STEP (Short TErm Prediction)

Workpackage 4B:

Netherlands Pilot of Short Term Predictions

May 2013

Project Nr. **832564**

Project acronym: **STEP**

Project title:

Short TErm Prediction (STEP)

Deliverable Nr 4B – NETHERLANDS STEP PILOT

Due date of deliverable: May 2013

Actual submission date: 24.05.2013

Start date of project: 01.10.2011 End date of project: 31.05.2013

Author(s) this deliverable:

Chris Van Hinsbergen, Fileradar, The Netherlands

Frank Zuurbier, Fileradar, The Netherlands

Version: Final

Executive summary

The ERA-NET ROAD Mobility research project "STEP" implemented short term prediction within traffic management centres to support traffic management decisions and ultimately improve network performance. STEP has explored the gaps between the state-of-the-art and requirements of operators in terms of functional application, data requirements, interfacing and the success of existing tools that are used. Central to the project were real-life trials conducted in an operational traffic management centre environment, testing the tools against user requirements while learning valuable (practical) lessons during implementation.

This document sets out the background to the pilot phase of the study undertaken in the Netherlands within a traffic control centre environment in Utrecht. A number of key tasks were progressed as part of the study including development of a detailed pilot plan that built in the necessary characteristics of traffic predictions and their operational use into the application as well as preparing data to support the predictions. A key element of the work was the configuration of the system undertaken by connecting traffic prediction software developed by Fileradar to available data and the existing system used in the Netherlands Traffic Control Centres. Finally, the work involved running with a live predictor for a period during which time the outputs were continually monitored to assess the accuracy of the predictor as well as periodic recalibration of the system as required.

At the outset of the project discussion took place with Rijkswaterstaat, the Dutch national road administration on involvement in the research work and it was found that both VCNL (the national traffic control centre) as well as VCMN (the regional control centre for the centre region of the Netherlands) were interested in hosting a pilot to test the functionality of the STEP prediction tool in a TCC setting. These two control centres share a building on the outskirts of Utrecht.

Initially, it was agreed that an attempt would be made to run a pilot scheme in both control centres simultaneously. Both the national and regional control centre perform different tasks and therefore feedback from both of these would present different views on how best to apply short term predictions to support daily tasks. At the same time work was initiated on setting up the prediction model including a server where the predictions were to be made, the collection of regional network data for the prediction software, calibration of the model for the region around Utrecht using automatic calibration routines and also the development of a web client.

An initial delay in commencing delivery of the Netherlands Pilot meant that it ran between November 2012 until March 2013, including a period during which the client was offline to allow for the development of a new version and improvements to the predictions. These changes were made on the basis of discussions with TCC operators/staff during the pilot period.

The pilot itself sought to gain a better understanding of several fundamental elements such as user acceptance and trust of the prediction tool, the development of an appropriate interface that meets user needs, as well as technical deployment and the quality of predictions.

Based on the findings of the pilot a number of essential aspects for short term predictions was identified. Throughout the pilot phase regular communication was established with the TCC operators who were using the system, which helped to establish a rapid response approach to dealing with specific issues that arose ensuring support engagement from both front-line staff and management. In addition, interactive sessions with TCC personnel were also hosted to detail and improve the predictor's visualisation. This close co-operation with the TCC management and users helped to provide a clear indication of the essential aspects

of short-term prediction that affect its use and acceptability. These include the following:

- it is important that the actual traffic flow prediction is of a very high quality, as TCC staff considered only the first 15 minutes to be of adequate quality to serve their needs. (The deterioration after longer lapse times were found to prevent the predictions being used for longer term forecasts);
- the preferred option for the presentation of prediction results selected by TCC staff was found to be through use of a split screen with actual traffic levels and animated predicted situation shown side-by-side;
- data transfer latency is critical when implementing short-term prediction tools, with TCC staff often comparing predictions with existing real-life data that they are used to seeing on their screen. The data volumes are significant when compared with real-time traffic data because each updated prediction contains not just a single snapshot in time but a series in time from now into the future. Considerable time was spent on streamlining communications elements of the system so that staff felt comfortable with the outputs.
- non recurrent traffic congestion and quantitative descriptions of that (in terms of minutes delay, length etc) were the most interesting, but also the hardest to achieve; and
- the provision of an aggregated graph showing the average and current total km network queue is considered of value by TCC staff.

Table of content

List of Figures

1 Introduction

"ERA-NET ROAD – Coordination and Implementation of Road Research in Europe" was a Coordination Action funded by the 6th Framework Programme of the EC. The partners in ERA-NET ROAD (ENR) were United Kingdom, Finland, Netherlands, Sweden, Germany, Norway, Switzerland, Austria, Poland, Slovenia and Denmark (www.road-era.net). Within the framework of ENR this joint research project was initiated. The funding National Road Administrations (NRA) in this joint research project are Belgium, Switzerland, Germany, Netherlands Norway and United Kingdom.

The research project as a whole is described in the STEP Project Inception Report (Ref [1]). The report is aligned with Work Package 3 "WP3 - Preparing Pilot Software Development of tools for scenario development and decision support".

This report describes the process and the results of Work Package 4B (WP4B) of the STEP (Short TErm Prediction) project. This work package consisted of a real life pilot where short-term traffic predictions were shown to traffic controllers in the National Traffic Control Centre ('VCNL') in Utrecht, the Netherlands to allow them to anticipate on future traffic congestion.
The goal of this work package was to find out what is needed in order to successfully deploy short-term predictions in a traffic control centre. During this pilot an actual application was developed and deployed in a traffic control centre. In this process many valuable lessons have been learned that will be shared in this report.

The remainder of this report consists of the following sections. First of all, the process of preparing for the pilot will be described. Next, the application that was used during the pilot will be introduced. Then, the pilot itself will be described from the moment it started until the moment it ended and what happened in between. An evaluation of the pilot is presented, followed by the conclusions.

2 STEP User Environment

2.1 Preparations for the trial

The first plans for the pilot were already made before the actual start of the project. When preparing for the application of the STEP project contact was made with Rijkswaterstaat, the Dutch national road administration, specifically with Henk Schuurman, Jose Hernandez and Jos van Hees. With them, an agreement was made that if the project would proceed, a Dutch pilot would be feasible and one of the Dutch traffic control centres would be sought where the pilot could be held.

Once the ERA-NET Road grant was assigned, it was found that both VCNL (the national traffic control centre) as well as VCMN (the regional control centre for the centre region of the Netherlands) were interested in hosting a pilot. These two control centres share a building in the city of Utrecht.

