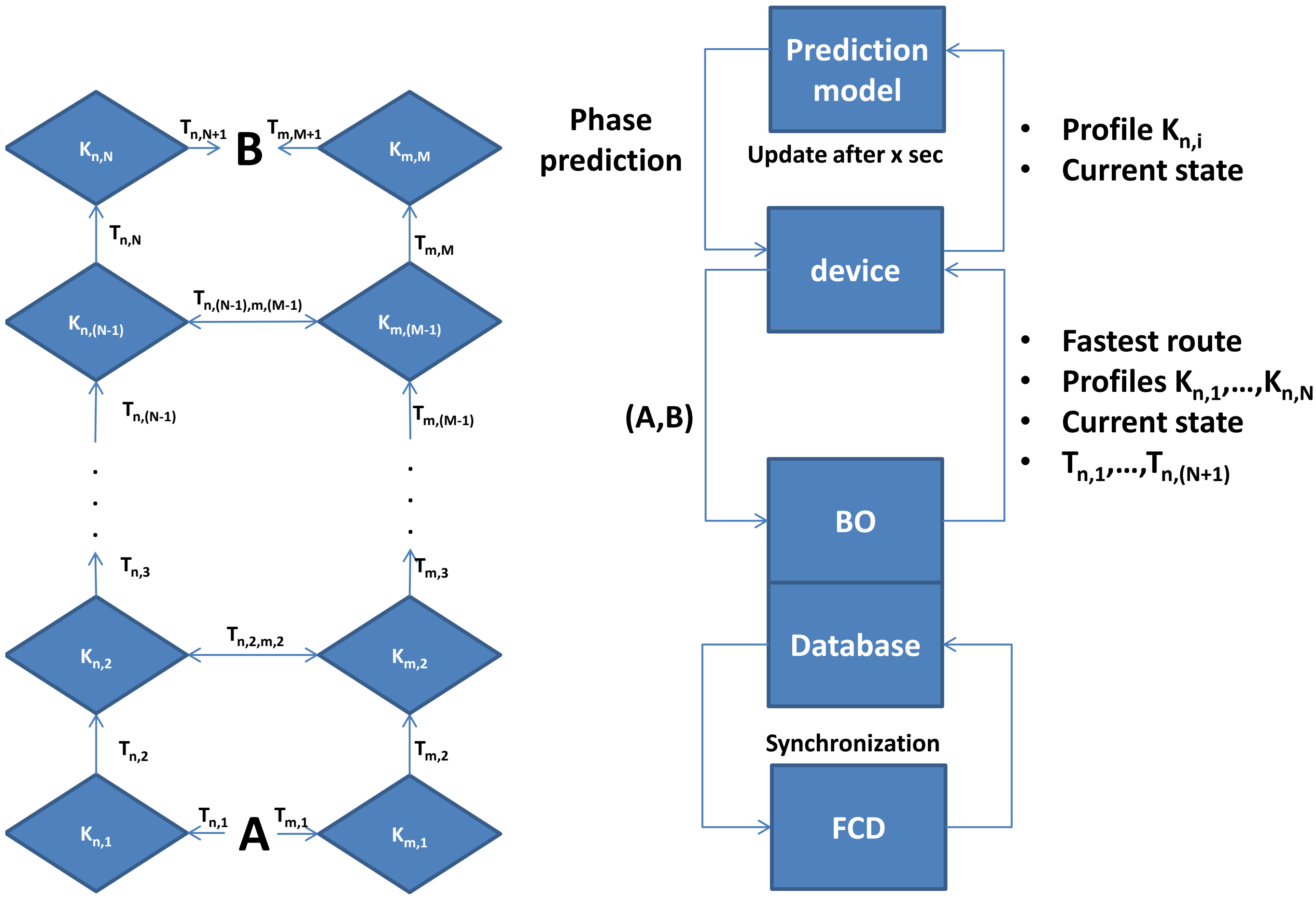


Figure 64: Possible option for implementation

8 Future research

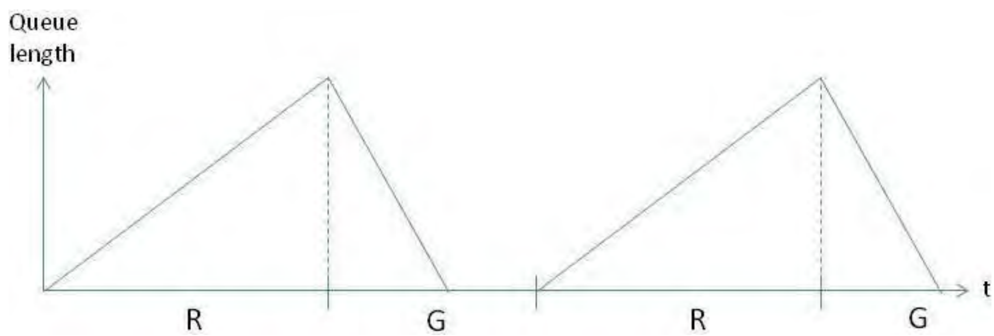
At this moment, the prediction models already give good results. But there is room for improvement.

In the thesis it is mentioned that traffic light data can come from FCD or V2V communication. At this moment we are able to determine the cycle for a statically managed traffic light. This is successfully implemented for an intersection in Portland, but it needs more testing and the speed of the method can be improved. See Section 4 for the details. For dynamically managed traffic lights, the model still needs to be