DRAFT

An Evaluation

The Mobile Data Communications for Bus and Rail Automatic Vehicle Location Demonstration Project

Performed by

Sound Transit
Research and Technology

and

Daniel J. Dailey, Fredrick W. Cathey, and Stuart Mclean
University of Washington
ITS Research Program
Department of Electrical Engineering

and

Washington State Transportation Center (TRAC)

August 2003

DRAFT

Table of Contents

DRAFT

List of Figures

List of Tables

DRAFT

Executive Summary

The Mobile Data Communications Demonstration Project provides a functional test and demonstration of a multi-agency, GPS-based, automatic vehicle location (AVL) system. This multi-agency system includes vehicles and data for: (1) Sound Transit bus service operated by Community Transit and Pierce Transit, (2) Community Transit and Pierce Transit internal transit service, as well as, (3) Sounder Commuter Rail service. Schedule data, both spatial and temporal, is obtained from all the agencies and is combined with software developed by the University of Washington called MyBus/ Busview. The original MyBus application (mybus.org) predicts arrival/departure times and presents them in a publicly available web page.

The application created in this project, (http://www.its.washington.edu/multi-modal/), performs these functions using data from all the agencies participating. In addition, it provides the functionality of Busview, a Web-based vehicle location display, for transit agencies and customers, across all the agencies. This document reports the results of a joint Sound Transit and University of Washington evaluation effort of the Mobile Data Communications Demonstration Project.

Automatic vehicle location (AVL) is the technology that allows transit agencies to perform such tasks as: (1) real-time schedule adherence, for operations and planning, (2) provision of traveler information, such as vehicle arrival/departure times for customers, (3) interior stop announcements and displays, for on-board customers, and (4) automatic vehicle location information to dispatcher, for operations and safety. For purposes of this demonstration, the two primary objectives of this GPS-based AVL are:

(1) To provide multi-agency, real-time vehicle location information to a central processing system on the ITS data backbone, called MyBus/ Busview, via a commercial carrier's wireless network. The central processing would then perform the tasks of calculating the vehicle's arrival time at a given location, as well as display the vehicle's real-time location on a Web-based map display.

(2) To identify those issues surrounding multi-agency data integration into the single application using TCIP standards and methods. The schedule data, spatial and

temporal, that is provided by each agency are different based on that agency's operational requirements and functions. There is no regional uniformity for this type of data, although Community Transit and Metro use Hastus scheduling software; therefore, it is of benefit to identify the issues and limitations when conflating data to these applications.

Creating a multi-modal version of the publicly available transit information applications is both technically and politically challenging. The four agencies who provide data to this effort are at very different stages in the use of real-time data. Obtaining usable schedule information, both spatial and temporal, can be a challenging activity in such an environment. However, in a relatively short time, we constructed multi-modal applications useful to all four agencies. This was facilitated by a database designed around TCIP (Transit Communications Interface Profiles) concepts and a pre-existing component architecture that allowed us to reuse components and plug together new applications incrementally.

The GPS-based tracking implementation proved to be effective in most cases, with the exception of some downtown Seattle locations. Since most of Sound Transit's services and the other agencies' bus services operate in areas outside the downtown core, a GPS-based AVL solution would work for most service. In the case of downtown service, the addition of dead reckoning equipment would improve the position estimates.

Coordination between the spatial and temporal schedule efforts and any future AVL effort will need to improve if the AVL system is to be successful. For example, in this project, we created much of the spatial information for the Pierce Transit schedule using the GPS information from the vehicles. A second example is the trip assignment for the Sounder train. The trip assignment of the car that had the GPS receiver mounted on it was not known in advance, and in some instances we were forced to guess the trip assignment, not always successfully. The current train trip operations randomly assigns cars to train trips on the day of operation, which makes it difficult to predetermine car trip assignment. This can be remedied through better coordination with the train operations and the addition of GPS receivers to all train trips.

Sound Transit received assistance from its partner agencies including Amtrak, Community Transit, and Pierce Transit. Also, the University of Washington (UW) ITS

Program provided the technical work and equipment selection with Digital Recorders, Inc. performing the equipment installation. Sound Transit provided the overall funding and project management responsibilities.

DRAFT

1. Introduction

This research project was designed to test the concept of providing low-cost, integrated, regional AVL capability using a publicly available communications network along with the existing Web-based Busview display and MyBus prediction application. The test and evaluation will be used to assess the benefits, costs, and implementation issues (technical and institutional) associated with the concept.

The project lasted for a period of eight months and included three phases: Phase I provided the agency with data inventory and initial application development; Phase II included equipment installation, testing, and configuration; and Phase III provided the functional demonstration of the applications, as well as an evaluation period. The participating agencies included:

Sound Transit Sounder Commuter Rail

Sound Transit Regional Express

Pierce Transit

Community Transit

King County Metro

Amtrak

UW ITS Research Program

Digital Recorders, Inc.

AT&T Wireless

2. Background

This project was approved for funding in 2000 as part of the Sound Transit Research and Technology Program. The project was proposed in response to needs identified in the December 1998 Regional Transit Communication Technology workshop. The project was also reviewed and recommended for funding by the multi-agency Regional Transit Technology Group (http://www.rttg.org/).

A long list of ITS achievements has led to the establishment of an Intelligent Transportation Systems Research Program in the College of Engineering at the University of Washington. This program is a multi-disciplinary effort between the Departments of Electrical Engineering, Civil and Environmental Engineering, and Technical Communication. Research efforts focus on the application of computer and communications technologies to solving transportation problems. The ITS research program actively collaborates with government and industry, making it a regional resource for advanced answers to transportation issues.

3. Project Objective

The objective of the project is to demonstrate the integration of multi-agency vehicle location data using GPS (Global Positioning System)-based technology transferred via a commercial wireless carrier's data network. For this demonstration, AT&T's CDPD wireless standard for data exchange was used. The location of the multi-agency vehicles were then displayed in "real-time" or used in the existing MyBus and Busview applications for multi-agency service monitoring, coordination, and customer service information.

Currently, MyBus and Busview operate only within the King County Metro service area providing locational information for Sound Transit and Metro bus services. King County maintains a fixed-point signpost-based AVL system as opposed to the other traditional type of AVL or GPS-based vehicle location. The expansion of this and the supporting applications into both Snohomish County and Pierce County transit service districts using the respective transit agency data is a primary outcome of this demonstration. Accomplishing this objective required the following:

1. Acquisition of local transit information from Community Transit, Pierce Transit, and Sound Transit
2. Integration of the static and dynamic data through the data representation standards of Transit Communications Profile (TCIP)
3. Use of the ITS SDD protocols for data transfer
4. Installation of GPS equipment on transit agency vehicles for Automatic Vehicle Location
5. Multi-modal facility demonstration at (a) King Street Station and the 4th Ave bus zone, (b) Lynnwood Park and Ride, (c) Tacoma Dome Station platforms and bus zone, (d) Puyallup Station and bus zone, and (e) Sumner Station and bus zone.
6. Integrated multi-agency vehicle tracking within the Busview application.

4. Expected Benefits

This project used applications created in SmartTrek (www.Smattrek.org) to leverage an AVL demonstration for several additional transit properties. Specifically, the project will derive the following short-term and long-term benefits:

Short Term

- This project will require that base maps for each of the transit properties be combined to provide regional coverage. Available public data sets were used to display the spatial information.
- The schedule information for the three agencies were combined in order to provide a demonstration of overlapping services. The TCIP-based database used as the data repository for the MyBus and Busview application was used to accomplish this data integration. This will be the first significant integration of this type of data for use in a real application for the Puget Sound region.
- The database created for this project has potential to be widely useful to Sound Transit and participating agencies well beyond this demonstration.
- The demonstrations project identifies issues with multi-agency data used in a single application and data format

Long Term

- If successful, the application can facilitate necessary transfers between transportation modes in order to complete an intra-regional journey. It is anticipated that as a result travelers will increasingly choose alternatives to the automobile for inter-city travel within the corridor.
- It is also expected that the operators of multi-modal facilities will be able to improve connections and better coordinate operations as a result of having ready access to schedule and real-time information about multiple transportation service providers.
- This project can lay the framework for both the public and the private sector to expand traveler information services in the state.