Table 1 Overall Timetable for the Development of the WP4B Pilot Scheme

Date	Event
March 2011	First contact with Rijkswaterstaat about potential pilot
June 2011	Acceptance of proposal by ERA-NET ROAD
August 2011	Agreement of pilot locations: VCNL and VCMN
November 2011	Kick-off meeting with Rijkswaterstaat
March 2012	Meeting with Rijkswaterstaat
June 2012	Brainstorm on User Interface
October 2012	Introduction to VCNL personnel
November 2012	Introduction to VCMN personnel
November 2012	Pilot starts at VCNL

Once these locations were agreed upon, a meeting was held with two representatives of Rijkswaterstaat (Henk Schuurman and Jose Hernandez). In this meeting, a few agreements were reached on how to best approach the pilot. In order to maximize the chance of success, the end users would need to be explicitly approached to participate in designing the contents and the user interface of the pilot application. Furthermore, to prevent running into IT-related problems, the pilot software would need to be made accessible via a web browser, so no installation of software was necessary. After this meeting, the Rijkswaterstaat representatives arranged for a formal “Go” of the pilot based on a short description on the pilot and an estimate of what would be needed in terms of man-hours.

After the formal agreement was made a discussion was held in which of the two interested control centres the pilot would run. After a discussion with Rijkswaterstaat representatives (again Henk Schuurman and Jose Hernandez) it was agreed upon that an attempt would be made to run the pilot in *both* control centres simultaneously. After all, the national and regional control centre perform different tasks:

- The national control centre is responsible for information dissemination, coordination between regional control centres and rerouting of traffic; and
- The regional control centre is responsible for incident management and the control of dynamic traffic management systems, such as peak hour lanes.

They would therefore have a different view on how best to apply short-term predictions in their daily work.

Work was initiated on setting up the prediction model (Work Package 3[2]). A server was set up where the predictions were to be made; the network and data for the region were collected and made available to the prediction software (see Appendix B: The Study Area); the model was calibrated for the region around Utrecht using automatic calibration routines; and a beginning with the web client was made.

In June 2012, a few months before the anticipated start of the pilot, a brainstorm meeting was held in Utrecht with both Rijkswaterstaat representatives as well as managers of both VCNL and VCMN. It was agreed upon that during the pilot the short-term traffic predictions would at first be used as a reference tool. It was not considered feasible to show more proactive advice of the system on which control measures to take or which routes to advice: first of all because of time and budget constraints of the pilot itself, but second of all because it was said that in order for users to trust the application, they would first need to trust the predictions themselves; i.e. the users would first like to see the predictions 'as is' and estimate their value before trusting advice given by such a system. This relates to the outcomes of WP2[3].

With management and operational staff a brainstorm was held on how to best present the predictive traffic information. The following ideas were generated during the brainstorm:

- Present the predictions in a split screen showing two maps: the current situation on the left and an animation of the predictions on the right;
- Daily reports on the accuracy would be appreciated;
- Visualisation of the remaining capacity on corridors; and
- Alerts when and where spillback will occur at certain predefined locations.

During this meeting it was decided that the client would first be tested in the national control centre VCNL, and when all teething problems were solved and the predictions were of a high enough standard, the client would also be tested in the regional control centre VCMN. However, during the pilot that started a few months later, it was decided not to test the client at VCMN after all, because all time available was invested in improving the prediction quality (see section covering the details of the Pilot).

Based on the brainstorm session a first 'prediction viewer client' was created. This demo client was then used as a starting point in a discussion with actual traffic controllers of the national control centre VCNL. In a meeting in October 2012, where 6 traffic controllers and the head of operations were present, the demo client was presented. This provided an opportunity to enthuse the traffic controllers for the new tool. Moreover, useful feedback was obtained from this group of end users:

- the traffic controllers are sceptical towards new tools because they have already seen so many of them; however, they all see the potential of the client but would like to be convinced;
- previous software tools lacked active support from their developers, which caused the tools to become unused in the end;
- the end users see the most potential in predictions of non-recurrent situations;
- some improvements were suggested for the colours and the location of components of the interface;
- some additional links were requested to be included in the area on which the predictions were made; and
- A long desired feature was to have an overview of the current and historic average queue

[2] See Deliverable Nr 3B – User Interface Specification

[3] See Deliverable Nr 2 – User Requirements

lengths.

The proposed changes and additions were all taken into account and a new version of the client was developed. A few weeks later, the first version of the client was ready for testing. A few initial attempts to run the web client from the browsers of the control centre revealed some technical issues; these were all resolved, after which the client was successfully put online on 15 November.

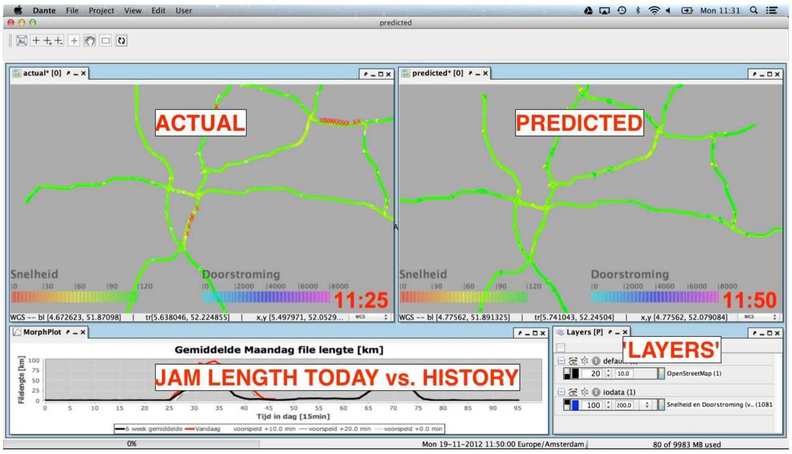


Figure 1 Annotated Screenshot of Initial STEP Client

In the next section, the application that was used during the pilot will be described in detail. Then, the course of events during the pilot itself will be explained, including issues raised by TCC staff and adjustments made to the tool.

3 The STEP Application

3.1 Introduction

On 15 November the pilot was started in the national control centre. Figure 2 shows the location of the STEP client in situ within the traffic control centre itself. The desk in the middle is the desk where the traffic situation around Utrecht is monitored 24 hours a day, seven days a week. The TCC personnel work in shifts, and each traffic controller manages a different region every shift. Therefore, many different controllers had the chance to view and use the tool. The red circle indicates the screen where the application was running for several months. The application ran in parallel to other, existing tools. However, none of these tools shows any predictive information.

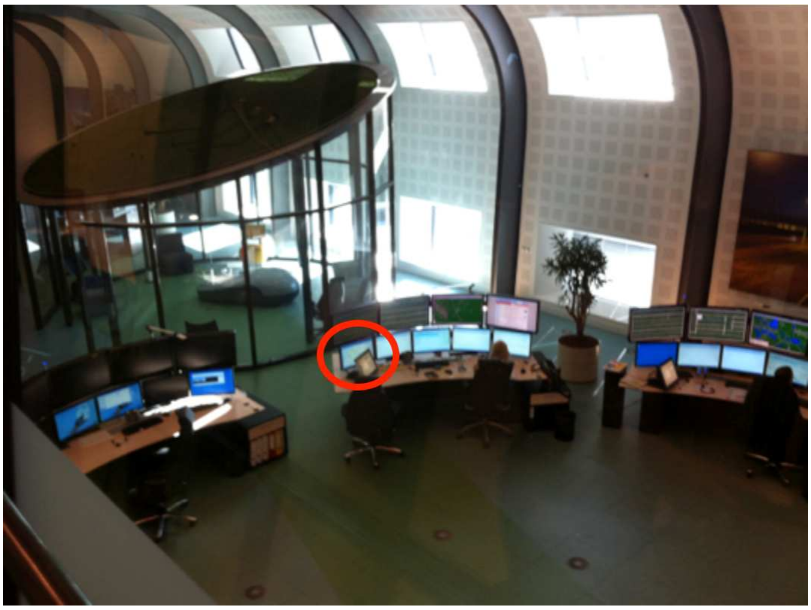


Figure 2 Overview of the Utrecht Traffic Control Centre and STEP Client

In this section the application that was used for the pilot in the traffic control centre is described. As indicated in the previously, this client was developed in co-operation with the managers and end users of the traffic control centre.