adjusted to give approximations which are accurate enough to give reliable traffic light phase predictions.

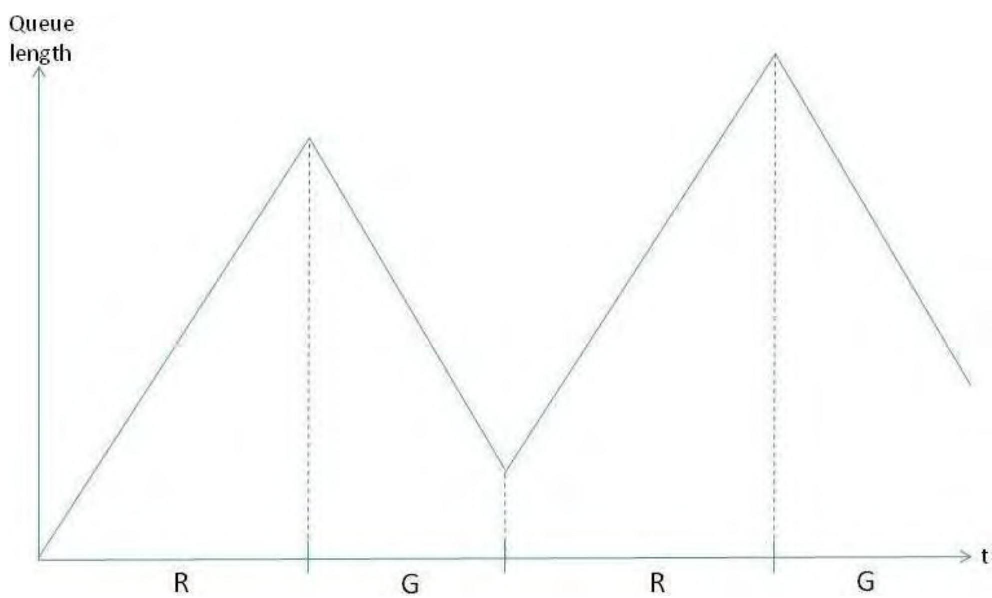
In Section 6 we mentioned that our faster route advices can improve the network. For public interest, it will be useful to measure how traffic light predictions can improve the network and can save fuel and result in lower emissions.

If the traffic light phase prediction will be tested in TomTom devices, several mathematical decisions have to be made for a good consideration between performance of the predictions, calculation power and amount of data that have to be stored/sent. These issues are discussed in Section 7.