5. Technical Description

In past work, we described a system of components that are used to construct transit traveler information applications. In this report, we describe an ongoing project that uses the same component architecture to combine maps, schedules, and AVL information from four transit agencies. The result is a set of web applications suitable for both transit management and traveler information. We describe both the technical and administrative challenges to be overcome in building a regional (10,000 square miles) transit AVL and information system.

5.1 Background Work

In past work, we described a system of components that are used to construct transit traveler information applications. These applications include Busview [1], MyBus [2] and Probes [3]. In this report, we describe an ongoing project that uses the same component architecture to combine maps, schedules, and automatic vehicle location system (AVL) information from four transit agencies. It is a real-time demonstration of a multi-modal, multi-agency traveler information system conducted over a three-county region of 10,000 square miles. It demonstrates the viability of traveler information and transit management systems that span four agencies in Washington State, USA, (Sound Transit, Pierce Transit, Community Transit, and Metro King County Transit) and two vehicle types (transit buses and commuter rail service) as well as two types of automatic vehicle location systems (GPS and signpost-assisted dead reckoning). The Busview and MyBus programs have been enhanced to now include information from all the agencies mentioned.

The objectives of this multi-modal, multi-agency project are to (1) identify data requirements necessary to create a regional system, (2) demonstrate the scalability of the component architecture, (3) demonstrate the viability of multi-modal, multi-agency displays to the transit management personnel, (4) demonstrate multi-agency traveler information, and (5) demonstrate the viability of using a standard transit scheme to support several applications using data from multiple agencies.

We present the architecture of the applications and discuss the data support needed to make these kinds of applications a reality. We assert that establishing a standard set of data structures from which all the applications obtain the schedule information and then mapping each agency's data to those structures is an efficient method to resolving data differences between agencies of varying sizes.

5.2 Component Architecture

The applications architecture presented here is created using the component architecture described in [4], where the interprocess communication is done using self describing data (SDD), as described in [5]. Figure 1 is a representation of the applications where the individual components are shown in the boxes and the lines connecting the boxes are interprocess communications channels that use SDD over TCP/IP. This kind of modular application is desirable for a variety of reasons: (1) The application started with one transit agency and easily expanded to four. (2) The application scales because the processes can be distributed across a number of processors as the data processing requirements expand. (3) The component model is "plug and play" in that many of the components are reused to expand to more agencies. (4) The multiple data-stream architecture supports the varied timing and format of the schedule updates for the four autonomous agencies. (5) The actual tracking and prediction algorithms can be different based on the type of AVL system. (6) The data representation uses the TCIP description language for commonality across agencies. (7) The modular nature allows for flexibility in creating derivative applications. For example, we have constructed a playback version for use by the senior management that combines recorded GPS data and components configured as if they were in a deployed version. This last point explicitly acknowledges that there are both technical and political aspects to creating a multi-agency application and both these aspects must be explicitly accounted for in any successful multi-agency, multi-modal design. The components that make up the applications presented here are shown in Figure 1.

Figure 1 shows two types of vehicle location systems at the top. Right-most data flow is for King County Metro and has been reported on numerous times [6]. The data flow diagram on the left represents the deployment of a group of heading-augmented GPS positioning systems. The individual vehicles report: vehicle_id, a unique integer identifier for vehicle hosting the GPS unit; age, the quality of the state data; GPS_time, the time of GPS report in seconds since midnight at Greenwich; and latitude, longitude, speed, and heading. The on-board systems communicate the location data, using UDP over CDPD modems, to a process

DRAFT

at the University of Washington that creates a self describing data flow, the GPS_SDD Transmitter. This “Transmitter” creates an SDD [5] data stream suitable for use with the component toolbox [7] available from the University of Washington ITS group. The SDD stream encapsulates the AVL reports from all of the available vehicles and adds the Cartesian state plane projection coordinates for each vehicle to support downstream tracking and viewing applications.

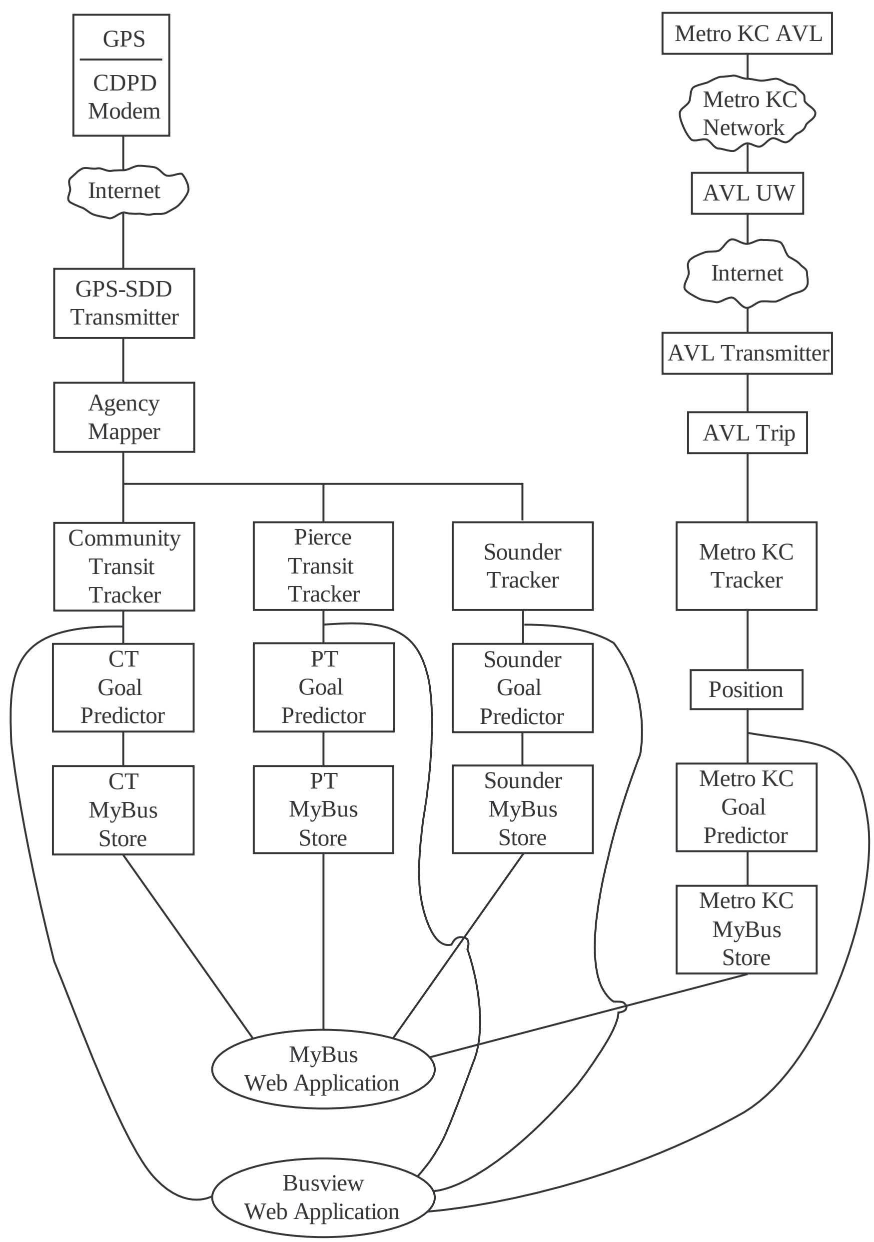

Figure 1: Application data flows.