3.2 Dante

Both the client and the prediction algorithms are written in a software environment called Dante[4]. This programme is developed and maintained by Fileradar BV. The software is written in Java and is designed to combine and connect large amounts of geographic and dynamic data. Due to its modular structure, new sources of data and new functionality are easily added. Each new piece of functionality is implemented as a different module or ‘bundle’. Figure 3 illustrates this concept. For the STEP project a few additional bundles have been created which are placed on top of existing bundles. Special STEP bundles are used at both the server side to set up the prediction model for this pilot, as well as on the client side to create an interface in which traffic controllers can see live traffic predictions.

[4] See www.dantesoftware.com

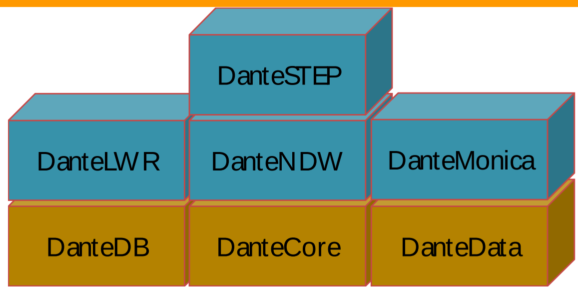


Figure 3 Modular Structure of Dante

3.3 Starting the Application with Java WebStart

In the previous section it has already been mentioned that it was agreed upon that the pilot software needs to be accessible via a web browser. For this, Java WebStart™ technology is used. WebStart allows for a Java program, such as Dante (which allows for easy addition of new functionality or data), to be downloaded and started by just clicking a link in a browser. Figure 4 depicts this process schematically.

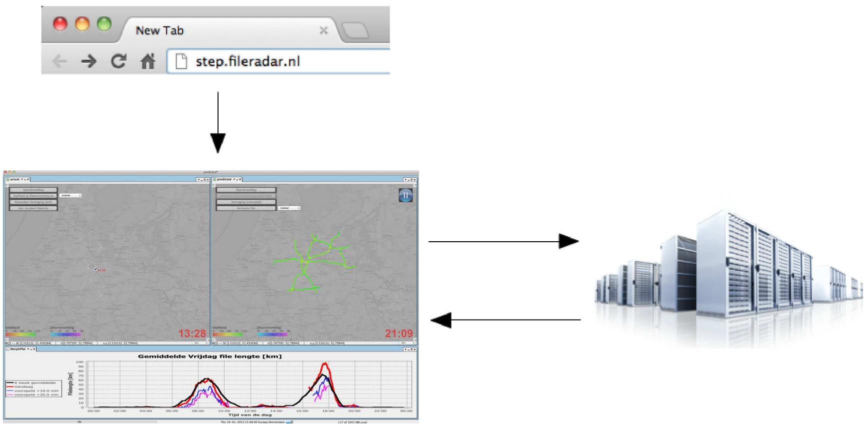


Figure 4 Summary of the Process for Starting Dante

The Java WebStart technology has a few important advantages that aided in successfully launching the client in the control centre:

- It is cross-platform, so that the software can run on any version of Windows, MacOSX, Linux or any other common operating system;
- No installation is needed and no system files or registry entries need to be changed; therefore no administration rights are needed to run the program; and
- Every time the user clicks the link in the browser, WebStart automatically checks if a new version of any of the bundles is available and downloads them if so; the user therefore always has the latest version of the software available.

The overall process involves the user visiting a website, clicking a link, waiting for the latest bundles to be downloaded after which Dante starts, automatically contacting the Fileradar server for the latest data. Once WebStart has launched the client, Dante automatically connects to the Fileradar servers to download the latest actual and predicted data. This data is then displayed in the interface that will be described later.

3.4 The Prediction Algorithm

The predictions used in the pilot are generated by a combination of algorithms and models.

The core model used is the Lighthill-Whitham-Richards model [5,6]. Traffic is modelled as a fluid stream, where 'bottlenecks' form regions where the 'fluid' flows slowly, i.e. where traffic congestion occurs. The model was already developed in the 1950s, but has proven to be powerful in predicting the global dynamics of traffic jams in many scientific studies[7].

The hard part however is to calibrate the model for a large-scale network such as the one under study in this pilot. Despite the models simple form, there are many thousands of parameters to find: the demand at each onramp at each time of the day, the average route choice at each intersection or off ramp in the network, the capacity of each section of the road, the average distance between vehicles when flow is at capacity, et cetera. To make matters even more complex, these parameters are stochastic and dynamic by nature. For example, the average distance that people keep to their predecessor determines the capacity of a road. The capacity therefore depends on human behaviour, introducing a stochastic component. Furthermore, the capacity changes over time as a function of the weather, lighting conditions, road works or incidents.

Since 2010 Fileradar has been developing advanced routines to perform the entire calibration automatically. Some of the routines have proved themselves since then and provide excellent results, such as the algorithms used to predict demand and route choice. Others were, during the course of the pilot, still in the experimental phase, such as the *capacity calibration algorithm* to automatically determine the capacity at bottleneck locations. In short, this algorithm detects queues and then tries to replicate the observed dynamics of each queue in the prediction model by changing the capacities of relevant segments. This way, after a short period of observation, predictions can be made of any queue, recurrent or non-recurrent, and the dynamic nature of the capacity is automatically taken into account. As the focus of the control personnel was especially on the management of non-recurrent situations, this is a highly desirable feature.

In preparation for the pilot Fileradar worked together with KU Leuven to analyse the performance and properties of the experimental capacity calibration algorithm. After an initial sharing of the basic concept, both Fileradar and KU Leuven independently worked on improving the algorithm and after a few months shared their outcomes. In Appendix C: Automatic Capacity Estimation Algorithm a brief overview is given of the resulting algorithm that was developed by Leuven. Elements of these outcomes have been taken into account into the algorithm that was eventually applied during the pilot.

In order to have useful results for traffic controllers, both the length of the jam and the speed of vehicles in the jam need to be predicted more or less correctly by the model. The capacity calibration algorithm that was used during the pilot works really well on isolated cases, where a single queue occurs due to a single bottleneck. Unfortunately, the network of Utrecht is a network with very complex traffic congestion patterns: spillback over intersections, interaction between different bottlenecks and so-called *moving jams* that all occur on a regular basis. The capacity calibration algorithm was not able to cope with these complex patterns. As a result, the predictions were not as good as hoped for when the pilot was planned. In Appendix A: Analysis of Prediction Quality During the Pilot the quality of the predictions is further analysed and quantified.