It is still essential to add queuing theory to the prediction model because queues at traffic lights lead to extra waiting times, which can let the driver miss a green light. A queuing model should be developed which uses TomTom's HD traffic flow and traffic light data. Below we present a proposal how the queue can be modeled. If we know the green/red cycle, we can predict when the queue length will increase (during red time) and when the queue will dissolve (during green time). The current arrival rate of vehicles can be predicted from HD Traffic. It is also possible to use historical traffic intensities and involve HD Traffic if this is significantly different. The dissolving rate can be derived from historical research. If the traffic intensity is low enough, the traffic light is able to handle all the arriving cars during the green times. If the traffic intensity is too high, the traffic light becomes saturated and the queue length will increase over time. This will give huge predicted delays at traffic lights and we can probably advise to avoid this intersection. The predicted queue length is illustrated in Figure 65.



(a) The traffic light can handle the predicted traffic flow



(b) The traffic light cannot handle the predicted traffic flow

Figure 65: Prediction for the queue length

The queue length predictions can be converted to extra delays for the traffic light phase predictions to improve the estimated travel times.

References

[1] Boon, M.A.A. (2011) *Polling Models, From Theory to Traffic Intersections*. Eindhoven, The Netherlands

[2] van den Broek, M.S. (2004) *Master's thesis Traffic signals, Optimizing and analyzing traffic control systems*. Eindhoven, The Netherlands

[3] van Leeuwaarden, J.S.H. (2006) *Delay Analysis for the Fixed-Cycle Traffic-Light Queue.* Transportation Science, Vol. 40, 189-199.

[4] Ross, S.M. (1996) *Stochastic Processes*. John Wiley & Sons, New York, USA

[5] TomTom: http://www.tomtom.com

[6] Webster, F.V. (1958). *Traffic Signal Settings*. Road Research Technical Paper No. 39, London, UK.

[7] Wilson, A. (2006) *Handboek verkeerslichtenregelingen.* CROW, Ede, The Netherlands.

A Expected waiting time until next green light

Define r as the probability density function of R_j. The expected waiting time until next green light in Equation (10) can be calculated exactly by:

$$\mathbb{E}[G_{is} - t | G_{is} < t + R_i \text{ and } G_{is} > t] = \begin{cases} \mathbb{E}[R_1] - a - t & \text{if } i = 1, \\ \dfrac{\displaystyle\int_{y=0}^{\infty} \left(\int_{x=t+a}^{x=t+a+y} x g_{i-1}(x) dx \right) r(y) dy}{\displaystyle\int_{y=0}^{\infty} \left(\int_{x=t+a}^{t+a+y} g_{i-1}(x) dx \right) r(y) dy} - a - t & \text{if } i > 1. \end{cases}$$

B Results of implementation in Portland

B.1 Implementation in Portland, 3:33:19 PM - 6:45:37 PM

Below we see the output results of the algorithm during evening rush hour. In this implementation the current time is 6:45:37 PM ($t = 0$).

Green times phase 2 and 6 from traces:
-4151.0
-10116.0
-4879.0
-4113.0
-8935.0
-5843.0
Green times phase 4 and 8 from traces:
-6065.0
-3167.0
-8368.0
-4085.0
After modulo calculation the green times of phase 2 and 6 become:
-79.0
-51.0
-43.0
-35.0
-16.0
-13.0
After modulo calculation the green times of phase 4 and 8 become:
15.0
32.0
33.0
35.0
The green points of group 2 and 6 after red driver filter (notice that -79.0 was detected as a driver through red light):
-51.0
-43.0
-35.0
-16.0
-13.0

The green points of group 4 and 8 after red driver filter:

15.0
32.0
33.0
35.0

Estimation green start phase 2 and 6 from algorithm: -57.0 ⇒ 6:44:40 PM
Green start phase 2 and 6 from Green Driver data: 6:44:42 PM

B.2 Implementation in Portland, 8:49:10 AM - 3:31:11 PM

Below we see the output results of the algorithm outside rush hour. In this implementation the current time is 3:31:11 PM ($t = 0$).

