

Predicting Transit Vehicle Arrival Times

Brendan Kidwell

GeoGraphics Laboratory

Bridgewater State College, Bridgewater, MA, USA

10 August 2001

Abstract

This paper presents a computer algorithm for predicting the arrival times of public fixed-route buses at their stops, based on real-time observations of the vehicles' geographic positions. The algorithm works by dividing each route into zones, and recording the time it takes to pass through each zone. Predictions are based on the most recent observation of a bus passing through each zone. The output of the algorithm can be displayed publicly to users of the system as the Estimated Time of Arrival of the next bus.

Introduction

Bus schedules tell users of a public bus system when buses are supposed to leave terminals and when they might arrive at important stops, but actual departure and arrival times can vary from the scheduled times. Traffic jams, traffic lights, and large numbers of people boarding and alighting a bus all affect the timing of buses running their routes. Besides having access to schedules, users like to be informed of the real situation so they can plan around deviations from the schedule.

Some transportation systems in this country have Automatic Vehicle Location (AVL) systems installed. AVL systems relay information about the vehicles (position, speed and heading, etc.) to a central office. From there, computer software can analyze that information and distribute it to dispatchers and/or to the public.

Needless to say, there is much demand from bus users who want to know how far away the next bus is from their stop. Using data from an AVL system, software can be built to fill this need.

The goal of this project was to create an Estimated Time of Arrival (ETA) computer algorithm for fixed-route public transit vehicles. This Bus ETA algorithm can be used to publicly display real-time predictions of wait times.

Possible Approaches

To gather data for a computer program that will predict the ETA of a bus at a stop, the bus system must have some sort of AVL system installed. An AVL system might have some sort of location sensing electronics installed in the vehicles and a means of relaying data back to an operations center, or it might have electronic tags in the vehicles and sensing equipment in the roads along the route.

In a paper¹, Dailey describes how a statistical model can be created to model the motion of buses along their routes. His model assumes that the position data for the buses are expressed in terms of a linear distance from the start of the route. This approach works well with a sign-post-based AVL system, where the electronics in the street sense, perhaps via radio tags on the vehicles, when the vehicles pass a certain point. The computer that receives the AVL data can easily map a sign-post to a definite linear distance by looking up that information in a table of known distances.

Some AVL systems, however, sense their location using Global Positioning System (GPS). GPS units in the vehicles receive a radio signal from GPS satellites in orbit around the Earth and use physics and trigonometry to find their exact three-dimensional position on the surface of the Earth, which is expressed as latitude, longitude, and altitude. Finding the linear distance along a bus route from a bus' known latitude and longitude is a complicated process that is prone to error, so Dailey's algorithm would be difficult to implement on bus systems with this type of AVL system.

One possible work-around for this problem is to create *artificial* sign-posts inside the

computer software. The software could watch the GPS data and set off a flag when the bus moved close enough to a special point, whose linear distance along the route is known. This would be difficult to implement in the real world because GPS data from the AVL system is available so far apart in time—positions are reported in one- to five-minute intervals—that a sign-post could easily be missed.

For this project, a different kind of algorithm was developed that involves partitioning a route into zones. The algorithm tracks a bus' entrance into and exit from a zone, and how much time it spent in the zone. The transit time through a zone is stored in a table, and it can later be used to predict how long it will take an individual bus to get from where it is to a particular stop.

Assumptions

The algorithm presented in this paper makes the following assumptions:

- There are GPS units in fixed-route buses out on the road, and at regular intervals of at most a couple of minutes, the following information is transmitted to an operations center and relayed to the algorithm in real-time, possibly along with other data:

BusID	unique ID of this vehicle
DateTime	date and time of the record
RouteName	Which route is this bus running right now?
Latitude	North-South position coordinate on the Earth's surface
Longitude	West-East position coordinate

- The buses follow simple routes from a starting point, to an ending point, and back.

How the Algorithm Works

Each route is divided into a series of zones. (See [Illustration 1](#).) Each zone is arbitrarily drawn to approximate how far a bus will likely travel in about 1½ to 3 minutes.

Transit times, t_i , through each zone in the route are stored in a table in the computer's memory. Every time a bus passes through a zone, the length of time spent in that zone is used to update the time table.

A prediction time for a particular bus can be calculated as follows:

$$t_{total} = \frac{t_1}{2} + t_2 + t_3 + \dots + t_{n-1} + \frac{t_n}{2}$$

where t_{total} is the time left before the bus arrives, t_1 through t_n are the most recently observed transit times through each zone involved, from the zone the bus is currently in through the zone for which the prediction is being made. The first and last zones are counted 50% because, on average, the bus' and the target's geographical position will be half-way through a zone.