---

[5] *Lighthill, M.J., Whitham, G.B., 1955. On kinetic wave II: a theory of traffic flow on crowded roads. Proceedings of the Royal Society of London, Series A, 229 (1178), 317- 345*.

[6] *Richards, P.I., 1956. Shock waves on the highway. Operations Research, 4 (1), 42-51*.

[7] See for example *Daganzo, F, 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory, 28 (4), 269-287*, or *van Hinsbergen, C.P.IJ., van Lint, J.W.C. and Sanders, F.M. Short Term Traffic Prediction Models, Proceedings of the 14th world congress on ITS, Beijing, 2007*.

3.5 The Different Components of the Application

A screenshot of the first version that was launched during the pilot is shown in Figure 5. Using the annotations in this figure, the different components of this client will be explained below.

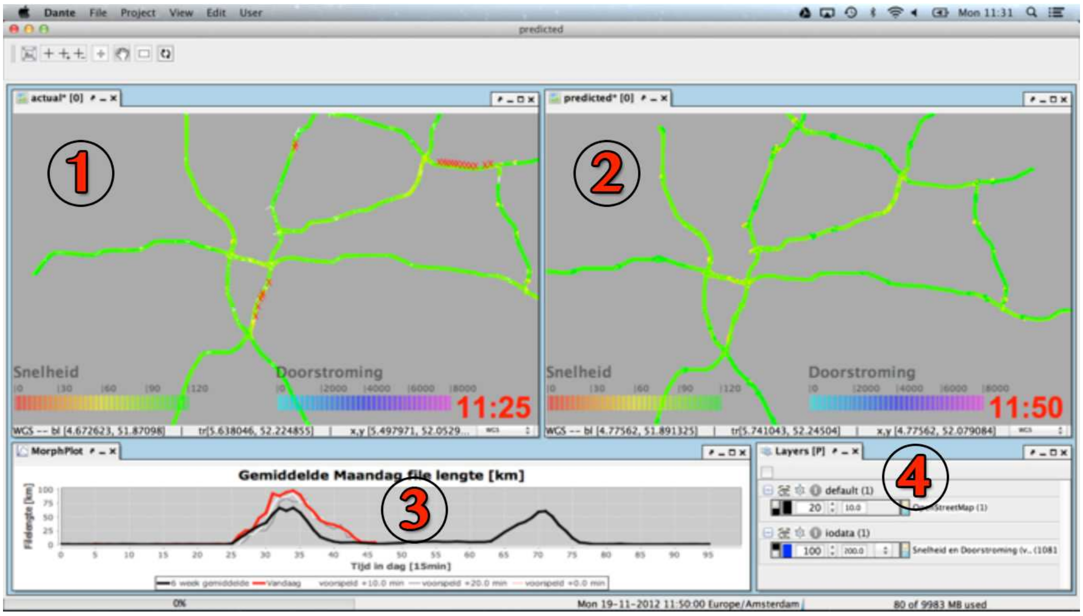


Figure 5 Initial Version of the Web Client used in the Netherlands STEP pilot

The Actual Situation

The left side of the split screen shows a map that works just like Google Maps or any other well-known interactive map. The user can zoom in or out and pan using his or her mouse. The last measured situation on the road network of interested is projected on this map. Rather than showing the raw detector information, data is interpolated over the network links using the Adaptive Smoothing Method [8]. This way, a speed and flow can be estimated even in between detectors, so that a smooth 2D-map of speeds or flows is visible.

As a data feed real-time data of the *NDW* [9] is downloaded every minute on a data server hosted by Fileradar. The data is then sent to a prediction server where the ASM algorithm is applied and predictions are made. Finally, the client in the control centre checks for new data every 10 seconds on this prediction server and downloads data when available. The map is then automatically refreshed, and the time (at the bottom right of the map, in red) is updated. On the bottom of the map a legend is also shown that translates the colour gradient to speed or flow values.

In the first version of the client a total delay of 7 minutes occurred before the real-time situation was visible on the map. After feedback of control centre personnel that this was longer than they experienced in other tools, a few bottlenecks in the process were identified and removed, so that the total delay reduced to 3.5 minutes.

The Predicted Situation

The map on the right shows an animation of the real-time predictions. The same server that interpolates the real-time data runs the prediction algorithms every 5 minutes. The experimental algorithms that were used in this pilot to make the predictions were already described earlier in this section.

---

[8] Reconstructing the spatio-temporal traffic dynamics from stationary detector data, M. Treiber and D. Helbing, Cooper@tive Tr@nsport@tion Dyn@mics, 1, 3.1-3.24 (2002)

[9] The Dutch National Data Warehouse, see http://www.ndw.nu

When the pilot started, the animation showed only the predictions. Inspired by feedback given by personnel, later in the pilot the recent history was added to the animation so that 40 minutes history and 20 minutes prediction were shown in one animation. The time in the lower right corner updates during the prediction, and changes colour when the historic data proceeds in predictive data.

During the pilot it was found that the quality of the predictions resulting from the experimental real-time calibration algorithm degraded too quickly with an increasing prediction horizon; this is in accordance with the feedback given by control personnel. Therefore, the predictions in their current form were not good enough to aid them in their decision-making on a regular basis. A detailed evaluation of the prediction quality is given in Appendix A: Analysis of Prediction Quality During the Pilot.

The Queue Lengths Plot

This feature was a special request of the operational employees of the traffic control centre. The chart shows today's total queue length as a function of the time and compares it to the historic average queue length in the network over the day. In one quick view the traffic controllers have an idea if, on the network as a whole, today is busier or less busy than average.

To compute the historic average queue length each night the last 6 comparable days are taken and the ASM filter is run over all days. A comparable day is a day that is the same day of the week; i.e. if today is a Monday, than the 6 previous Mondays are taken; public holidays are excluded. The total queue length is determined in 15-minute intervals, which is then shown in the plot. The actual queue length follows from the actual data received every minute by the prediction server. The plot checks for updates every 10 seconds on the server and refreshes itself as soon as it detects that new data is available.

The Layer selection

Dante is a dynamic GIS (Geographic Information System) software tool. As in any GIS application, the user can select which types of information he or she wants to plot on the map. Using this panel the user can add or remove information from the map or change the type of information to be displayed; for example he or she can switch between *speeds* or *flows* to be displayed. The layers of information that were available during this pilot are shown in Table 2. One layer was added during the pilot: the Automatic Incident Detection layer.

The incident detection algorithm is designed by the regional control centre VCMN and tested successfully on a short stretch of road. The algorithm is based on a combination of speed and flow information both upstream and downstream of a certain location. Because Dante can so quickly be extended with functionality, the incident detection algorithm was implemented for the entire network. From the second part of the pilot this layer was available to the control centre personnel.