Green times phase 2 and 6 from traces:

-9535.0
-18232.0
-3377.0
-2948.0
-9738.0
-14336.0
-8081.0

Green times phase 4 and 8 from traces:

-17757.0
-4467.0
-8795.0
-12230.0
-11111.0
-17493.0

After modulo calculation the green times of phase 2 and 6 become:

-56.0
-32.0
-31.0
-17.0
-15.0
-8.0
-8.0

After modulo calculation the green times of phase 4 and 8 become:

7.0
13.0
19.0
20.0
23.0
25.0

The green points of group 2 and 6 after red driver filter (notice that -56.0 was detected as a driver through red light):

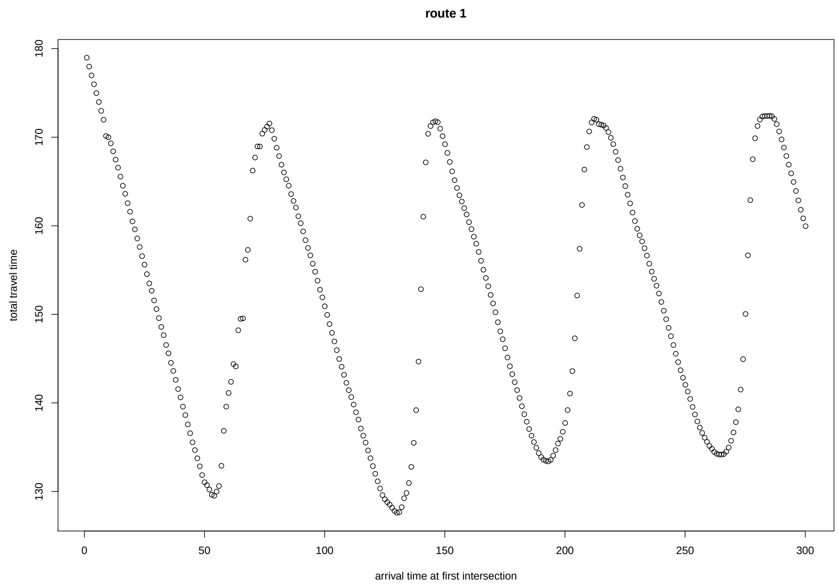
-32.0
-31.0
-17.0
-15.0
-8.0
-8.0

The green points of group 4 and 8 after red driver filter:

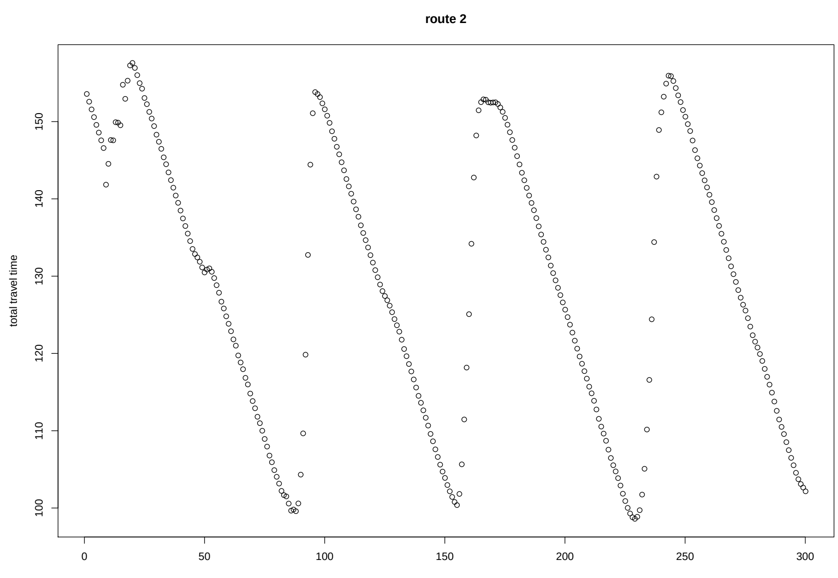
7.0
13.0
19.0
20.0
23.0
25.0

Estimation green start phase 2 and 6 from algorithm: -38.0 ⇒ 3:30:33 PM
Green start phase 2 and 6 from Green Driver data: 3:30:36 PM