The next component is the "Agency Mapper." The purpose of the Agency Mapper is to identify the agency that owns the vehicle reporting and "de-multiplex" the data stream. The Agency Mapper creates a stream of data for each agency. It augments each input AVL Report with both an agency_id, in this case, one of four, and a block_id, that is the piece of work that each vehicle is scheduled to perform. The Agency Mapper makes a stream of data from each agency available on an individual TCP port, one port per agency.

The purpose of the Tracker is to assign schedule adherence information to each vehicle operated by a given agency. This information includes identification of the route, pattern, and trip that the vehicle is currently on, as well as estimated distance into trip and schedule deviation. There is a Tracker for each agency that processes the corresponding stream of Position Reports output by the Agency Mapper. Each Tracker is configured with the schedule database of the corresponding agency. A multi-hypothesis tracking technique is used to manage candidate trip hypotheses generated from vehicle report times and locations and to select the most likely hypothesis for output. The tracker functionality is described in more detail in [8]. For each input Position Report, the tracker outputs a corresponding Block Track Report consisting of the information from the Agency Mapper augmented by a number of calculated values in the form of a "standard" AVL report that is shown Table 1

Table 1: Standard AVL report.

Message	Meaning	TCIP
Element		Data Dictionary
vehicle_id	The vehicle number	OB_J1587_VehicleIdentificationNumber
agency	The agency that owns the bus	CPT_AgencyID_cd
status	Ok, no_block_assignment, no_trip_assignment, ...	
local_date	year, month, day, hour, minute, second, milli	CPT_DateTime_tm
block_id	The piece of work the bus performs	SCH_BlockID_id
trip_id	Specifies the start and end time for a TPI	SCH_TripID_id
pattern_id	A sequence of TPI's	SCH_PatternID_id
route_id	Identification number for a route	SCH_RouteID_id
distance_into_block	The distance already traveled on a block	
distance_into_trip	The distance already traveled on	

	a trip	
deviation	Deviation from schedule	OB_ScheduleAdherenceOffset_tm
flags	Special vehicle properties like express bus	SCH_RunTypeDescription_txt
measurement_time	UTC time	CPT_DateTime_tm
latitude, longitude	Location of the vehicle in latitude and longitude	SP_Longitude_sp,SP_Latitude_sp
x, y	Local state plane coordinates	SP_SPEasting,SP_SPNorthing_sp_sp
speed	Ground Speed of Vehicle	OB_RollingAverageSpeed_rt
heading	The direction the vehicle is headed	OB_J1587_CompassBearing
last_beacon_id	The identification Number of last beacon	
distance_last_beacon	Distance traveled since last beacon	

In the US, there is a standard named "Transit Communications Interface Profiles" (TCIP) that attempts to describe data types for transit communications systems. In the third column of Table 1 is the TCIP data dictionary descriptive name for the element in our standard AVL message, where they exist. Several notions used by our applications do not seem to be present in the TCIP. The notion of tracking vehicles over an entire block is central to our algorithms but does not seem to be present in the TCIP framework. Also, one of our four agencies, Metro KC, has an AVL system that uses beacons, and this notion does not seem to be in TCIP.

Downstream of each tracker, one per agency in Figure 1, is a Goal Predictor component. For each vehicle, the Goal Predictor computes arrival/departure times at selected locations. The algorithms for this component can be found in [8]. The output of the Goal Predictor is placed in a component we have named a "Store." The function of the Store component is to hold all the predictions for all the vehicles and provide them to downstream applications as they request the information.

Two Web applications, Busview and MyBus, shown at the bottom of Figure 1, make the information available on the world wide web. These applications connect into this set of components at differing locations due to the different information required by the Web application. Busview is a graphical representation of the location of the transit vehicles on a regional map; as such, it needs location, agency, service route, heading, and vehicle identification. These data are available from the Trackers for each agency, and the Busview Web application connects to the trackers, as shown in Figure 1. The MyBus Web application displays prediction of departure at selected locations; as a result, it is connected to the Store. Both of these applications demonstrate the advantages of shared data streams and reusable components.

5.3 Standard Form for Transit Data

The schedule data, both spatial and temporal, comes from four agencies in four different formats and contains data with differing levels of completeness. In this report, we assert that by establishing a set of "standard" data structures for use by all types of applications and transforming these disparate sets of agency data into a common representation, we can efficiently create a multi-modal, multi-agency set of applications. We have established six data structures that make up our "standard" set of information for any transit agency: *time-point, time-point-interval, pattern, trip, event, block.*

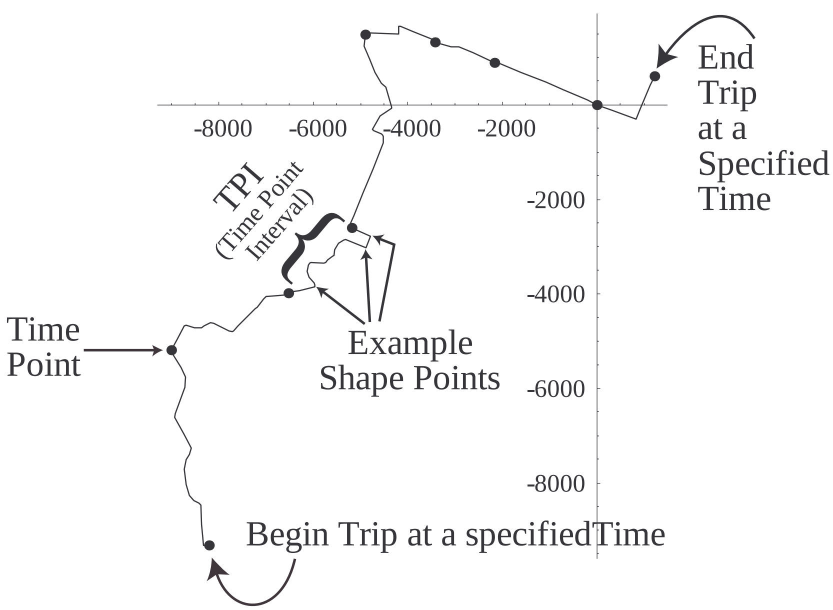

Figure 2: Five of six standard concepts.

A ***time-point*** is a named location (See Figure 2). The location is generally defined by two coordinates, either Cartesian state-plane coordinates or geodetic latitude and longitude. For the example applications, we use NAD-83 state-plane coordinates and also assume that the transformation from geodetic to state-plane coordinates, and its inverse, are known [9]. We choose to use Cartesian coordinates for computational reasons as the geometry and metric are Euclidean rather that ellipsoidal. Transit vehicles are scheduled to arrive or depart at time-points.

A ***time-point-interval*** (TPI) is a polygonal path representing a stretch of road directed from one time-point to another (See Figure 2). The path is geographically defined by a list of "shape-points," where a shape-point is simply an unnamed location. Since one frequently needs to determine the distance of a vehicle along a path, each shape-point is augmented with its own distance into path. The length of a TPI is the distance into path of the last shape point (the ending time-point).

A ***pattern*** is a "route" made up of a sequence of TPI's, where the ending time-point of the i-th TPI is the starting time-point of the (i+1)-th. Note that the sequence of TPI's on a pattern determines a sequence of time-points on the pattern. (The converse is not necessarily true since there may be more than one TPI running from one time-point to another.) The distance-into-pattern of a TPI is defined to be the sum of the lengths of the preceding TPIs, and the length of a pattern is the sum of the lengths of its TPIs. For a vehicle traversing a pattern, the index of the TPI that the vehicle is on and the vehicle's distance-into-TPI uniquely determine the distance-into-pattern of the vehicle. Figure 2 shows a sample pattern plotted in state-plane coordinates.