Creating Software to Implement the Bus ETA Algorithm

A combination of Microsoft Visual Basic and Maptitude were used to build a software implantation of the prediction algorithm. Maptitude is an inexpensive Geographic Information Systems application program from Caliper Corporation for Windows computers. While it is usually used interactively by an operator, its functionality can also be incorporated into other software.

A Visual Basic class module was created called cRouteModel. (See listing "[cRouteModel.cls](#)" and [related files](#).) A parent Bus ETA program for the algorithm creates one instance of the RouteModel object for each bus route it tracks. Incoming GPS data from our prototype site, Cape Cod Regional Transit Authority (CCRTA), is sorted by RouteName and then sent to Maptitude to determine which zone the records belong to. Then each record is passed on to its appropriate RouteModel object.

Following the method previously described, each RouteModel object keeps a table of zone times, and when queried, produces a prediction as a sequence of time values. To make a prediction, the RouteModel object takes an imaginary walk from the hypothetical user's position (where he would like to be picked up) back along the route to each bus that is currently being tracked. How far away each bus is, in time, is noted and passed on to the parent Bus ETA program for presentation to the public.

Throughout the process, the Bus ETA software takes care to note the direction of the movement of the buses through the zones, and the direction a user would like to travel if he is picked up. Transit times through a zone in one direction are recorded separately from times through the same zone in the opposite direction. This feature helps the Bus ETA software to allow for phenomena, such as rush hour traffic, that cause buses to move faster in one direction along a route than the other.

Uses of the Bus ETA Software

Public web site

The Cape Cod Advanced Travel Planner web site (www.e-transit.org/cape_cod) already shows the Cape Cod bus routes. (See [Illustration 2](#).) and the current positions of all the buses. The software that draws the buses' positions on the web site's map can be extended to use this algorithm to publish ETA information on a same web site.

Wireless phones and palmtop computers

With the growing popularity of so-called "wireless Internet" services, the next logical step for the Bus ETA software is to make its predictions available to users of small wireless Internet devices. Such devices are generally capable of using the Wireless Application Protocol (WAP), and adding WAP capability to an existing old-fashioned web site is relatively easy.

Electronic signs

If an electronic sign is capable of being connected to the Internet via radio or telephone, it could be programmed to receive ETA data for its location from the Bus ETA software running on a web server. Such signs could be installed at bus stops and bus terminals where there is heavy traffic, keeping even the most casual users informed of the state of the bus system.

Quality control / quality assurance

Dispatchers and administrators of a bus system could use the Bus ETA software to keep an eye on their system. By comparing the ETA data to the printed schedules, they can tell at a

glance whether or not buses are running on time.

Conclusion

The predictions made by this algorithm are more accurate than the published schedules because they are based on where the bus is right now, and how long it took to cross the distance between the user and the bus on the last run of the route. The algorithm has been tested and proven valid by feeding it a few hours' worth of historical data and then testing its predictions against later data which was not given to the algorithm. A live demonstration of the Bus ETA software is being developed using only the Provincetown Shuttle route on Cape Cod.

Some issues still need to be worked on. The communications link between CCRTA Operations and the GeoGrahpics Laboratory, which relays GPS data to the Bus ETA software, goes down often, causing the software to lose track of buses. Because of technical issues at CCRTA Operations, the RouteName field in GPS data coming into the Laboratory doesn't reliably represent which route the record belongs to; until this is fixed, the Bus ETA program can't sort all the data to the correct route, and much of it gets discarded.

After the problems have been addressed, the software will be deployed on all of the Cape Cod routes, and its output will be accessible via the web site (www.e-transit.org/cape_cod).

Acknowledgments

This project was made possible by a grant from the Bridgewater Undergraduate Research Opportunities Program (BUROP).

Thanks to: Dr. Uma Shama, advisor; Mr. Larry Harman; GeoGrahpics Laboratory at BSC; Cape Cod Regional Transit Authority; Dr. Edward Brush and Dr. Andrew Harris, co-directors of the BUROP program; BSC Honors Program; BSC Academic Affairs;

Bridgewater Foundation.

1D. J. Dailey, S. D. Maclean, F. W. Cathey, S. R. Wall. "Transit Vehicle Arrival Prediction: An Algorithm and a Large Scale Implementation." Transportation Research Board, 80th Annual Meeting, January 2001, Washington, D. C.

[\(back to web site\)](#)