# Predicting Bus Arrival Times


## Report Contributors

Michael Chapwanya[*,†] Andrew Gordon[‡] Patrick McDowell[§]

## Study Group Contributors

Mark Burke[†] Karol Cwalina[¶] Fionn Fitzmaurice[‖] Natasha Hajanirina Lord[**]
Stephen McCarthy[††] John Miller[‡‡] John Morrison[‖] Mick O'Brien[*]
Doireann O'Kiely[*] Ieuan Stanley[§] Josh Tobin[§]

## Industry Representative

Brian Carrig, Dublin City Council (DCC), Ireland



**Abstract**

A Real-Time Passenger Information (RTPI) for bus and light rail is in the process of being rolled out on a nationwide basis by the National Transportation Agency (NTA). Dublin City Council are providing the technical implementation for services encompassing physical street signs, SMS messages, a public web site and a number of smart phone applications. This report summarises progress made towards the problem submitted by Dublin City Council at the $82^{nd}$ European Study Group with Industry organised by Mathematics Applications and Consortium for Science and Industry (MACSI) and held at the University of Limerick from the $26^{th}$ of June – $1^{st}$ of July 2011.



[*]Corresponding author. *Email address*:m.chapwanya@up.ac.za (Michael Chapwanya)
[†]Department of Mathematics & Applied Mathematics, University of Pretoria, South Africa
[‡]MACSI, Department of Mathematics & Statistics, University of Limerick, Ireland
[§]St Patrick's College Dublin, Ireland
[¶]University of Warsaw, Poland
[‖]School of Physics, Trinity College Dublin, Ireland
[**]Department of Mathematics & Statistics, University of Strathclyde, United Kingdom
[††]School of Mathematical Sciences, University College Cork, Ireland
[‡‡]Institute for Numerical Computations & Analysis, Ireland

# 1 Introduction

Although the system will eventually cater for Bus Eireann, private bus operators and the LUAS, currently the only user of the system is Dublin Bus. Prediction times for when a bus will arrive at a particular stop are generated by software designed by Init Systems for Dublin Bus and forwarded to Dublin City Council. This information is subject to certain constraints such as a look ahead window and a maximum number of buses to receive information for. Currently Dublin Bus has placed a limitation of 550 bus stop 'subscriptions' for the predictions their software generates. It is possible that their servers can be upgraded to handle a thousand subscriptions but it is uncertain and the original goal of obtaining four and a half thousand subscriptions looks unlikely by this method

Currently there are 80 physical street signs in place in Dublin and a website that provides predictions for 550 of the 4500 Dublin Bus bus stops. However, the noisiness and variability of prediction data has considerably slowed the progress of the roll-out. Thus DCC have requested the Study Group to find a method to accurately predict the arrival time of a bus at any stop (monitored or unmonitored).

Dublin City Council also receives all of the GPS location co-ordinates of every in-service bus in the Dublin Bus fleet, subject to the bandwidth constraints of the Dublin Bus private radio network. At peak times this amounts to almost 1100 buses. In practice we find that the bandwidth limitation amounts to a location update for each bus every 30 *sec*. The location is calculated using differential GPS and is said to be accurate to within 5 meters. Other information provided includes schedule deviation, whether a bus is at a stop or not and whether a bus considers itself to be in congestion or not.

Bus arrival time is important information for passengers but providing it is not an easy task. For example, bus arrival time at stops in urban networks are difficult to estimate because travel times on links, dwell times at stops, and delays both at signalized and non-signalized intersections fluctuate both spatially and temporally. A variety of prediction models for forecasting traffic states such as travel time and traffic flow have been developed over the years. The five most widely used models include historical data based models [*Williams and Hoel (2003)*], time series model [*Thomas et al. (2010)*], regression models [*Jeong (2004)*, *Ramakrishna et al. (2006)*], Kalman filtering model [*Chien et al. (2002)*, *Vanajakshi et al. (2009)*] and machine learning models [*Bin et al. (2006)*, *Yasdi (1999)*]. However, no single predictor has yet been developed that presented itself to be universally accepted as the best, and at all times, an effective traffic state forecasting model for real-time traffic operation.

# 2 Description of the Problem

The aim of this project is to provide all Public Transport users with high quality reliable information, on street and through Web and SMS, see Fig. 1. Hence Dublin City Council is seeking to answer two separate and distinct but related questions about the system. In particular, the Study Group was asked the following questions.

Assuming it is not possible to provide accurate predictions from just the location information stream provided (due to the close proximity of bus stops to one another within a city

and the infrequency of updates), what additional information would be required to deliver a system that can accurately predict the time that a particular bus will arrive at a particular stop? If this additional information were present, what level of complexity or processing constraints might be encountered for a system attempting to generate predictions for 1100 buses servicing four and a half thousand bus stops?

Secondly, the stated aim of the NTA for the project is to achieve 98% accuracy of predictions for the system. Assuming the location information to be always accurate, how could Dublin City Council approach verifying whether the predictions are sufficiently accurate? The current approach is to manually survey sites but this is both time consuming and expensive.

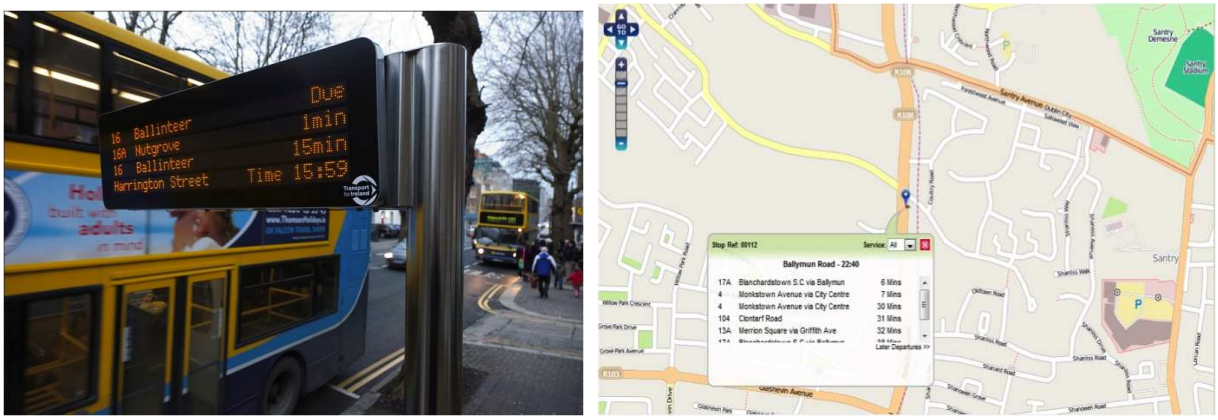


Figure 1: Dublin Bus information: street displays or on the web (www.rtpi.ie).