A ***trip*** specifies a start time and end time for every TPI on a pattern in such a way that the end time for the i-th TPI is no greater than the start time for the (i+1)-th. Note that it is possible for the same time-point to be assigned two successive times, in which case we say that a *layover* is scheduled at that point. Note also that a trip specifies a travel time for each of its TPI's, and different trips on the same TPI may specify different travel times depending on the time of day. In Figure 2, the traversal of the path at the times specified is an example of a trip.

An ***event*** is the triple of trip, time-point, and time. At any given time-point at a given time, many events may be scheduled, since many buses on different patterns may be scheduled to arrive/depart the same location. The "Begin Trip at specified time" in Figure 2 is an example.

A ***block*** is a sequence of trips such that the schedule time for the end of the i^{th} trip is no greater than the schedule time for the start of the $(i+1)^{th}$ trip. If the end time-point of the i^{th} trip is the same as the start time-point for the $(i+1)^{th}$, but the schedule times are different, then we say that a layover is scheduled at that point. Each transit vehicle is assigned a block to follow over the course of the day. Some blocks are long and are covered by different vehicles at different times of day.

With these six constructs, it is possible to represent a transit schedule in a database from an arbitrary agency; further, these map directly to like named concepts found in the 1999 Transit Communications Interface Profiles (TCP) Documents [10] labeled NTCIP 1404.

5.4 Agency Legacy Schedule Data to Standard Schema Transformation

The four agencies who provide data to this effort are at very different stages in developing the use of real-time data. The requirements on the schedule, spatial and temporal, are more stringent when it is used with real-time AVL data. To address the data disparities between the agencies, we have created a conceptual model of the data framework required for real-time applications. The conceptual model we use is that data from each of the agencies will be transformed or normalized to fit into a standard set of data structures against which all of the applications will operate. Figure 3 shows this concept graphically.

5.4.1 MetroKC

The first agency, Metro King County (MetroKC), has a mature odometer-based and sign-post-augmented AVL system with both management function and traveler information. Metro contributes to the TCIP standardization process, and the applications presented here were originally constructed using data from MetroKC. A simple normalization of the Metro data makes them compatible with our standard schema.

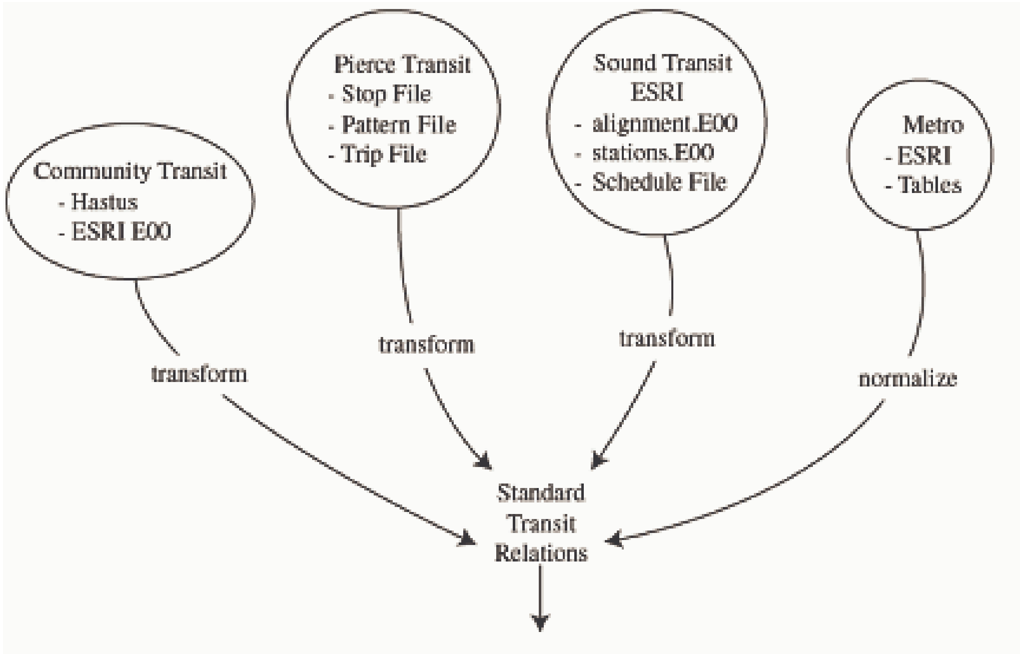

Figure 3: Transforming legacy data to the standard schema.

5.4.2 Community Transit

The second agency is Community Transit (CT), the transit operator for Snohomish County, Washington. This agency does not presently have a full AVL system; they do have some GIS (Geographical Information System) capability and plan their schedule using the product Hastus, made by Giro Inc.

Table 2: Example of CT data.

route	direction	Schedule	variant	seqnum	time_pt
411	South	Weekday	sb	1	99air
411	South	Weekday	sb	2	
411	South	Weekday	sb	3	
411	South	Weekday	sb	4	
411	South	Weekday	sb	5	marpr
411	South	Weekday	sb	6	
411	South	Weekday	sb	7	
411	South	Weekday	sb	8	9stew
411	South	Weekday	sb	9	
411	South	Weekday	sb	10	
411	South	Weekday	sb	11	
411	South	Weekday	sb	12	2pike

The CT spatial data model is based on a concept called a segment. In principle, a “segment” is an ordered sequence of geodetic arcs. In addition to spatial points, the segments have attributes that include a service route identifier, day of week for operation, route direction, and route variant, as well as a sequence number and possibly a time point name. A subset of the segment data, showing segments for Route 411 Southerly, is shown in Table 2. Segments with the same key first four values are used to build up TPI’s and patterns. Taking all segments with the same key, ordering them according to the ‘seq_num’ attribute, and noting time_pt values, it is, in principle, possible to build standard TPI’s and patterns from the data provided. In practice, since the agency does not operate an AVL system, the geographic information the agency maintains does not have to be self consistent. The arc data underlying the segments had missing arcs, adjacent nodes that were actually the same node, and multiple names given for the same node. In order to assemble the spatial schedule information, we defined node equivalency to be a small region, ordered the arcs, and filled the gaps either from another map source or by linearly connecting the nodes.

For example, using the date from Table 2, we create time-point intervals between time-points ‘99air’ and ‘marpr’, between ‘marpr’ and ‘9stew’, between ‘9stew’ and ‘2pike’. The shape-point sequence for each TPI is built by chaining together the arcs comprising the TPI. Thus, the TPI from ‘99air’ to ‘marpr’ contains segments with seq_nums 1 through 5. The 4 TPI’s in order make the new pattern whose numeric id is derived from the 4 tuple string values (411,South,Weekday,sb).

CT’s Hastus files included time points, trip, and event information. The key values to link the temporal to the spatial information, for each trip, are the 4 values: route identifier, schedule type (day of week run), route direction, and route variant. These four values appeared in the spatial data above and in the temporal data output from Hastus. The CT trip data contains a block identifier. However, the block identifiers provided with the schedule were preliminary place holders inserted by the planners. The true block assignment was done by Coach USA Inc, a contractor who operates the vehicles for CT. The application architects combined the CT schedule with the Coach USA blocking data to create the standard data set required.

DRAFT

5.4.3 Pierce Transit

The third agency, Pierce Transit, has the least sophisticated bus scheduling system of the three bus operators and supplied schedule information in three text files. One file provides information about every bus stop. Bus stop data includes stop identifier, location (specified in Washington South state plane coordinates), and a descriptive name.

The avlstops file provides information about every bus stop. Bus stop data includes (among other things not relevant to our purposes) stop identifier (integer), location (specified in Washington South state plane coordinates), and a descriptive name.