B.3 Network predictions in Portland

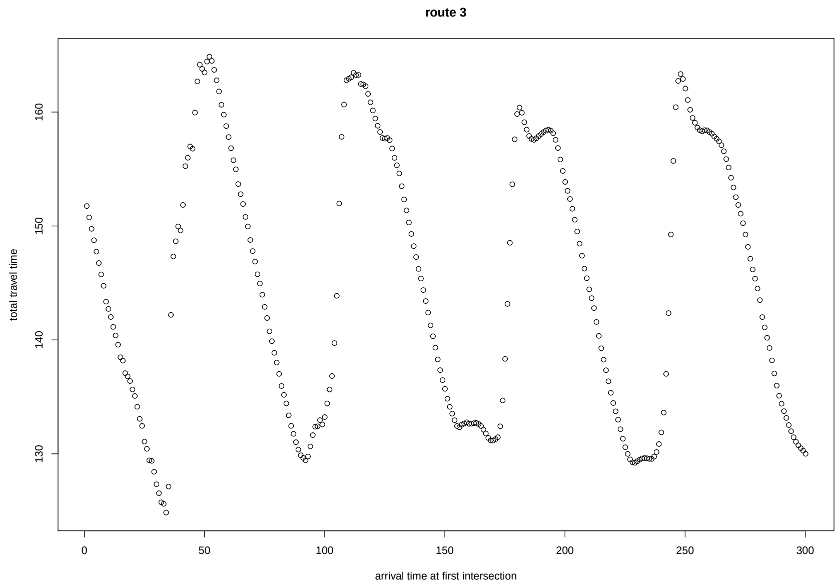


(a) Sojourn time route 1

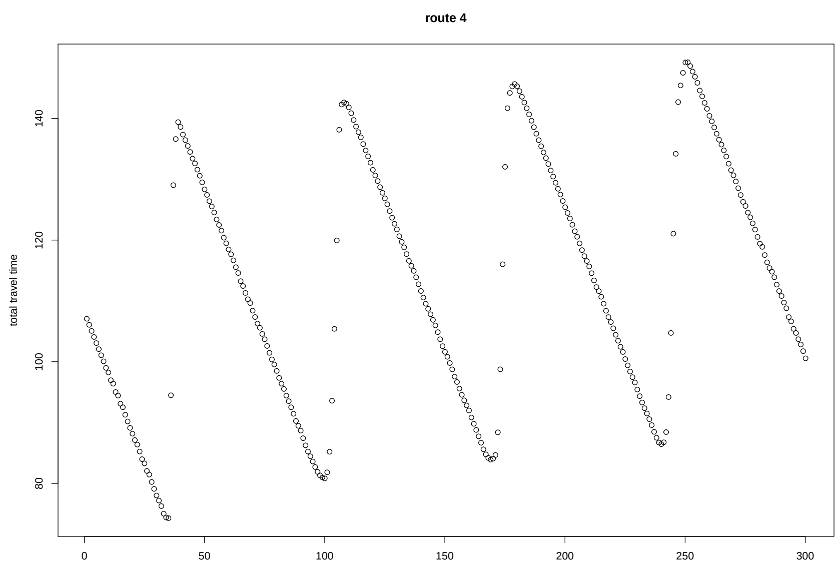


(b) Sojourn time route 2

Figure 66: Sojourn time of routes in network