Currently there are 80 displays in operation and an additional significant number of sites have been identified. However, the current approach has revealed several problems including

- buses arriving at the stop without being on the sign (ghost buses),
- predictions counting down without a bus arriving,
- or errors to do with missing data.

To minimize the errors associated with the predictions, specific flags are now used to indicated the accuracy of the predictions. In addition, cameras are also used to record display signs and bus arrivals.

The objective of the Study Group was to develop a dynamic model that can provide accurate prediction for the Estimated Time of Arrival of a bus at a given bus stop using the provided global positioning system (GPS) data and/or the observed travel time data. In particular, the Group aimed to use the current available data to provide a model that can efficiently predict the arrival times.

# 3 Approach I: Average Travel-time Model

The Study Group investigated a model of predicting the arrival time based on the average time taken by several buses on the same route. The idea was to use data from buses with

same inbound/outbound departure times over a certain period of time. The group based their analysis on data for bus number 4 on selected Tuesdays within the period from June - July 2011 with the aim of extending this to cover all the other routes. Five outbound 4:00 PM – buses on route 4 were selected. The GPS data indicated that the route is approximately 23 $Km$ and it will take each bus an average of 78 $mins$ for a single outbound trip. The average travel time of the five buses was calculated using a least square approach.

## 3.1 Prediction Based on Average Time

Here we predicted the arrival time of the 4:00 PM bus on route 4 using the average time taken by the selected 5 buses. We will refer to this as our simple model. The algorithm is given by

$$t_p(k+1) = t_p(k) + \Delta t_{\text{av}}(k+1), \tag{1}$$

where $t_p(k)$ denote the total predicted time to arrive at stop $k$ and $\Delta t_{\text{av}}$ is the average time taken by the selected 5 buses to travel between stops $k$ and $k+1$. A comparison of the observed arrival time and the predicted time is shown in Fig. 2.

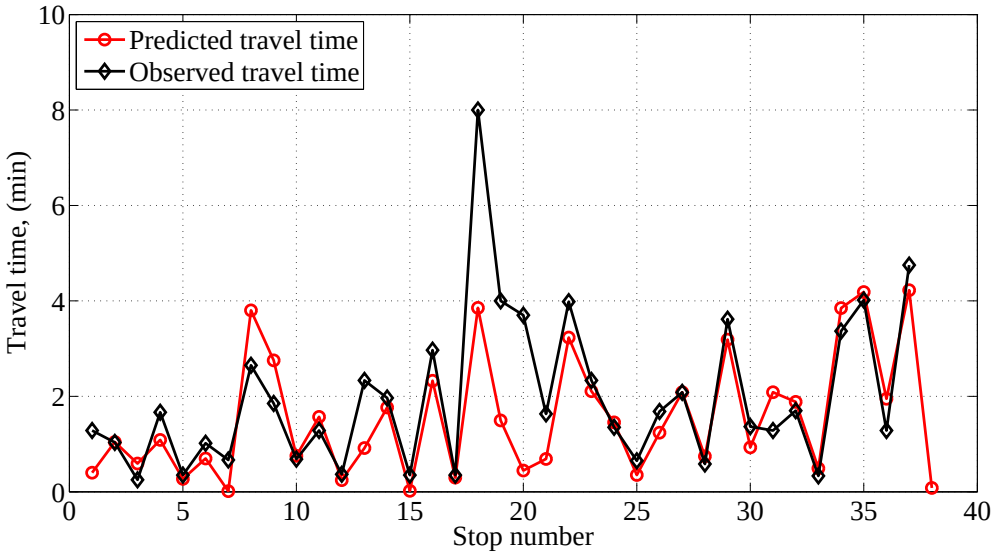


Figure 2: A comparison of the arrival times for the simple model.

In general, we observe a good fit between the predicted time and the actual time. However, in some cases the predicted time is lower than the observed time with errors of up to 5 $mins$. See the error histogram in Fig. 5. This may not be the most efficient way since travel times are updated only once when the bus leaves the first stop. We are likely to see 'ghost' buses or 'no show' in this setup. Next we refine the predictions by using the observed times at the previous stop other than the last prediction, i.e.,

$$t_p(k+1) = t_{\text{a}}(k) + \Delta t_{\text{av}}(k+1), \tag{2}$$

where $t_{\text{a}}(k)$ is the observed time recorded at stop $k$. We will refer to this model as the modified simple model. This model requires that the updates be done every time the bus

clears a bus stop. In addition, predictions can be done at any number of stops from the current position. The simulations are shown in Fig. 3.

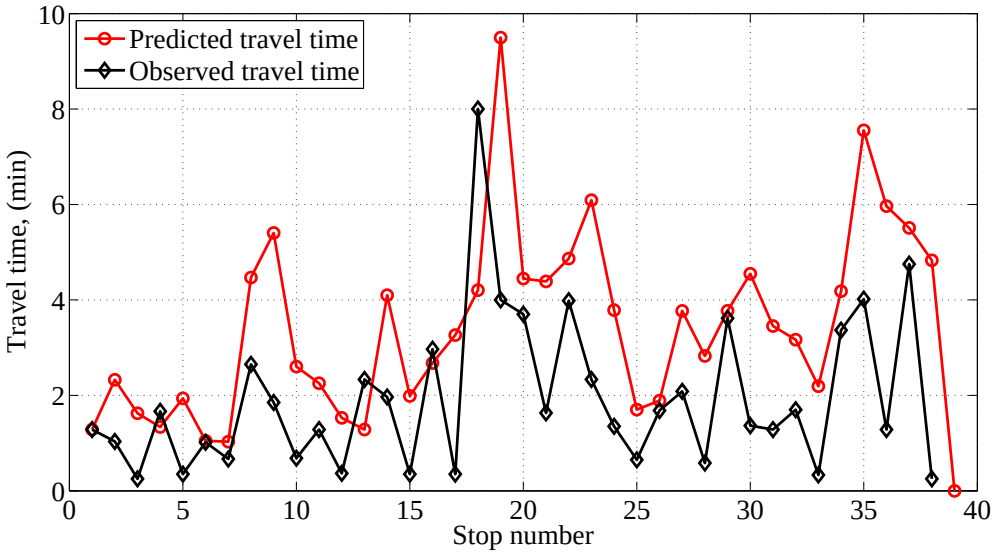


Figure 3: A comparison of the arrival times for the modified simple model.