Here is a representative line of this file:

3580;;other;shill;39TH;7TH;10;5TH;;S Hill Mall Transit
Ctr;;;X;;;;17610;;;Middle;;669359;1192995;;X;;;;SOUTH HILL
TC;NW;N;10;;;;;;;;;;;X;X;98373;;P;0;S Hill Mall Transit Ctr

This is an encoding for the bus stop with the properties: stop_id = 3580, x = 669359, y = 1192995, and name = 'S Hill Mall Transit Ctr'.

A second file provides a list of “trip-patterns.” Each trip-pattern is specified by an integer identifier, a route number, direction (e.g., Inbound, Outbound), and a list of “shapes.” A shape is comprised of a pair of bus_stop_ids and a list of geographic points (in Washington South state plane coordinates) defining a polygonal path from the first stop to the second.

trip-pattern : {id, route_id, direction, list of shapes}

shape : (from_stop_id, to_stop_id, x1, y1, x2, y2, ...)

Here is a representative record in the file (a single line):

10;410;Inbound;
3580;72;1192995;669359;1192301;669375;1192266;669376;;
;;
72;70;1192266;669376;1191635;669390;1191555;669392;;
;;
70;68;1191555;669392;1190956;669410;1190945;669411;1190295;669430;1190224;66943
2;;
...

It identifies a trip-pattern with

```
pattern_id = 10
route_id = 410
direction = Inbound
list of shapes:
  {from = 3580, to = 72,
   (1192995,669359),
        (1192301,669375),
        (1192266,669376)}
  {from = 72, to = 70,
        (1192266,669376),
        (1191635,669390),
        (1191555,669392)}
  {from = 70, to = 68,
        (1191555,669392),
        (1190956,669410),
        (1190945,669411),
        (1190295,669430),
        (1190224,669432)}
  ...   .
```

A third file provides temporal schedule information. Each line in this file defines a scheduled trip along a trip-pattern in a certain block of work. Information includes pattern_id, block_id, description, and a list of events. There is an event for each bus stop occurring in a shape on the associated trip-pattern. Each event consists of a triple (stop_id, time, checkmark). The time of an event is specified in minutes after midnight and has a resolution of 1 minute. The times for distinct events are not necessarily distinct for nearby stops. The checkmark for an event specifies whether or not the event is used for schedule adherence purposes, a “time-point-event” in our standard terminology. The first and last events on a trip are always “checked,” and the times on checked events are always distinct. A number of trips contained duplicate events which had to be removed.

Here is representative line from this file:

```
 0; 410;      68;Inbound;410 Parkland;;30;;02/02/03;06/07/03;10;3580;  495;1;;72;  495;0;;70;
496;0;;68;  496;0;;66;  497;0;;64;  498;0;;80;  498;0;;62;  499;0;;93;  499;0;;60;  500;0;;58;
501;0;;56;  501;0;;4115;  501;0;;76;  502;1;;74;  503;0;;53;  505;0;;89;  505;0;;51;  506;0;;91;
507;0;;49;  508;0;;87;  509;0;;47;  510;0;;82;  511;0;;3880;  512;0;;78;  513;0;;491;
514;0;;492;  514;0;;495;  515;0;;496;  515;0;;2121;  516;1;;
```

It identifies a trip with

block_id = 68
pattern_id = 10
description = '410 Parkland'
list of events:
{3580,495,1},
{72,495,0},
{70,496,0},
... .

The first event on this trip has stop_id = 3580, time = 495 minutes after midnight (8:15 am), and is checked. Note that the second event at stop_id = 72 is at a different location. It has the same time as the first event but is unchecked.

The Pierce Transit data is converted to standard form by creating time-points from the bus stop data and time-point-intervals from the shapes listed on the trip-patterns. The coordinates of all geographic points were transformed into the Washington North state plane coordinate system in order to place the data from all the agencies into the same coordinate system. This was accomplished by transforming each point given in Washington South state plane coordinates to WGS-84 latitude and longitude, using the inverse of the Lambert Conformal transformation for that zone, and then transforming the latitude and longitude into Washington North state plane coordinates using the direct Lambert Conformal transformation for the north zone. This introduced no more than 1 foot of error in the length of any shape.

Standard pattern structures were created from the trip-pattern data and the time-point-intervals. Standard trip and time-point-events were created from the Pierce trip data. Only checked events were used as time-point-events. Trips with the same block_id were sorted by their start times to form standard block structures.

Spatial data was unavailable from Pierce Transit for the I-5 freeway portions of the blocks joining Seattle to Tacoma. As a result, the real-time trip data cannot be assigned to the vehicle scheduled to work these blocks whenever it is on I-5. GPS data are collected from the vehicle while it travels from Seattle to Tacoma in the afternoon. The vehicle locations are used to construct a new block with complete spatial data. A “bread crumb trail” of points is created by sampling the vehicle locations every 3 seconds, starting at a point just before the location of the first scheduled event and continuing to a point just after the location of the last scheduled event. Each event time-point, TP, on the trip is mapped to the closest point on the trail, P'. The distance between TP and P' is small for most time-points (20 ft). For each time-point, TP, a modified time-point, TP', is constructed at location P'. Time-point-intervals are created using sub-sequences of the bread crumb trail that joins consecutive points P'. Time-point-events for the new block are created by modifying the existing ones to use the new time-point locations.

5.4.4 Sounder/Sound Transit

The fourth agency, Sound Transit, operates a commuter train service called the Sounder. The Sounder spatial data were provided in ESRI Arc/Info format (E00 files) named cr_alignment.e00 and cr_stations.e00, and the temporal data were in a text file. The temporal data consist of arrival/departure times for each of the sounder trains, north-bound and south-bound, at each of 7 commuter rail stations from Seattle to Tacoma. Unlike the agencies above, there is no information related to the concepts of block, trip, or pattern in the data provided by Sound Transit. These data structures are created by the application architects.

An example of the schedule.txt:

```
Prj_Num   Train1N   Train2N Train3N Train1S  Train2S Train3S
245  6:15 6:30 6:45 16:55     17:10     17:35
243  6:27 6:42 6:57 17:11     17:25     17:51
241  6:32 6:47 7:02 17:18     17:33     17:58
239  6:40 6:55 7:10 17:25     17:40     18:05
237  6:47 7:02 7:17 17:34     17:49     18:14
235  6:54 7:09 7:24 17:38     17:53     18:18
231  7:15 7:35 7:45 17:55     18:10     18:35
```

The cr_alignment file, that defines the spatial layout of the sounder rail system, is a network of arcs. Each arc specified in this file consists of the following information: arc_id, segment_id, fnode, tnode, list of shape points. The segment_id, when numeric, provides a cross reference to the name of a train station associated with the arc. Fnode and tnode are simply numeric identifiers for the start and end points on the arc. The shape points are provided in Washington North state plane coordinates.

Several records are needed to construct an arc. For example, the following record partially specifies the King Street Station arc:

```
      14        33        13        12         0         0         3
 1.2713483E+06 2.2142978E+05 1.2713571E+06 2.2172553E+05
 1.2713630E+06 2.2192180E+05
```

Here we have

```
arc_id = 33
fnode = 13
tnode = 12
list of shape points =
  {1.2713483E+06, 2.2142978E+05},
  {1.2713571E+06, 2.2172553E+05},
{1.2713630E+06, 2.2192180E+05} .
```

Elsewhere in this file is a record associating a segment_id and length with this arc. For example, the following lines associate segment_id = 231, length = 492 feet to arc_id = 33.

```
      16         15          0          0 4.9223666E+02         14         33
CR   operate   231                            0
```

The cr_stations file specifies which arcs are associated with train stations. Each station is assigned a numeric segment_id that belongs to a unique arc. For example, the following record associates the King Street Station with segment_id 231, (arc_id = 33).

```
 0.0000000E+00 0.0000000E+00          4          4 2.3100000000000000E+02King S
treet Station        Seattle     301 South Jackson Street              Commuter Rail Station
```

Operational

A “path,” consisting of 20 arcs, that joins the King Street Station in Seattle to the Tacoma Dome Commuter Rail Station, was constructed manually.