Table 2 Overview of all information layers that were available in the final version of the STEP client

Actual map	**Predicted map**
Open street map background	Open street map background
Actual speed / flow	Animation of speed / flow
Delays in minutes	Delays in minutes
Automatic incident detection	

During the pilot several changes have been made to the client based on feedback that was received as set out in Table 3 below. The end result is shown in Figure 6, which contains a number of adjustments compared to the first version as shown in Figure 5. In Table 3 a list of these adjustments is given.

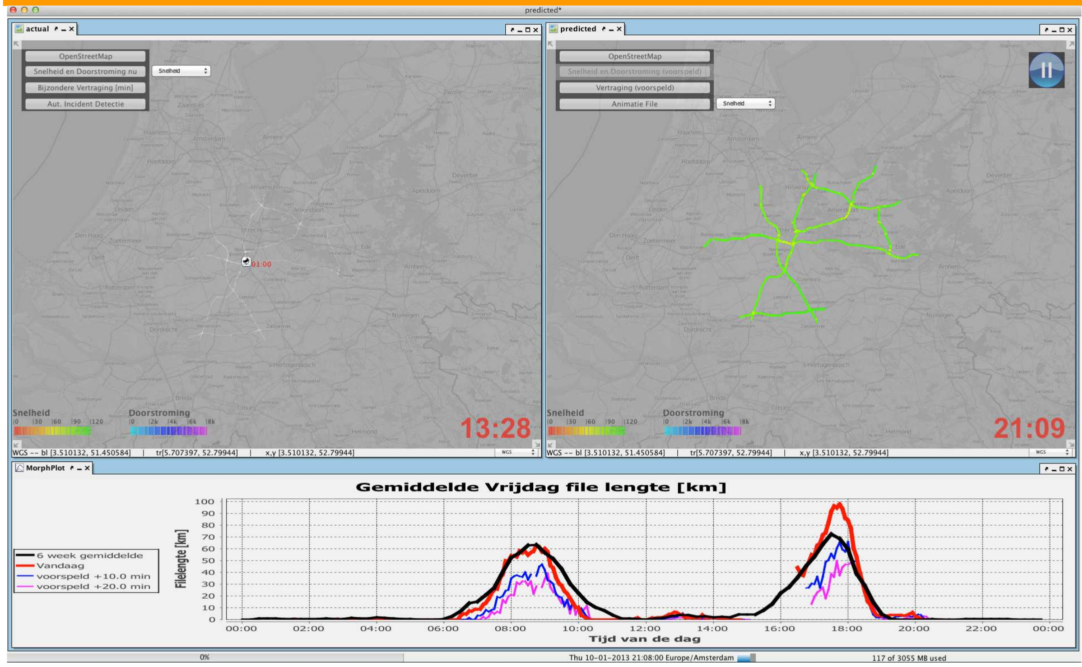


Figure 6 Final Version of the STEP Web Client Developed

Table 3 Change log of the STEP web client during the pilot

Date	Action
2012-11-15	• Client live
2012-11-20	• A button was added to pause or play the animation • Bugs resolved in configuring GUI after startup • Adjustment of colours in queue length plot • Timeliness of data improved: from 7 minutes to 3.5 minutes
2012-11-28	• Delay layer added to predicted map • Information density adapted to zoom level • Heavily reduced amount of data sent from server • Other small improvements to GUI
2013-02-05	• Renewed version of the prediction algorithm • Added 40 minutes of history to animation • Removed the layer panel and added it as 'HUD'-style information, transparently in the corners of the map • New information layer: automatic incident detection • Further reduced amount of data sent from server • Bug resolved that sometimes caused a Windows system to become unresponsive

4 The Pilot

4.1 Introduction

This section describes the events that took place during the pilot. The pilot ran for about 4 months, although the client was offline for a period of 7 weeks in the middle due to the development of a new version of the client and improvements to the predictions.

Table 4 Programme of Pilot Delivery

Date	Event
15 November 2012	Step web client live
20 November 2012	Update of web client software
22 November 2012	ERA-NET Board visits live trial at VCNL
27 November 2012	Fileradar visits VCNL
28 November 2012	Update of web client software
20 December 2012	Client offline
2 January 2013	Start of major revision of client software and of predictions
11 January 2013	Fileradar visits VCNL
7 February 2013	Fileradar visits VCNL , new client online
13 March 2013	Client offline
20 March 2013	Evaluation of pilot

The pilot officially started on the 15th of November 2012, when the client was put online. A short manual was left behind in print, together with a form on which users could leave their feedback and with contact details for support. The traffic controller on duty was instructed on how to use the client; he in turn would tell the controller in the next shift about the client, et cetera.

After a few days the first feedback was gathered during a visit to the control centre. This feedback consisted of a few examples where the predictions did not work as expected, as well as possible improvements on the user interface. In the following days improvements were made to the interface as discussed in the previous section (see Table 3), and a new version of the client was put online on 20 November 2012. The controller on duty was contacted by phone to re-launch the client.

On 22 November, a week after the first version of the client was launched, the ERA-NET board visited the control centre after their board meeting in Delft the day before. Fileradar gave a presentation about the pilot, the client and the prediction software and the head of operations of the control centre gave a tour of VCNL. The visit ended in a demonstration of the actual client in the control centre and a discussion of the board members with the traffic controller on duty.

A week later, on 27 November, Fileradar paid another visit to the control centre to gather more feedback and talk to the controller on duty. Based on this visit, additional changes were made to the client; a day later these changes were available and the control centre was phoned to restart the client.

The client software was by that time stable and good enough to leave running for a few weeks. However, by the feedback provided by the users and analyses of the quality made by Fileradar, it then became clear that the capacity calibration algorithm that was used in this pilot was not able to cope with the complexity of the traffic congestion patterns in the Utrecht region. The prediction quality fell behind what both the developers (Fileradar) and the traffic controllers felt was a minimum for the predictions to be useful for actual decision support.

Having tested everything in the client as it was, the web client was taken offline by 20 December so possible improvements could be made to its interface as well as to the predictions. In the beginning of January a visit was paid to the control centre to further discuss the second half of the pilot with the heads of operation and to talk through some of the feature requests that were made by the operators.

During this stage it was also decided to invest time into the prediction algorithms, rather then testing the tool the second control centre. The tool was therefore not deployed into the regional control centre as was originally planned.

From the moment the client went offline, work continued on improving the user interface (see Table 3 in the previous section) and on removing a bug from the client software that caused the system on which the client ran to become very slow.

Apart from the work on the interface, much effort was put into improving the capacity calibration algorithm so as to have the model predict the more complex patterns. A detailed analysis was made of the situations in which the calibration algorithm failed, and possible improvements were identified. From December until March, some of these possible improvements were tried. Developing and testing such algorithms consumes time; therefore, by the time the first real improvements to the prediction quality started to occur, it was time to finish the pilot.

Nevertheless, by February 7 a new version of the client was ready and some of the new calibration routines were put in place on the server side. The client was then restarted in the control centre, where it ran until the 13th of March when the pilot was officially ended. A week later, an evaluation session was held in the traffic control centre to see what was learned and to obtain detailed views from TCC operators on the tool. This evaluation will be discussed in the next section.