## 3.2 Prediction Based on Kalman Algorithm

In another approach, we follow the work of *Vanajakshi et al. (2009)* who used an algorithm based on the Kalman filtering technique. In their paper, the arrival time for a particular bus was predicted using GPS location of the current bus and the times predicted by two probe vehicles on the same route. Our approach is similar to *Vanajakshi et al. (2009)*, but here we choose the data for the two 'probe' vehicles from B1, the average of the selected 5 buses and B2, the previous bus - in this case the 3:45 PM bus. The travel time for each $k^{th}$ subsection was estimated from

$$\Delta t_{\mathrm{p}}(k+1) = a(k)\Delta t_{\mathrm{p}}(k) + w(k),$$

where $a(k)$ is a parameter associated with bus B1 and $w(k)$ is the disturbance associated with the subsection.

For completeness, we outline the steps in the algorithm as follows

1. We divide the route into $n$ points with each point representing a bus stop.

2. The travel time from B1 was used to compute $a(k)$ via

$$a(k) = \frac{\Delta t_{\mathrm{B1}}(k+1)}{\Delta t_{\mathrm{B1}}(k)}, \quad k = 1, \cdots, n-1,$$

where $\Delta t_{\mathrm{B1}}$ is the travel time of bus B1 in each $k$ subsection.

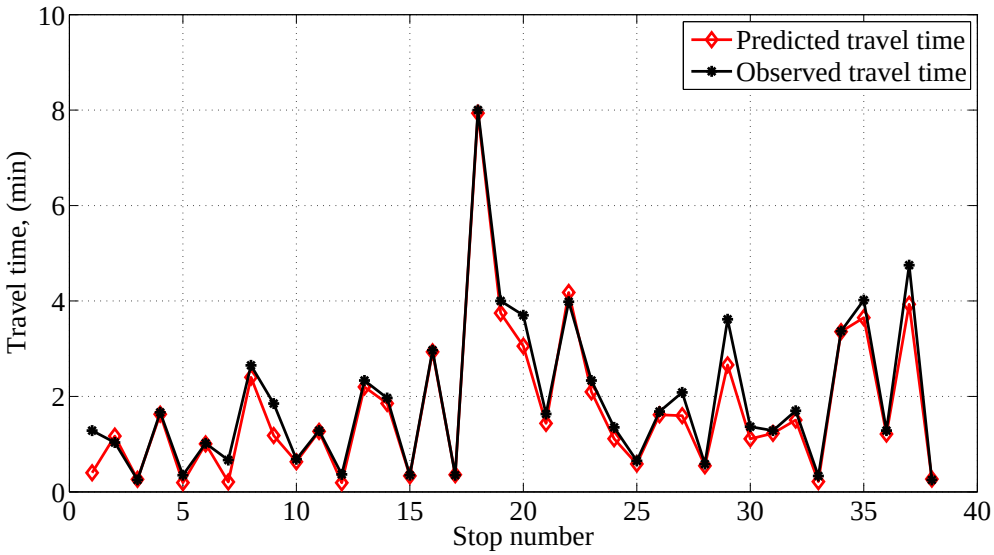


Figure 4: Comparison of predicted time and observed time using the algorithm from [5].

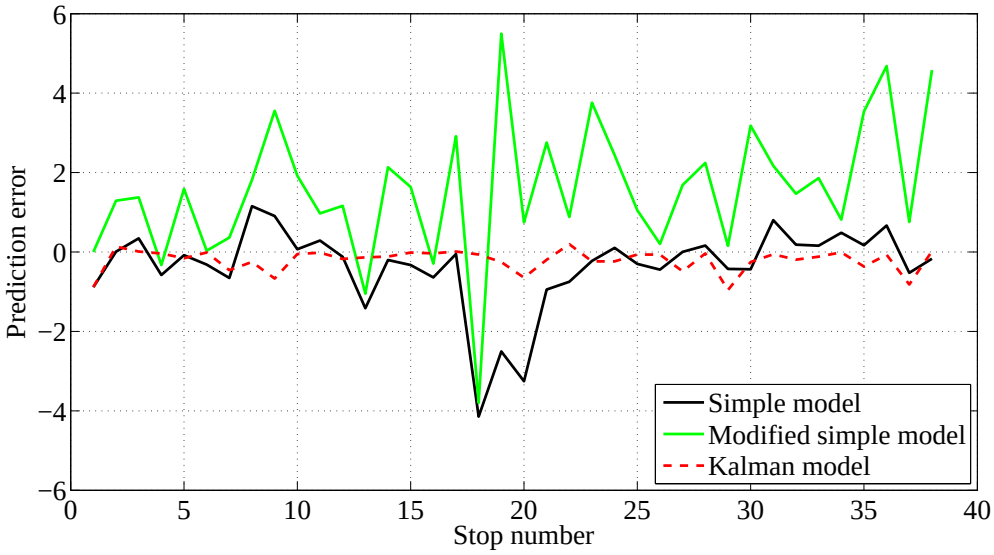


Figure 5: Comparison of the errors in the three models.

3. The Kalman algorithm is a predictor corrector method, i.e.,

$$
\begin{aligned}
\text{priori estimate} \quad & \Delta t_{\mathrm{p}}^{-}(k+1) = a(k)\Delta t_{\mathrm{p}}^{+}(k) \\
\text{priori error variance} \quad & P^{-}(k+1) = a(k)P^{+}(k)a(k) + Q(k) \\
\text{Kalman gain} \quad & K(k+1) = P^{-}(k+1)[P^{-}(k+1) + R(k+1)]^{-1} \\
\text{posteriori travel time} \quad & \Delta t_{\mathrm{p}}^{+}(k+1) = \Delta t_{\mathrm{p}}^{-}(k+1) + K(k+1)[\Delta t_{\mathrm{B2}}(k+1) - \Delta t_{\mathrm{p}}^{-}(k+1)] \\
\text{posteriori error variance} \quad & P^{+}(k+1) = [I - K(k+1)]P^{-}(k+1).
\end{aligned}
$$

Here the superscripts '–' denotes the a priori estimate and '+' the posteriori estimate. The prediction together with the observed travel times are given in Fig. 4. Note, the current

model differs from *Vanajakshi et al. (2009)* in several ways. The current model uses data from the average of previous (weeks) buses and the last bus while *Vanajakshi et al. (2009)* uses data from two previous vehicles. In addition, no GPS data is required in the current model.