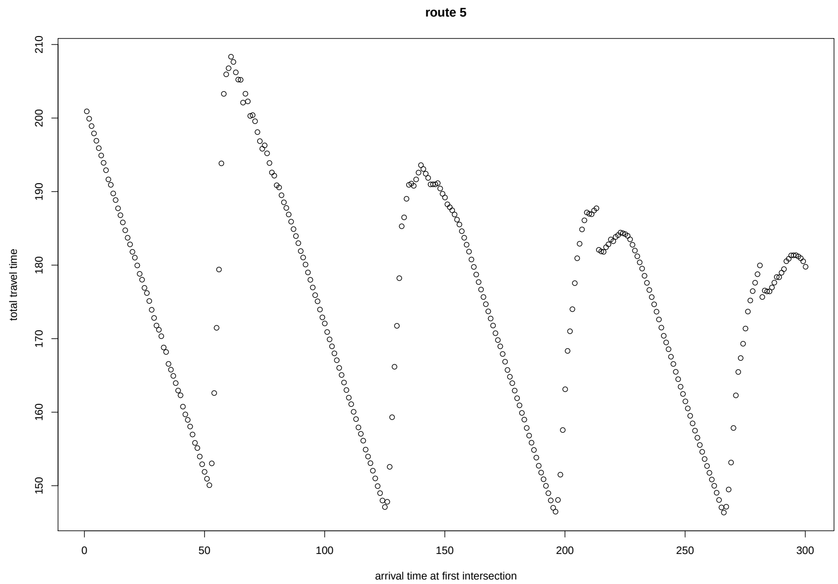


(a) Sojourn time route 3

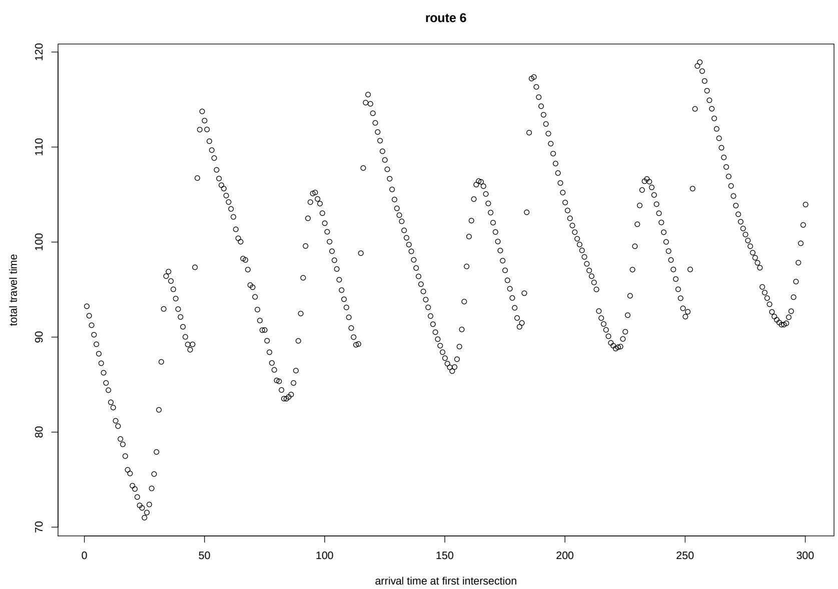


(b) Sojourn time route 4

Figure 67: Sojourn time of routes in network

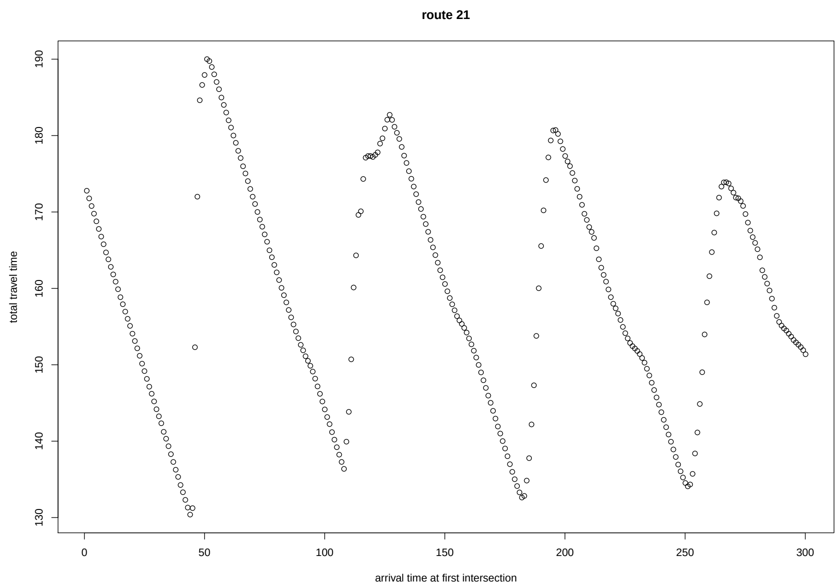


(a) Sojourn time route 5