5 Pilot Evaluation

5.1 Introduction

On 20 March 2013, an official evaluation was held at the national control centre. Attending the meeting were Ary Koot, head of operations, and Ron Kerkhof, team leader of the traffic controllers. Preceding the meeting, they both had informally talked to the individual traffic controllers to get an overview of their experience with the predictions. Below, an overview of the their feedback is given:

5.2 Good Start, Good Support

The control centre personnel experienced the start of the pilot as very positive:

- Short-term traffic predictions are a long desired feature in the control centre;
- Operators enjoyed the enthusiasm of Fileradar in this project, together with the speed with which feature requests were implemented in the client. This stands in contrast to the way things are done normally, through service desks and weeks or months of phone calls and emails before changes occur;
- Traffic controllers were very happy with the fact that they had a direct influence on the software itself; usually new software is designed by others and presented ‘as is’; and
- The resulting user interface was rated as of a good quality. Especially the dual map feature was experienced as user friendly and provided a good overview. The queue length plot was a long desired feature and was therefore received well by the users.

5.3 Other Developments During the Pilot

During the pilot the traffic controllers had to deal with more changes. A new system was deployed where live camera images became available to monitor the road network. Therefore, there was even more pressure then normal on the 9 screens that are available to a traffic controller. This caused in some cases for the Dante web client to be put behind other applications. Because the client was offline when the new version was developed, the pilot was pushed to the background.

5.4 Prediction Quality

As said before, the prediction quality was not sufficient, which together with the fact that other new tools appeared during the pilot caused the use of the client to degrade, until at the end of the pilot it became apparent that the tool wasn’t used anymore. However, had the prediction quality been sufficient, the control centre managers indicated that it would be very likely that the client would still be used. This indicates that every step needed to successfully host predictions in a traffic control centre was indeed taken, and all that needs to improve is the prediction quality itself.

5.5 Areas for Improvement

One thing that the managers of the traffic control centre indicated was that if they would do the pilot again, they would focus more on a few interested users that would work with the tool, rather than just leave the tool open for all users. Naturally some users are more interested in a new tool than others who are more sceptical by nature (or nurture). If a small group of users are more proactively involved into the pilot, then they are more prone to call in case of problems or new ideas because they feel more that the pilot is ‘theirs’ than in the current setup.

5.6 Use Cases for Predictions

At the question for what situations the predictions would be used, the managers responded that especially in the light of the economic crisis they see many potential applications, where amongst others the predictions support decisions regarding:

- Answering what-if questions to efficiently apply the right traffic management scenario; currently there is no good decision support system available to the control centre for answering ‘what-if’ questions and ex-ante evaluation of different traffic management scenarios;
- Replacing fixed hardware by mobile hardware that is deployed at locations where predictions indicate that congestion problems will arise; and
- Efficiently sending road inspector / traffic officers to the right locations where predictions indicate that problems will arise.

6 Conclusions: How to Successfully Apply Predictions in a Traffic Control Centre

6.1 Introduction

In this section the main conclusions are presented that can be drawn based on the experiences gained before and during the pilot. The main goal of the STEP project is to investigate how to successfully apply short-term predictions in a traffic control centre. The conclusions are presented from different perspectives on how to answer this question: user acceptance and trust, the user interface, technical deployment and the prediction quality.

6.2 User Acceptance and Trust

Lack of user acceptance is a significant impediment to the success of new information systems. Applying short-term predictions in a TCC is no exception to this rule.

- Traffic control centre personnel are by experience sceptical to new tools, because they often have seen so many of them.
- If you involve the personnel, managers and actual end users into the entire process from design to rollout you can increase the chance of a success.
- It is technically possible to build a decision support system based on predictions, such as automatic rerouting advice or advice on the optimal scenario to deploy based on the traffic predictions. However:
    - As the traffic controllers will be responsible, trust is an important factor in proactive decision support systems;
    - Users first need to trust the predictions themselves before they trust proactive advice of the system;
    - It is therefore advisable to first show the predictions ‘as is’ and to get the prediction quality up to a standard where the user fully trusts the predictions; and
    - Once that is achieved, more proactive decision support tools can be built upon those predictions.

6.3 Predictions

Although (or because) the predictions that were used in this pilot where not of the necessary quality that users required to trust and use the predictions, still valuable lessons have been learned:

- Reasonable accurate predictions on a horizon of at least 20 minutes are needed in order for the predictions to support decisions made by the controllers.
- The prediction model needs to be able to predict the effect of non-recurrent situations such as incidents or road works, because it is especially around those situations where traffic controllers really can make a difference by taking the optimal actions. Predictions only based on historic averages are therefore not very useful for traffic controllers.
- The accuracy of the predictions needs to be expressed in both the predicted length of the queues (determines whether spillback is correctly predicted) as well as the predicted speed of traffic (determines whether delays are correctly predicted). If both the length of future queues and the speed of future traffic are accurately predicted, the traffic controllers can make good decisions on which dynamic traffic management measures to apply, such as rerouting, ramp metering, et cetera.
- Timeliness is a critical factor, as generally traffic controllers need to respond swiftly to emerging traffic congestion. The predictions need to be available to the end users within

minutes.

6.4 User Interface

The user interface developed for this pilot was received well by the end users. Based on the brainstorms in the preparation phase and feedback during the pilot, the following conclusions can be drawn:

- Displaying the predictions on an interactive map provides an efficient way for the traffic controllers to consume the traffic predictions;
- Using a dual screen, where the actual situation is shown besides (an animation of) the predictions allows the traffic controllers to compare the predictions with the current situation, so that they can quickly estimate whether action needs to be undertaken; and
- The user interface needs to update frequently, because timeliness is such an important factor.

6.5 Technical Deployment

It proved to be a real challenge to deploy the pilot software in the traffic control centre because of a very strict IT-policy, something that is observed in other traffic control centres as well (such as in the UK, see WP4A). However, a successful deployment could take place because a Rich Web Client was used. The following conclusions can be drawn:

- A Rich Web Client works well in a traffic control centre because:
  - It is a good way of preventing a strict IT-policy to be an obstacle for deployment;
  - It ensures that the client always runs the latest version of the software;
  - Updates can therefore quickly be made, and an agile response to requests and bug reports keeps users engaged; and
  - It ensures that it can be run on any platform and network
- By having all communication go over port 80 and HTTP, the protocol that is used for web browsers, firewall issues are prevented; and
- Having good and fast technical support available is very important, not only because this keeps users engaged, but also because the software usually runs on critical systems with multiple applications that are vital to the functioning of the traffic control centre (and thus the functioning of the traffic system itself).

Appendix A: Analysis of Prediction Quality During the Pilot

In this appendix an analysis is made of the prediction quality of the system used during the pilot in the traffic control centre.

In order to quantitatively assess the quality of the predictions an error measure must be defined. Given the potential applications for traffic controllers two important features of the predictions must be considered: queue length and speeds.