In Fig. 5 we compare the efficiency of the three presented models by plotting an error histogram for each model. The error in the predictions is given in *mins* and is calculated from

$$\text{error} = \Delta t_p - \Delta t_a.$$

We observe that while the simple model under-predicts the arrival times, the modified simple model over-predicts the arrival times with the Kalman model falling in between. In general, the Kalman based model significantly outperforms the other two models. In the next section we present a model which considers all the buses in operation as a single process.

# 4 Approach II: The Polling-time Model

This model relies on the polling time of the reporting system Dublin Bus currently use. This polling time is unitary for all operational busses at any particular time of day, so in effect it reduces a substantial number of virtual 'threads' (i.e., systemic or parallel processes as in Approach I) down to one. For example, if there were 1100 busses in operation, 1100 process calculations would have to be made in predicting average times for all busses. In this alternative model, a 'number of polls' variable is simply incremented (one for each bus) subsequent to a report back. For each bus, a new 'element' of this variable is created as it goes from one stage (the 'measurement' between two subsequent stops), and the next. The model simply accumulates the difference between the de-facto or expected number of polls for a particular stage and the actual number of polls for that stage. This difference for each stage called $\varepsilon_i$, can be negative, null, or positive. That is, the model sums-up these $\varepsilon_i$ for $r$ retrospective stops and projects $m$ stops ahead of the last stop visited. Process-wise, 1100 threads are reduced to one.

## 4.1 The Model

In predicting the time of arrival for the next stop, the equation is as follows:

$$T_n = \Delta t_{n-r-1} + t_n + t_p \left(\frac{r+1}{r}\right) \sum_{i=n-r}^{n-1} \varepsilon_i, \quad r \neq 0, \tag{3}$$

where $T_n$ is the estimated arrival time at stop $n$, $\Delta t_{n-r-1}$ is the difference in the actual arrival time at stop $n-r-1$ and the de-facto arrival time at this stop, $t_n$ is the de-facto or expected arrival time at stop $n$, $t_p$ is the polling time (time between two subsequent/adjacent polls), $r$ is number of preceding or retrospective stops we are examining with respect to 'error' in the number of polls, and $\varepsilon_i$ is the difference in the actual number of polls and the de-facto number of polls for a particular stage $i$ (between stop $i-1$ and stop $i$).

So,

$$\Delta t_{n-r-1} = T^*_{n-r-1} - t_{n-r-1}, \quad \text{and} \quad \varepsilon_i = p^*_i - p_i,$$

where $T^*_{n-r-1}$ is the actual arrival time at stop $n-r-1$, $p^*_i$ is the actual number of polls for stage $i$ and $p_i$ is the de-facto number of polls for stage $i$ and can be calculated as follows

$$p_i = \frac{t_i - t_{i-1}}{p'},$$

where $t_i - t_{i-1}$ is the de-facto time between stops $i-1$ and $i$ and $p'$ is the polling rate. For example, if $t_i - t_{i-1}$ were expressed in minutes, and each bus is polled every 30 *sec*, then $p' = 0.5$

Of course, if we are considering the block numeric difference in the number of polls between stop $n-r$ and $n-1$, $\varepsilon$, the equation is as follows:

$$T_n = \Delta t_{n-r-1} + t_n + t_p \left(\frac{r+1}{r}\right) \varepsilon, \quad r \neq 0.$$

When estimating the time for m stops ahead of the n-1 stop, the equation becomes

$$T_{n+m-1} = \Delta t_{n-r-1} + t_{n+m-1} + t_p \left(\frac{r+m}{r}\right) \sum_{i=n-r}^{n-1} \varepsilon_i, \quad r \neq 0,$$

and this should be the general equation in the model, the fore-mentioned equations do not have to be used.

We can see that the model is centered around $t_p$ the polling time, and there is really no influential change in this for any conceivable sequence of iterations. However, if there were an abrupt change, the model can handle this.

Also, the term $\left(\frac{r+m}{r}\right)$ gives quite a smooth or flowing prediction or update for each $m$ stop ahead. This also portrays effectiveness should there be an abrupt change in the time it took the bus to get to (a) particular stop(s) between $n-r$ and $n-1$ stops.

In effect, the model considers a combination of timing between each successive stop and the polling times. The timing between each stop is exclusively expressed by the platform $\Delta t_{n-r-1}$ (no summation is required), from which to launch the more accurate polling time considerations, and of course all considerations are determined by $r$. As the polling time is usually less than the stage time, greater accuracy is ensured when considering polling times, and this accuracy can be fine-tuned by the consideration of $r$. Dublin City Council can predetermine r to optimise accuracy, and of course $r$ can vary depending on time of day and traffic conditions, and also special occasions such as St. Patrick's day parades, etc. Obviously, as the bus is traveling to the first 'few' stops on its route, $r$ would increment progressively to a predetermined value. However, predicting a considerable number of stops ahead based on a relatively small number of initial or retrospective stops is not advisable. There is only one summation in this model, which effectively contributes to processing resources, simplification, and testing.

This model is based on retrospection up to stop $n-1$, the last stop. If abruptness occurs subsequent to the last stop, no elegant model can effectively come up with accurate predictions. However, a recursive approach can be used with this model – the formula function calling itself, i.e., the segment between where the bus is currently at and the last bus-stop is broken into a number of sub-stages, which of course depends on the degree of

recursion we are currently in. Accuracy increases at each degree, and is related to the number of polls since the last stop. These sub-stages are not pre-determined or pre-fixed, but are dynamic and related to reported GPS data.

Obviously, there is no need for recursion if no abruptness occurred since the last stop. Abruptness can become apparent if the bus has not yet reached its desired stop after a considerable number of polls since the last stop. This is very useful in raising an alert.

There are additional benefits when considering alerts.

1. A high-degree of confidence in resolving the 'cleardown' problem. An alert can trigger a positional check, and if a bus is deemed to have already passed the next predicted stop, or a series of stops that have not been subject to cleardowns, cleardowns can be evoked. The time to the actual next stop, $n$, can now be estimated as follows:

$$T_n = t_l + \left(\frac{t_x - t_l}{t_e - t_l}\right)(t_n - t_l),$$

where $t_l$ is the actual time of arrival at the last stop to be registered by the system, $t_x$ is the actual time of the positional check, $t_e$ is the expected or de-facto time for the bus to be at the location of the positional check, and can be calculated as follows:

$$t_e = t_l + \frac{d_{l,x}}{d_{l,n}}(t_n - t_l), \quad d_{l,n} \neq 0,$$

where $d_{l,x}$ is the distance from stop 1 to the location of the positional check and $d_{l,n}$ is the distance from stop 1 to stop $n$.

No doubt the present system uses software to calculate these distances.