An initial obstruction to completing the path was encountered between Puyallup and Tacoma. Arc 46, segment_id = SL-7, joining nodes 28 and 30 seems not to be correctly specified. However, by deleting its first shape point, we obtained an arc from node 26 to node 30 and we can complete the path. The path from King Street Station to the Tacoma Dome Commuter Rail Station is listed below.

id, segment_id, fnode, tnode (station name)

{14, 231 , 13, 12} King Street Station
{11, SL-1, 14, 13}
{13, 233 , 15, 14}
{28, SL-2, 16, 15}
{29, 235 , 17, 16} Tukwila Commuter Rail Station
{30, SL-3, 18, 17}
{27, 237 , 19, 18} Kent Commuter Rail Station
{26, SL-4, 20, 19}
{25, 239 , 21, 20} Auburn Commuter Rail Station
{24, SL-5, 52, 21}
{2, 241 , 53, 52} Sumner Commuter Rail Station
{12, SL-6, 55, 53}
{3, 243 , 54, 55} Puyallup Commuter Rail Station
{55, SL-7, 54, 32}
{51, SL-7, 32, 24}
{49, SL-7, 24, 26}
*{46, SL-7, 28, 30} (drop first shape point to get arc from 26 to 30)
{31, SL-7, 30, 31}
{45, SL-7, 31, 34}
{44, 245 , 34, 35} Tacoma Dome Commuter Rail Station

However, the Sounder start and stop points in Tacoma are at the site of the “new” Tacoma Dome Commuter Rail Station which is under construction. This site is approximately 1/4 mile north of the existing station. A train traveling south from Seattle begins on the originally specified path but then follows an uncharted path to the site of the new station. To fix this problem, we constructed an arc using recorded GPS train locations (a “bread crumb trail”) that replaces the last six arcs of the originally specified path.

To transform Sounder data to a standard form, time-points were constructed for each station and time-point intervals were constructed by “gluing together” intermediate arcs. For south-bound trips, the time points consist of the southern node of the King Street Station arc, the southern most node encountered along the path at each intermediate station arc, and, finally, the terminal node located at the site of the new Tacoma station. A similar procedure was used to define time-points and TPI’s for north-bound trips. A south-bound pattern was constructed from the south-bound time point intervals, and a north-bound pattern from north-bound time point intervals. Three south-bound blocks and three north-bound blocks were defined. Each block consisted of a single trip whose time-point events were defined from the schedule file.

5.5 Applications

There are two demonstration applications available to display the multi-modal, multi-agency data: Busview and MyBus. Screen shots of each of the applications can be seen in Figures 4 and 5. Figure 4 is the downtown map for Busview for a Wednesday afternoon. On the screen are largely Metro vehicles (as they operate about 1,200 at this time of day), two Sound Transit Vehicles, a Community Transit Vehicle, and the Sounder Train. Figure 5 is the MyBus departure predictions for Puyallup, a small city south of Seattle where the Pierce Transit vehicles meet with the Sounder Train at the Puyallup Station. The entire day of transit activity is shown on this page.

DRAFT

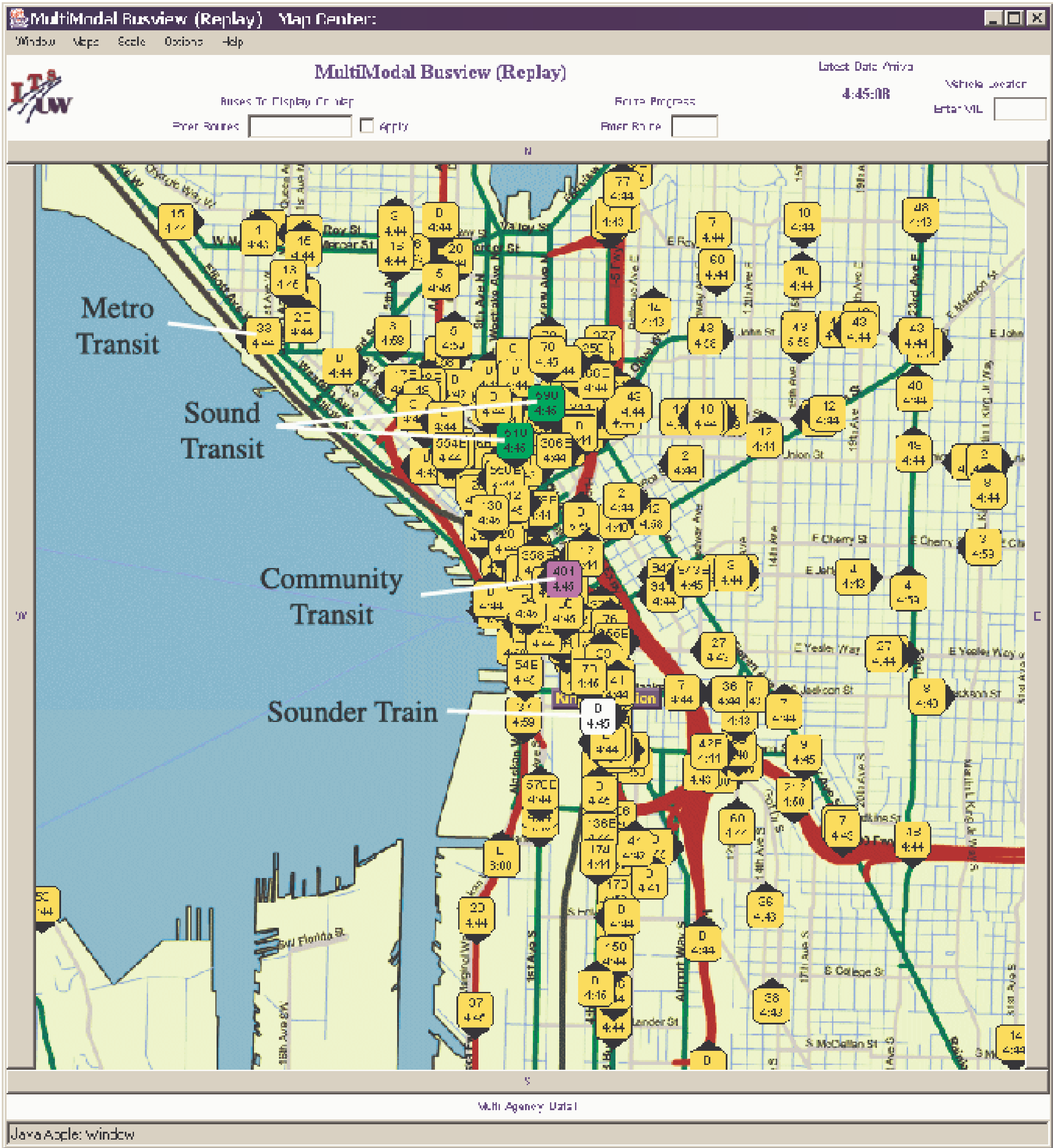

Figure 4: Busview Multi-Modal screen.

Figure 5: MyBus Multi-Modal screen.

5.6 Performance Evaluation

This section examines the on-time performance for the GPS-equipped vehicles. It considers the accuracy of the GPS positioning, compares the GPS reported positions to the roadway-centerline information in the various GIS coverages, and reports the on-time performance for some of the GPS equipped vehicles.