Queue Length Accuracy

The length of the queue determines where congestion occurs. If the length of queues is predicted correctly, **spillback** is also predicted correctly. Based on this information a traffic controller can decide to intervene and try to shorten the future queue so as to prevent such spillback so that for example an important intersection remains in free flow conditions.

Queue lengths are analysed by investigating the so-called 'regime error' with this error defined as:

$$E_r(t) = \sum_{c \in C} \left| S_c(t) - S_m(t) \right| \quad (1)$$

Where c is a cell in the spatially discretized traffic network, C is the collection of all of these cells, $S_c(t)$ is state of the cell in the model at time t and $S_m(t)$ the measured state at the time t. The state S is defined as 0 if it is uncongested and 1 if it is congested. In other words, the regime error counts two types of errors:

1. If the measurement says that a cell is congested but the model is in free flow,

2. If the measurement says that a cell is in free flow but the model says it's congested.

Both types of errors have the same weight, i.e. both types of errors are considered equally important.

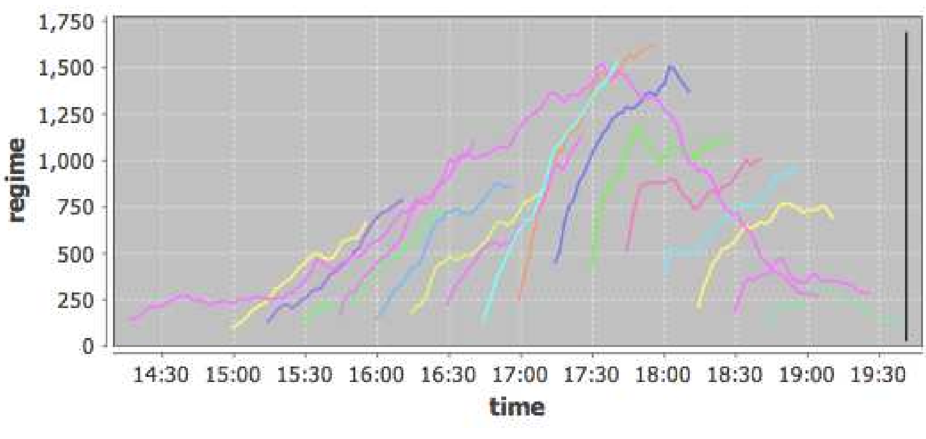


Figure A1 The 'regime' error for 16 different predictions (short lines) starting at 15-minute intervals, compared to the 'do-nothing prediction'(long, purple, mountain-shaped line).

In Figure A1A1 a graph is presented where the regime error is displayed for a number of predictions. First of all, a 'do-nothing prediction' was made for a regular evening peak period in the network that was used in the pilot (see Appendix B: The Study Area). In this 'do-nothing' the autocalibration algorithm was turned off, so that no congestion was predicted at all. This scenario forms a baseline against which the predictions can be compared. The long, purple continuous line that runs through the whole chart plots the regime error as a function of the time. As can be seen, at the height of the evening peak a total of 1500 cells are congested in reality (all error is of type 1 as the model predicts no congestion).

All other lines in Figure A1A1 are the result of predictions with the autocalibration algorithm turned on. The algorithm tried to replicate the traffic queues of the past hour in the prediction model and then predicted 1 hour ahead. Each line is the result of one run of the model. The first prediction training on the data between 14:00 and 15:00 and then predicts from 15:00 to 16:00; the next starts training at 14:15 and predicts from 15:15 to 16:15, and so on. As can be seen, the slope of all individual lines is upwards; this means that as soon as the predictions start the regime error starts to increase. Occasionally the regime error becomes larger than the 'do-nothing prediction', an indication that too much congestion is predicted.

The relatively steep slope of the regime errors is caused by a number of reasons:

- The emergence of new traffic queues in the future was not predicted correctly. Due to the importance of non-recurrent traffic queues to the traffic control personnel, most of the focus during this pilot was placed on predicting the evolution of non-recurrent queues that are by definition not expected to emerge in the future. Less effort was placed on predicting the more or less predictive emergence of recurrent queues, but this does influence the regime error too heavily.
- The model couldn't, at the time of the pilot, predict all types of traffic congestion. Especially the dynamics of so-called 'moving jams', which are queues of which both ends move in an upstream direction, are hard to predict. The emergence of such a moving jam is a stochastic event and thus hard to predict in itself. More importantly, moving jams interfere with congestion with a fixed bottleneck. This interference caused the auto-calibration algorithm to go wrong in some cases. This causes predictions to be either too optimistic or pessimistic depending on conditions. Both faults result in a larger regime error.

Speed Accuracy

A second important feature to look at when considering prediction accuracy is the predicted speeds. Speeds are important because they determine the travel times; get the speeds right and you get the travel times right. From travel times delays can be computed, and delays are an easy measure for traffic controllers to consider the gravity of certain traffic conditions. Furthermore, delays determine whether alternative routes are feasible or not.

The speed error used in this evaluation is defined as:

$$E_s(t) = \frac{1}{N_s(t)} \sum_{c \in C} \begin{cases} |v_c(t) - v_m(t)| & S_c(t) = 1 \vee S_m(t) = 1 \\ 0 & otherwise \end{cases} \quad (2)$$

$$N_s = \sum_{c \in C} \begin{cases} 1 & S_c(t) = 1 \vee S_m(t) = 1 \\ 0 & otherwise \end{cases} \quad (3)$$

where $v_c(t)$ is the speed in the model in cell c at time t, $v_m(t)$ is measured speed at cell c at time t and S_c and S_m were defined at equation (1). This function computes the average absolute difference of the speeds in all cells that are either congested in the model or in the measurement or both. This error is thus the sum of three types of errors:

- The model predicts congestion and the measurement is in free flow;
- The model predicts free flow and the measurement is in congestion; and
- Both are in congestion but the speed in the queue in the model is different than the speed in the queue in reality.

In Figure A2 A2 the speed error is visualized for the same predictions for which the regime error was displayed previously. The long purple line is again the 'do-nothing' error. It lies between 50 and 65 km/h, which means that the speed in a cell that is congested in reality is about 50 to 65 km/h lower than the speed limit. It can be seen that for most predictions the speed error remains below the 'do-nothing' error during the entire prediction period of 1 hour. Only at the end of the evening peak the speed error is larger; this has to do with too heavy congestion being predicted at the end.