Basically this equation is just a linear equation, the graph of which is suspended on two axis, expected/de-facto times ($x-$axis) and actual times ($y-$axis), $t_l$ being the origin and we are projecting up from $t_n$ and across to get $T_n$

As a matter of fact, this linear equation can be used as a coarse alternative to the 'polling time' model entirely, we are just basing our estimation on two known points, and $t_l$ does not have to be the last stop - just a prior positional check, the appropriate selection of which is important for optimisation.

However, in regard to missed clear-downs, each $\varepsilon_i$ for these stops can be estimated by projecting back/leftwards on the linear equation to determine an estimation for the actual time of arrival for these stops, and therefore the 'polling time' model can be re-implemented to determine greater accuracy and continuity in the process system/algorithm. This of course assumes that the reason why an alert was triggered was due to clear-down skips, and not due to abruptness in which case the recursion suggestion may be viable.

Another viable suggestion regarding abruptness would be to use the 'least square fittings' method on a series of positional checks during the abruptness to obtain a line for projecting forward, similar to the linear equation fix.

As abruptness is normally severe, estimations based on the 'polling time' model are based on stops prior to the abruptness and this model would have to be suspended if

the platform for predictions is from the last stop. After abruptness, the 'polling time' model would be re-engaged and $r$ re-initialised.

2. An alert can alert a radio operator in Dublin Bus who can ask the driver for an update.

3. An alert can focus DCC's traffic monitoring system to a particular location.

Other considerations would be to incorporate output from DCC's traffic management system into the new model, i.e. iterations of cycle-times can be added or subtracted for each iterative stage (between adjacent stops), and incorporated into formula.

If quick-fire or global error analysis or results are required at a meeting for example, the least-squares fitting (aka, best-fit line) approach can be used. The de-facto line would be plotted on a $x - y$ graph for a particular route or segment in a route, and a scattering of points from actual or accumulated information would be plotted alongside. The least-squares method would be used to get the best straight line which suits these points. The 'quick-fire' or 'meeting-friendly' error (numerically and visually [comparisons can be made with other graphs, for different routes or times, say]) is proportional to the angle between the two lines on the graph, it is actually proportional to the tan of the angle. So if the two lines were the same, the angle and the error would be zero.

## 4.2 Simulations

This model was tested with data from the Southbound 16:45 Dublin Bus on Route No. 4. This particular time was chosen because it is immediately prior to the Dublin rush-hour, and thus it was felt that most scenarios would be naturally included in the testing. Initially the polling-time model envisaged a de-facto or standard time-table or stop-schedule for each planned journey. Such schedules can be evolved stepwise through time to determine the most appropriate and accurate schedule for a particular journey, i.e., yearly or seasonal evaluations could be made.

No scheduling data was at-hand at the time of testing, however such a requirement could be considered to be somewhat redundant as it would suppress the need to evolve a de-facto timetable during the limited time of testing. Data for five adjacent Tuesdays spanning June and July 2011 was available, and as such an initial timetable could be developed. However, data for three Tuesdays in June was exclusively used as it was determined that inconsistencies in bus-stop identifiers would be minimised. This was not seen as an impedance as randomness was introduced into the testing during the later stages.

In determining a de-facto timetable based on actual journey information, three approaches were taken.

1. A timetable was built based on the average intervals between each successive stop. These intervals were then added to the base time of 16:45.

2. A timetable was calculated based on averaging each time of day a bus was at a particular stop on the route.

3. The third approach is the most radical and is based on a synthesis of the previous two. It is suggested that Dublin Bus relegate timetable information and adopt de-facto interval data in preference. As there is a greater likelihood of inconsistency in the times a particular bus starts its journey, i.e., leaves a terminus, due to human and systematic error, building a specific or unique timetable for each journey based on de-facto interval information is a suggested strategy. Such a timetable can be build when the bus has left its second stop on the journey. In this sense, it is inescapable that the bus is in progression. Also, this approach aids Dublin Bus in their requirement to supply dynamic and live web-based information to their customers. In relation to technological advancement, using periodic polling-times is an effective way to update customers, whether via web-page or RSS feeds to hand-held devices. If such a consideration has never been previously envisaged, it may be an advancement for Dublin Bus to progressively update subscribed customers of when the next bus is due at a particular stop, and to give him/her prior notification regarding the last bus. To return to the interval approach, should any abruptness occur during a particular journey, interval based data can quickly be used in recovery once such abruptness has terminated. Furthermore, de-facto interval information fits into the wider picture when route diversions occur. In relation to the polling-time model, it was determined that a 'mathematical conceptual dimension' is reduced with this approach (in comparison to the first two approaches), thereby contributing to its effectiveness.

Testing for all three approaches was undertaking. In each case, various values were given to $r$ and $m$ in the model. i.e., predicting the arrival time $m$ stops ahead based on polling counts of $r$ prior or retrospective stops. The polling time between 16:00 and 18:00 was found to be very consistent at 21 *sec*. Testing is by no means complete, however an advantage of the polling-time model is that values for $r$ and $m$ can be evolved to best-fit particular criteria. In conjunction with the interval approach, either by synthesis or opting in and out, an appropriate system can be imaged.

#### 4.2.1 Test 1: Timetable determined by average intervals added to an accepted base-time

From Fig. 6, it can be determined that for small values of $r$, successive predictions to a distant pre-determined stop are more likely to be erratic. This is unacceptable when customers are to be progressively updated. For large values or $r$, the erratic nature diminishes and updates are more congenial. For this particular data-set, the differences between the current prediction and the previous prediction were summed. When the summation exceeded 45 *sec*, the summation was reinitialised. In the case of $r = 5$, 6 initialisations occurred. For $r = 20$, there were no initialisations. It can also be seen that for $r = 20$, the initial error is the smallest, i.e., 90 *sec*. When the bus is approximately 8 stops away from the object stop, predictions for $r = 15$ are the more accurate, and are less than 45 *sec* in error. The actual travel time from the start of predictions to the desired stop is slightly less than 10 *mins*.

Fig. 7 displays predictions for a fixed value of $m$, i.e., a predetermined number of stops ahead from the start of the journey. Predicting to a particular stop does not occur here, but rather to a sequence of stops determined by $m$. As expected, predictions are more accurate

for shorter distances ahead. However, an appropriate value of $r$, determined by route, traffic conditions, and time of day can be determined for optimisation. As this figure expresses predictions for the entire route, the value of $r$ had to be counted up from 1 at the start of the route to its predetermined state, this explains the 'tails' for $r = 1$ on the extreme left of each graph. The plot settles at $r = 3$.