5.7 GPS Accuracy, Availability and Roadway Centerline Correspondence

To evaluate the accuracy, availability, and roadway centerline correspondence, we present a data set from each of the agencies: Sound Transit (ST), Community Transit (CT), Sounder, and Pierce Transit (PT), as well as the vehicles operated by CT and PT for Sound transit. Table 3 shows the agencies and trips in the order that the graphs are presented in this report. Each case reported on is represented by a row in Table 3. The set of figures for each case include: (1) a plot of the deviation from the road centerline for the GPS measurements, and (2) a plot of the spatial trajectory overlaid on the digital map representing the roadway scheduled for use.

Table 3: Agencies and trips in the order of graph presentation.

Date	Agency	Time	Block	Trips	VehicleID	Routes
4/02/2003	CT	AM	204	1,2,3	2	401, 402
	CT	PM	654	4	2	401
3/02/2003	CT/ST	AM	704	1,2	4	511
	CT/ST	PM	704	3,4	4	510
3/02/2003	Sounder	AM	31	3	5	B1
3/03/2003	PT	AM/PM	68	33	1	410/411
	PT	PM	68	34,36	1	410/411
	PT/ST	PM	222	1	3	590

Figure 6 is a time series plot of the distance from the GPS-reported vehicle position and the closest point on the scheduled pattern as determined by the tracker for trip 1. This quantity is denoted by “Delta.” Notice that most of the values are less than 50 feet but that there are spikes a little after time = 330 and again at time = 350 (defined as minutes past midnight).

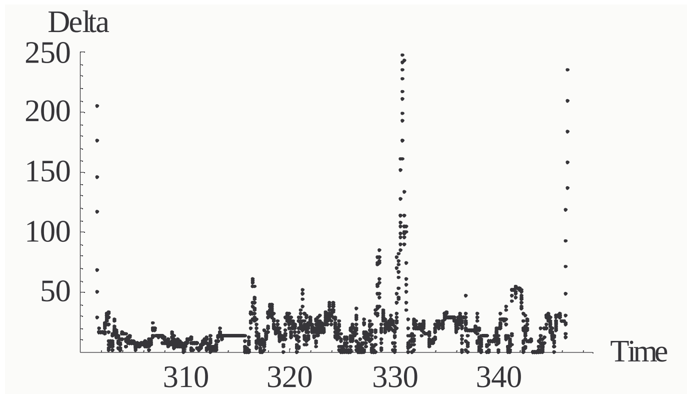

Figure 6: Deviation of GPS-reported location and roadway centerline map.
Vehicle 2, CT, Block 204, Trip 1, AM

Figure 7 is an xy-plane illustration (in state-plane coordinates) of the vehicle position data superimposed on the spatial pattern data for this piece of work. Note the deviation from pattern around time = 332. This corresponds to a section of I-5 just before the exit to Stewart Street. Using Figures 7 , 9, and 11 for the three AM trips, the pattern description is not in error but rather the GPS is having some difficulty here, perhaps due to some tall buildings. Note also the deviation of vehicle trajectory from pattern at the end of trip. This deviation was observed on subsequent trips and suggests a discrepancy in the pattern description for end of trip.

Subsequent, Figures 9 through 32 are similar pairs of GPS centerline deviation and overall x-y deviation for representative data sets for each of the cases enumerated in Table 3.

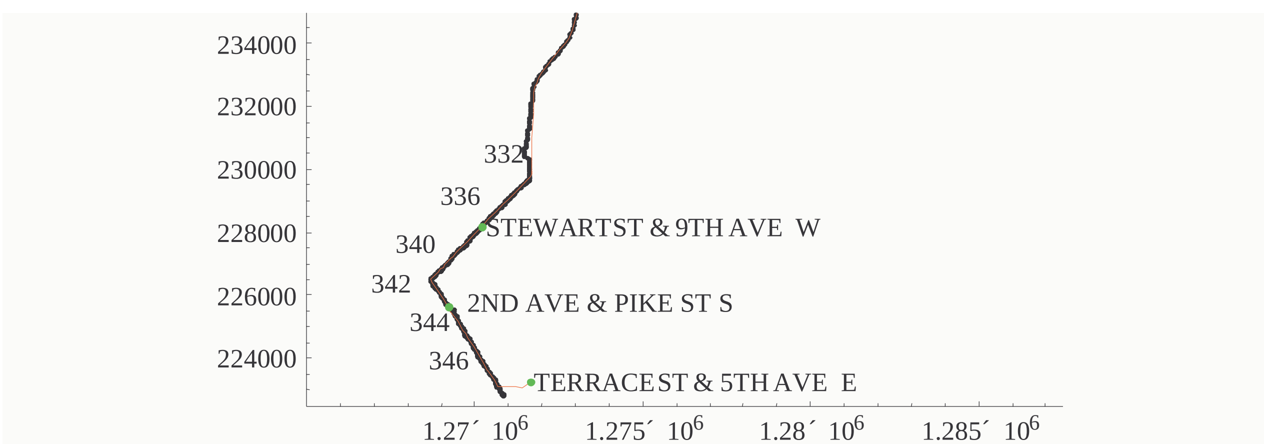

Figure 7: Roadway and GPS reporting.
Vehicle 2, CT, Block 204, Trip 1, AM

DRAFT

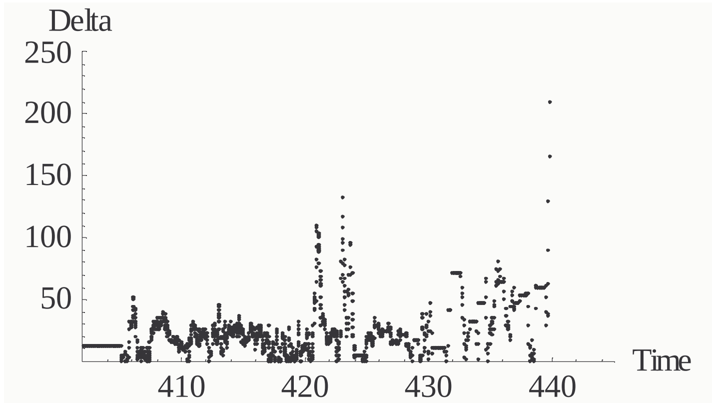

Figure 8: Deviation of GPS-reported location and roadway centerline map.
Vehicle 2, CT, Block 206, Trip 2, AM