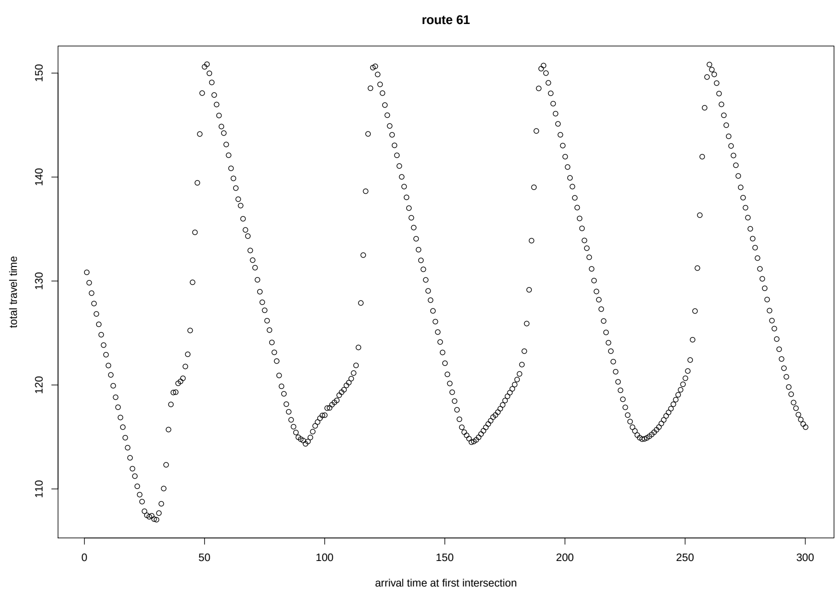


(b) Sojourn time route 6

Figure 68: Sojourn time of routes in network

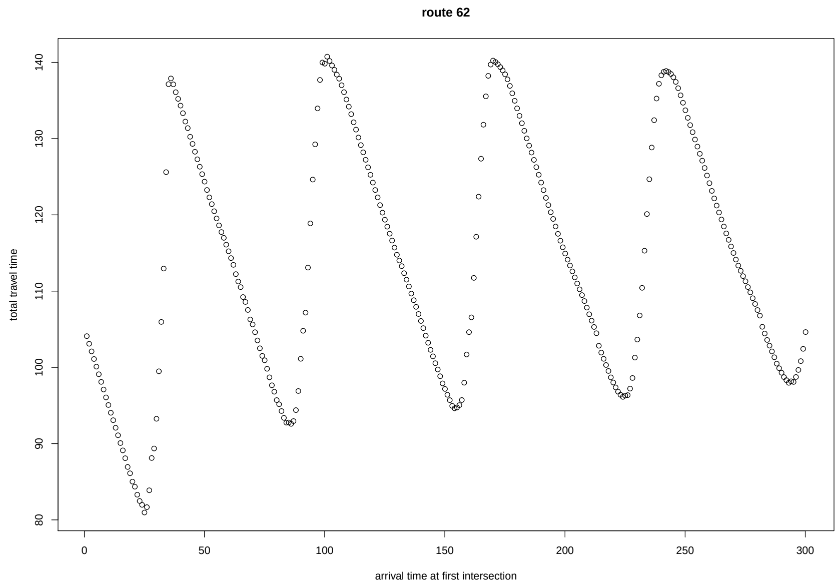


(a) Sojourn time route 21

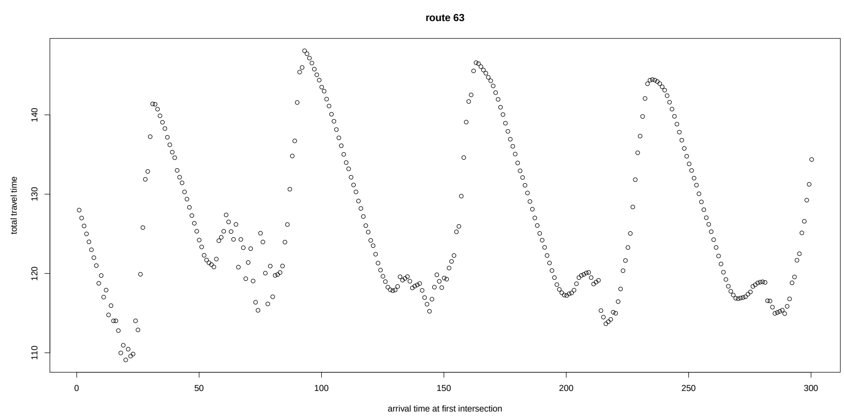


(b) Sojourn time route 61