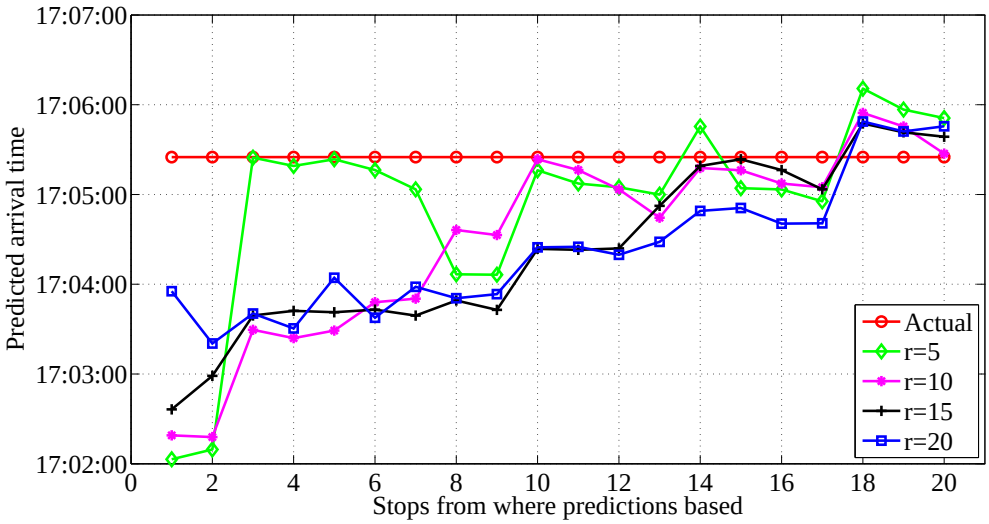


Figure 6: Successive bus-stop-wise predictions to a specific stop initially twenty stops ahead (m=20).

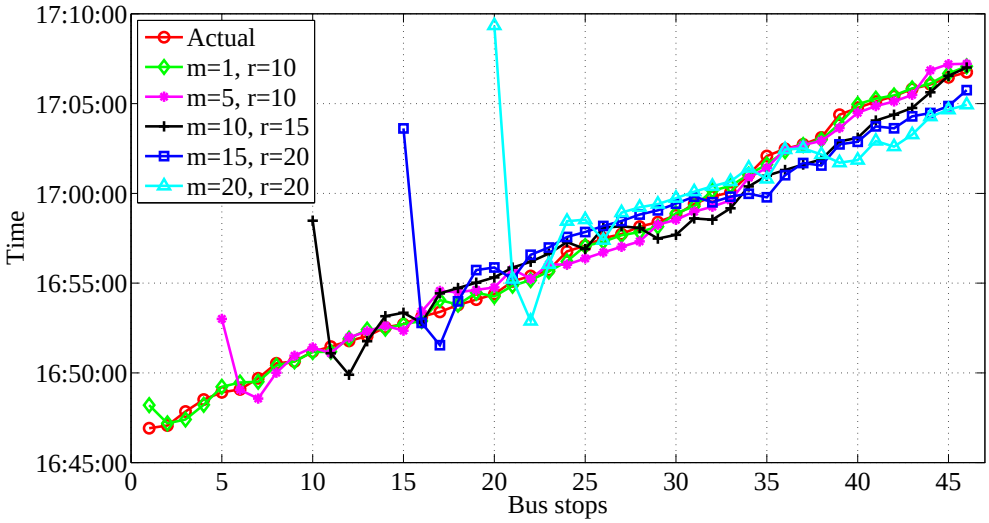


Figure 7: Predictions based on predetermined values for $r$ and $m$.

### 4.2.2 Test 2: Timetable built exclusively on averaging arrival times for each stop

In this approach, a de-facto timetable is exclusively built by determining the average time for each stop on the route. No intervals are consciously added to a predetermined base time. Although mathematically there is effectively no difference between the timetable determined in 'Test 1' and the one in this approach as the intervals are the same, by chance the same bus journey data used in 'Test 1' produces a slightly more erratic graph in 'Test 2'. This underlines the rationale of applying de-facto intervals to determine a busses unique schedule or timetable once it has started its journey.

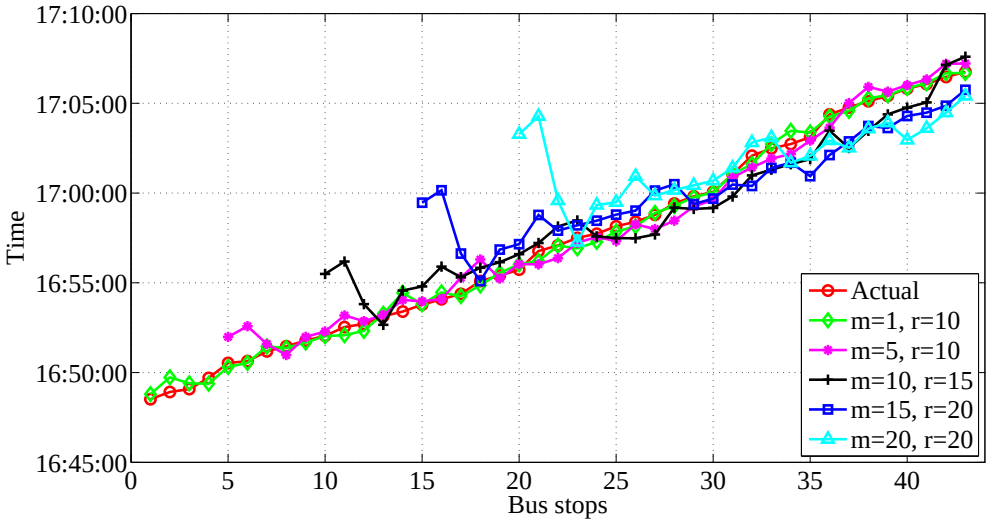


Figure 8: Predictions based on predetermined values for $r$ and $m$ (Average arrival times approach in timetable).

### 4.2.3 Test 3: De facto Interval Approach

This approach is based on creating a new timetable for any particular bus each time it has been undoubtedly established that it has started its route. The timetable is built from existing de-facto bus-stop time intervals for that particular planned route. A reservoir of de facto intervals could be held for any planned route, and the appropriate one applied to build a timetable. The range of timetables in the reservoir may be categorised under weather conditions, holiday season, specific public holidays, time of year, etc.