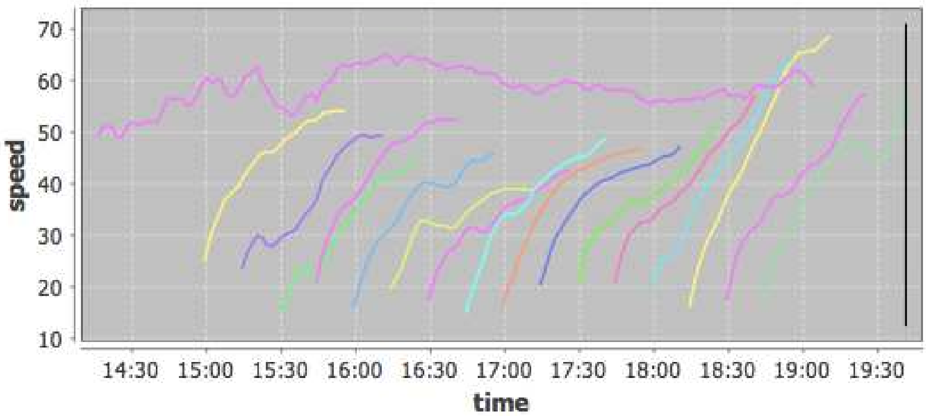


Figure A2 The 'speed' error for 16 different predictions (short lines) starting at 15-minute intervals, compared to the 'do-nothing prediction'(long purple line).

Conclusions

The two examples given in the previous analysis are typical for a regular day during the pilot in the area of consideration. While it is clear that predicting traffic is in most cases is better than doing nothing, there still remains much room for improvement. Most improvement can be expected if future queues that are expected (recurrent queues) are predicted, while additional improvement can be expected if the evolution of moving jams and their interference with 'bottleneck queues' are taken into account by the calibration algorithm.

Considering all said the error increases quickly as the prediction horizon increases. This limited the usability of these predictions for traffic controllers. In order for users to trust the predictions and for good decision support, the 'hockey stick' shaped error lines need to become more horizontal, both for the regime error and for the speed error. This is a valuable lesson for the STEP project, as it indicates that high quality predictors will only suffice and that much effort has to be placed on predicting traffic accurately under all possible circumstances.

Appendix B: The Study Area Covered by the Pilot

Figure B1 Overview of the network used fort his study (© Google Maps)

Figure B2 Over 1850 detectors are placed throughout the network (white dots)

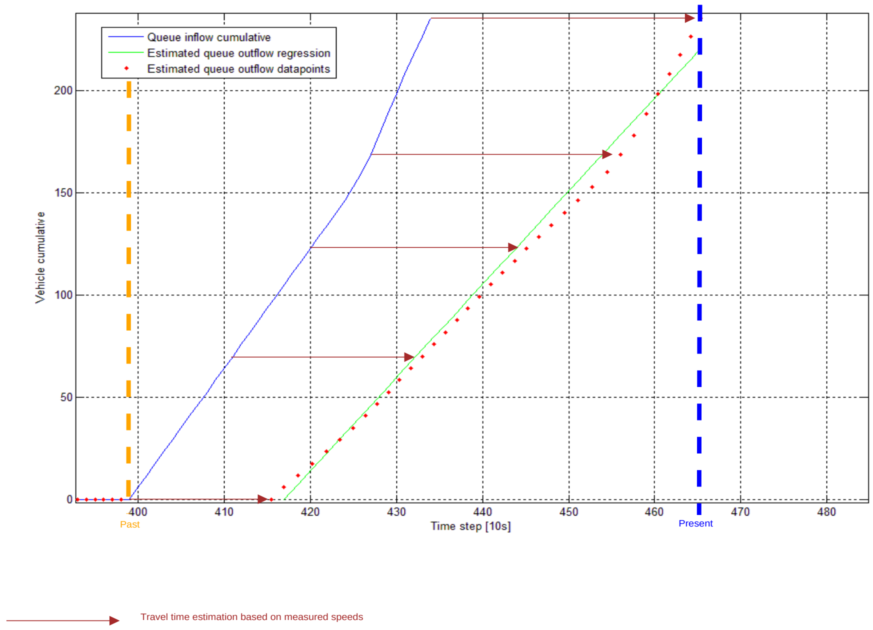
Figure and Figure B2 show the network that was used in this study.

The network consists of a total of 900km of network, with a total of 1856 double loop detectors (the white dots in Figure) evenly spread over all roads. The average spacing between detectors thus is about 500 m. These double loop detectors provide both speed and flow data on a minute-by-minute basis. It can be stated that the network is quite well equipped with detectors.

Appendix C: Automatic Capacity Estimation Algorithm

The bottleneck capacity and queue density estimation that is proposed can be outlined as follows: assuming that a bottleneck position is detected, at present time (*dashed blue line*), a function computes, based on current speed measurements, where the relative queue ends, in terms of link and cell id.

Given this knowledge, the algorithm then computes the travel time $\widetilde{TT}$ that, at present time, would take a vehicle to pass through the whole queue, from the first congested cell to the bottleneck cell.



This travel time $\widetilde{TT}$, with an additional 10% margin, is then used as the amount of time the correcting scheme will use to look into the past, therefore yielding the *orange dashed line* in the picture above.

The correcting scheme will, then, compute the cumulative number of vehicles $q_{in}(f_{\bar{m}})$ that enters the queue (*blue line*), beginning from the past and up until present, using flow information $f_{\bar{m}}(t)$ stored **by the model** in the past. This cumulative is then coupled with an estimate of travel times $tt(u_m)$, once again from the past up until the present, based on the historical measured speeds $u_m(t)$.

This yields values for the queue outflow cumulative q_{out} (*red dots*), which are simply the values of the inflow cumulative, shifted forward by the amount of travel time (*brown arrows*) needed to traverse the queue and the bottleneck.

Finally, by employing linear regression we approximate a slope (*green line*) for the outflow cumulative curve, which yields the capacity of the bottleneck cell, also employing the θ approach for ramp inflows and outflows depicted in your procedure to account for disturbances on the estimation process.

As for the density of the queue, the vertical difference between the inflow curve and the outflow curve at present time (not shown on the graph) yields the number of vehicles currently in queue:

$$n_q = q_{in}(present) - q_{out}(present)$$

This information can then yield the average density of vehicles in the queued cells, this can then be compared with the current density of the cells in order to adjust their capacity.

The idea described so far has been implemented into a small Proof of Concept, and adapted to that test case network I already used to test your own approach; the initial results appear very good.

Apart from this “core” idea, we have thought about several possible issues and outlined some strategies that should take care of them:

- When detecting the back of the queue, don’t just stop at the first free flow cell who is found, but allow some extra kilometres to be free flow before stopping the search process, this allows to take into account possible stop and go patterns that wouldn’t be detected otherwise;
- When moving backwards in time, an upper limit to the time $\widehat{TT}$ is set to avoid computational time overgrowth. This means that even if at present the length of the queue would need, for example, a total travel time of two hours, only a subset of cells close to the bottleneck will be used in the correcting scheme;
- When detecting the back of a queue $\hat{q}$, if a merging node is met, the back of the current queue is set at the most immediate downstream cell, and new bottlenecks $\{b_1, \ldots, b_N\}$ are activated on the N incoming arc(s) of the merging node. These new bottlenecks will then be corrected firsthand, so that then their outflows can be combined as inflow for the queue $\hat{q}$;
- When a previously detected bottleneck is removed, the capacity of the bottleneck cell is restored to its original value, but the adjustments made to the capacities of its upstream cells will not be immediately cleansed; instead they will be adapted dynamically whether the model predicts lower speeds than what is reported by measures, thus signifying that the capacity is underestimated