Figure 69: Sojourn time of routes in network



(a) Sojourn time route 62



(b) Sojourn time route 63

Figure 70: Sojourn time of routes in network

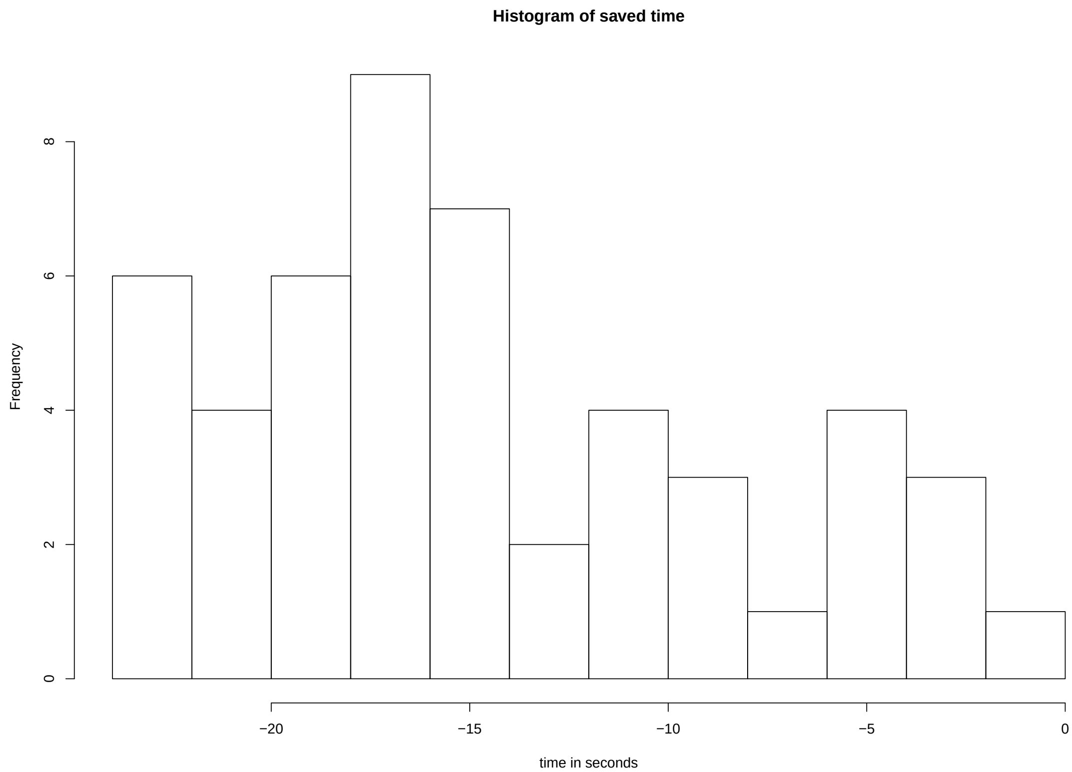


Figure 71: Histogram of time saved in network