As with the first two approaches, the same bus data was applied to the initial testing of this approach. A graph of which is displayed below, Fig. 9. In this approach and in 'Test 2', the absolute or positive value was summed for the difference between the expected and actual arrival time of every stop on the route. The summation amounted to 29 *mins* 8 *sec* in 'Test 2', while with this approach the sum is 25 *mins* 43 *sec*. When the difference in these sums with respect to each sum is considered, a significance can be understood in relation to errors in stop arrival times.

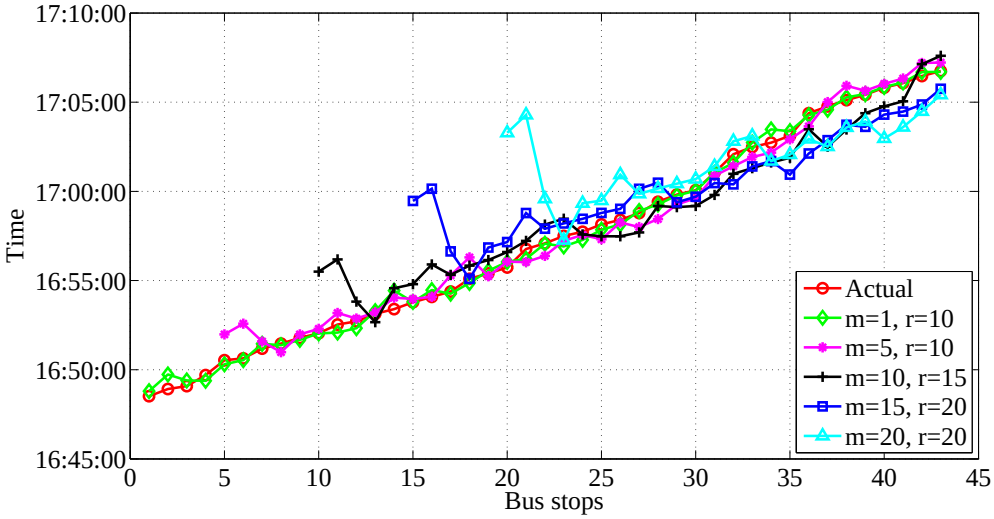


Figure 9: Predictions based on predetermined values for r and m (De facto intervals).

### 4.2.4 Randomising

Random delays ranging between one and 70 *sec* were accumulated to the expected arrival times at each stop. Given that the average de-facto interval between any two adjacent stops is 24 *sec*, such delays and their erratic nature are very unlikely. However, this simulation can be suggestive of snow or icy conditions, or were there are large numbers of people embarking and disembarking.

Fig. 10 below is a display of this unlikely scenario. It can see that the model works well for $m = 15$, i.e. predicting fifteen stops ahead, once the consideration of $r$ retrospective stops has reached or exceed 10. Considering that the journey time has more than doubled, the model is favorable to this situation.

Fig. 11 displays the output when an additional 3 *mins* was added to the 'actual arrival time' of 'Stop 15'. Predicting 10 stops ahead seems to buffer this anomaly.

Fig. 12 highlights data when three 3 minute delays are added to a series of arrival times. As well as selecting appropriate values for $r$ and $m$, strategies discussed in the section on 'abruptness' may need to be considered.

Fig. 13 shows a profile of progressively predicting to a particular stop on the data that was used in Fig. 12. The value $r = 20$ seems to give the smoothest transition.

## 4.3 Further Consideration

This model, as with all viable models, is based on a de-facto Dublin Bus timetable/interval data, which determines expected arrival times at all stops for all busses. Obviously, progressively this information would be refined, as is the case with all organic systems. Perhaps when publishing hard-copy or web-page standardised timetables, a number of standardised timetables could be used throughout the year, depending on the 'season'. i.e., school holiday times, characteristic weather conditions, day-light hours, etc. Furthermore, say for example four timetables were used, the expected arrival times at a particular stop can be posted on

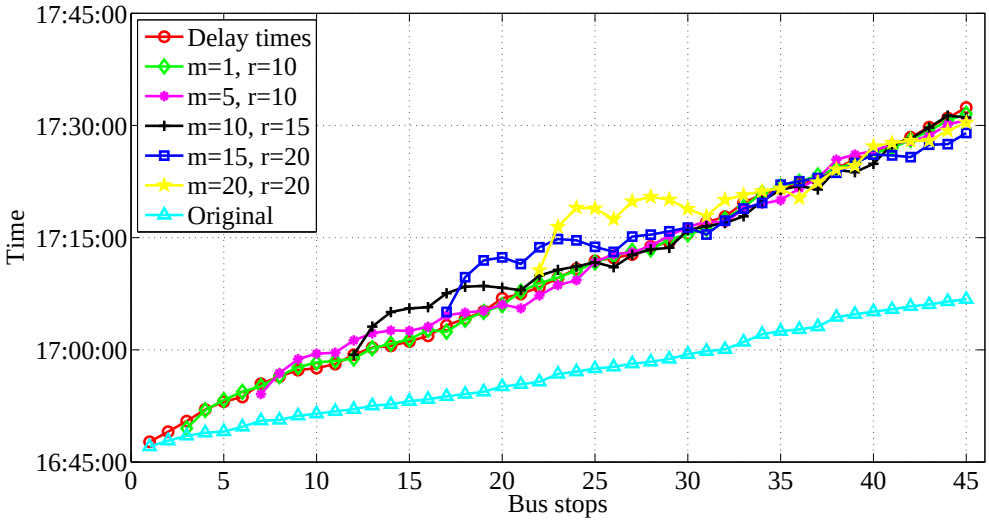


Figure 10: Random delays of between 1 and 70 *sec* accumulatively added to 'actual arrival time' of each stop.

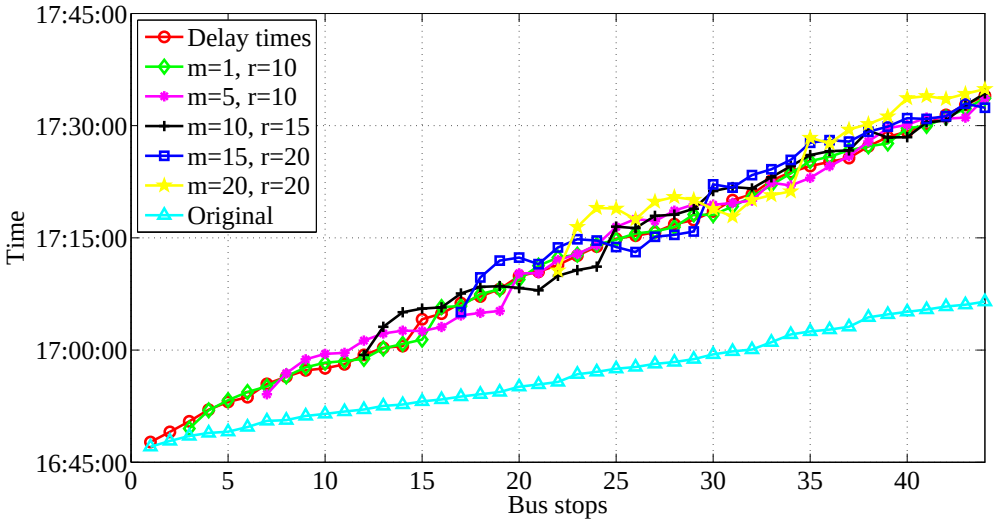


Figure 11: Additional 3 *mins* added to arrival time at stop 15.

that bus-stop, this is globally generic. However Dublin Bus can use the advertising edge or gimmick to impress and socialise passengers that they factor seasonal conditions into their timetables, and that the set of four timetables do not change for a 'block' number of years. This set of four timetables for a particular bus route can be posted on the web, and for stops where there are a limited number of 'bus-routes' stopping (as in most suburban areas), the 'set of four' timetable can be posted without need to change from season to season, or year to year. Also, when a timetable is being updated for a particular season based on retrospective experience, there is a nine month to one year buffer in which to do so.

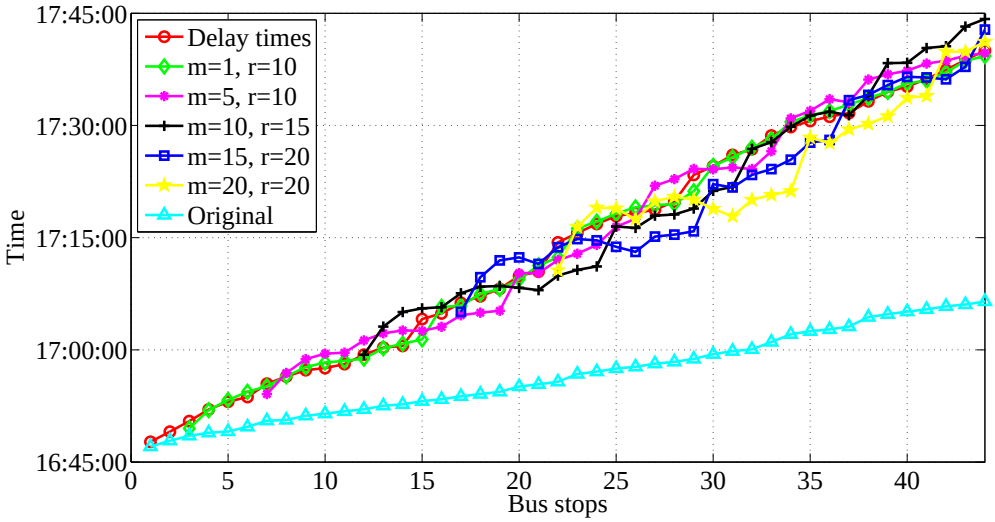


Figure 12: Delays at Stops 15, 22, and 29 are 3:10, 3:34, and 3:10.

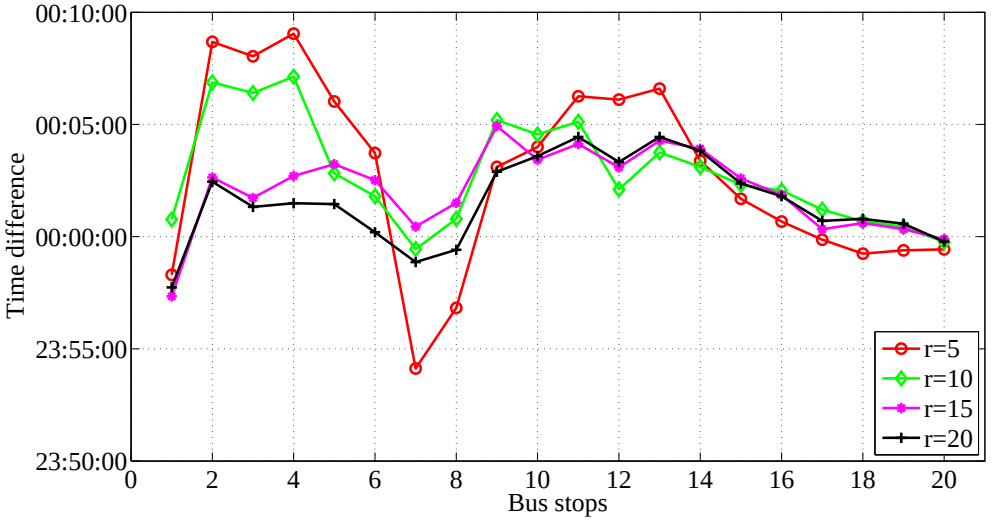


Figure 13: Progressively predicting to a particular stop initially 20 stops ahead. Stop No. 19 is the initial stop in this test, and its 'profile' can be seen in Fig. 12.

# 5 Conclusions and Future Work

This report outlines two approaches in predicting bus arrival times for DCC. In particular, we aimed to use the currently available data and minimise errors associated with the current predictions. In section 3 we reviewed some models in the literature and tested them using the current data. The key result of this section being that the classical Kalman algorithm outperforms the other presented sub-models. In section 4 we presented a polling-time model which reduces the need for parallel process for each bus in operation to just one process. The model was validated by comparison with existing data from Dublin Bus network.

Future work will link models to the average number of passengers waiting for a specific

bus, change in conditions, e.g., accident, demand surge, road works, etc.

## Acknowledgements

All contributors would like to thank Brian Carrig from Dublin City Council for introducing the problem and answering questions during the entire week.

We acknowledge the support of the Mathematics Applications Consortium for Science and Industry (www.macsi.ul.ie) funded by the Science Foundation Ireland mathematics initiative grant 06/MI/005.

# References

[1] B. Williams and L. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering (ASCE)*, 129(6):664–672.

[2] S. Chien, Y. Ding, and C. Wei. Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering (ASCE)*, 128(5):429–438.

[3] Y. Bin, Y. Zhongzhen, and Y. Baozhen. Bus arrival time prediction using support vector machines. *Journal of Intelligent Transportation Systems*, 10(4):151–158, 2006.

[4] R. Yasdi. Prediction of road traffic using a neural network approach. *Neural Computing & Applications*, 8(2):135–142, 1999.

[5] L. Vanajakshi, S. Subramania, and R. Sivanandan. Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses. *IET. Intell. Transp. Syst.*, 3(1):1–9, 2009.