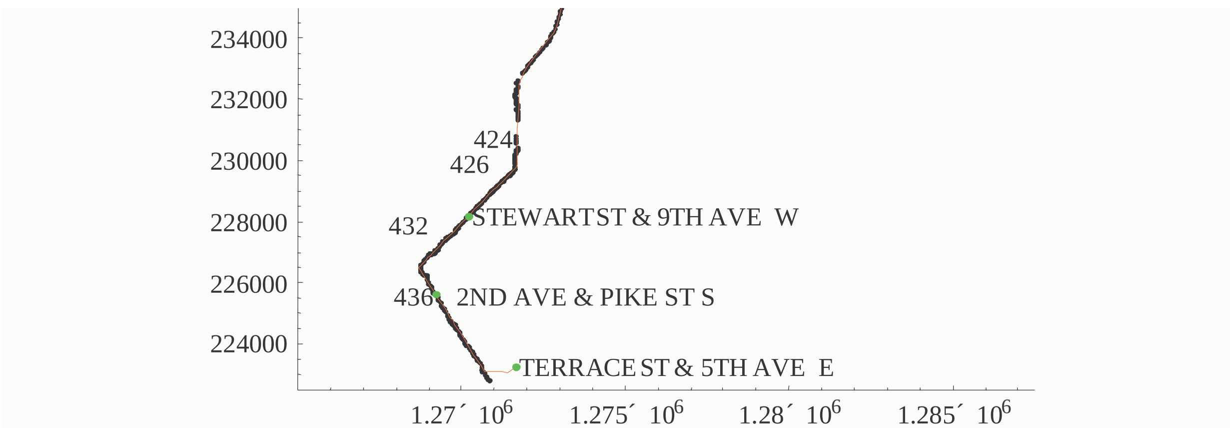

Figure 9: Roadway and GPS reporting.
Vehicle 2, CT, Block 206, Trip 2, AM

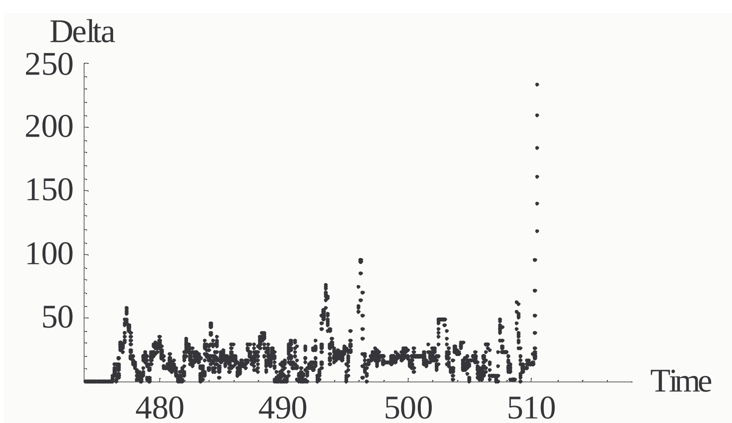

Figure 10: Deviation of GPS-reported location and roadway centerline map.
Vehicle 2, CT, Block 204, Trip 3, AM

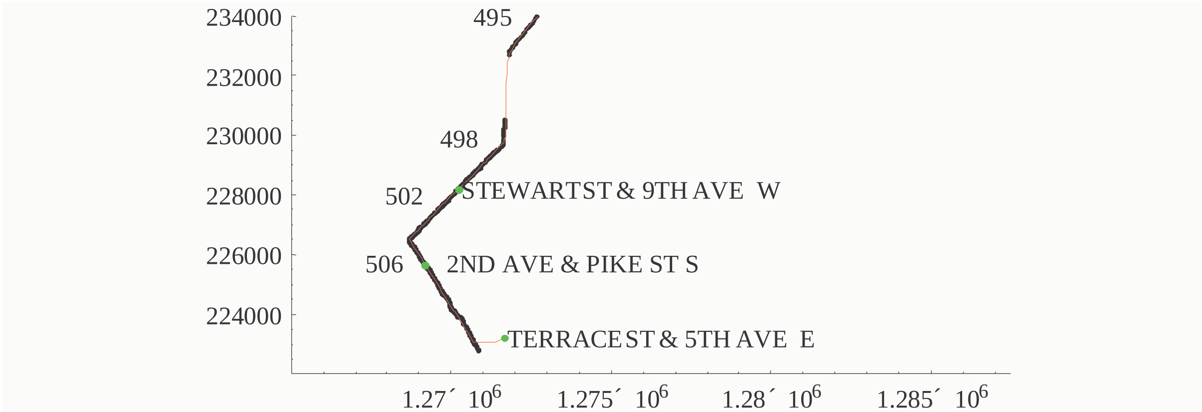

Figure 11: Roadway and GPS reporting.
Vehicle 2, CT, Block 204, Trip 3, AM

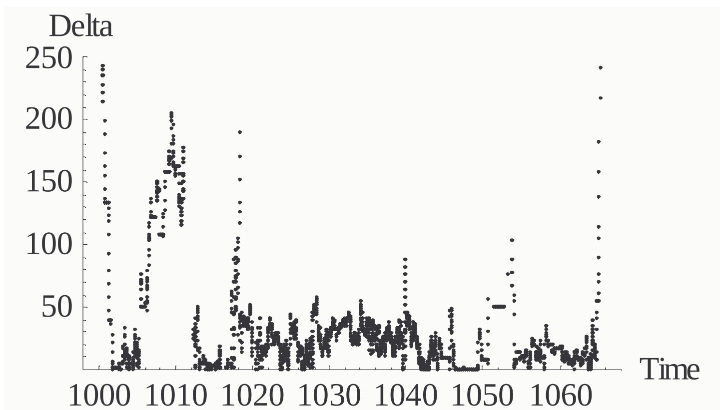

Figure 12: Deviation of GPS-reported location and roadway centerline map.
Vehicle 2, CT, Block 654, Trip 4, PM

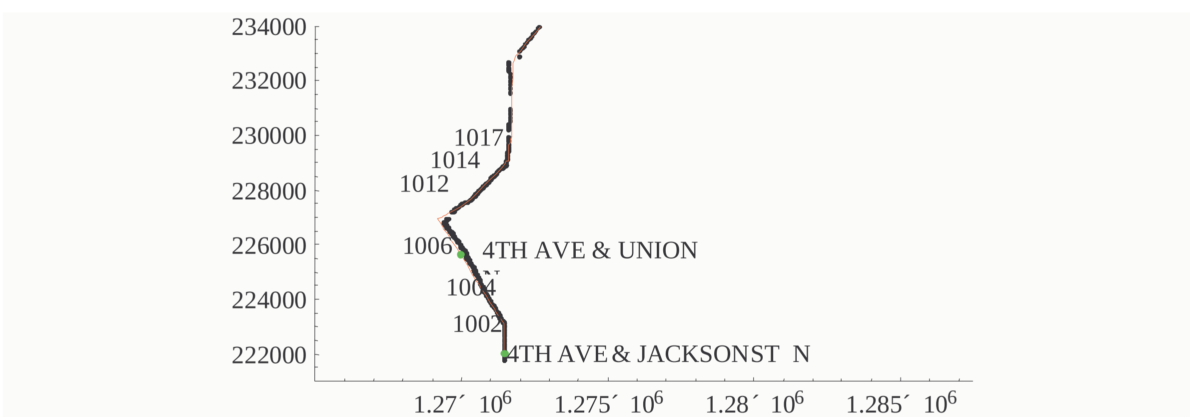

Figure 13: Roadway and GPS reporting.
Vehicle 2, CT, Block 654, Trip 4, PM

DRAFT

Figure 14 shows the spatial deviations for trip1. The distance bias of 100 feet just after time = 390 (6:30) is also illustrated in the xy-plot of Figure 15.

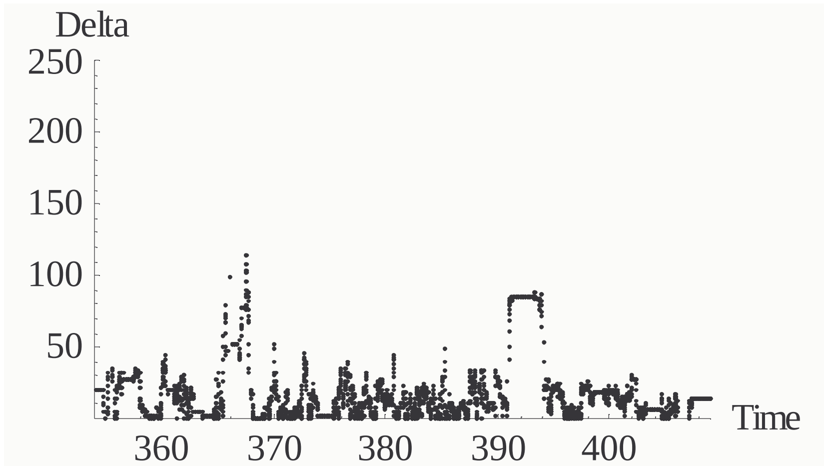

Figure 14: Deviation of GPS-reported location and roadway centerline map.
Vehicle 4, CT/ST, Block 704, Trip 1, AM

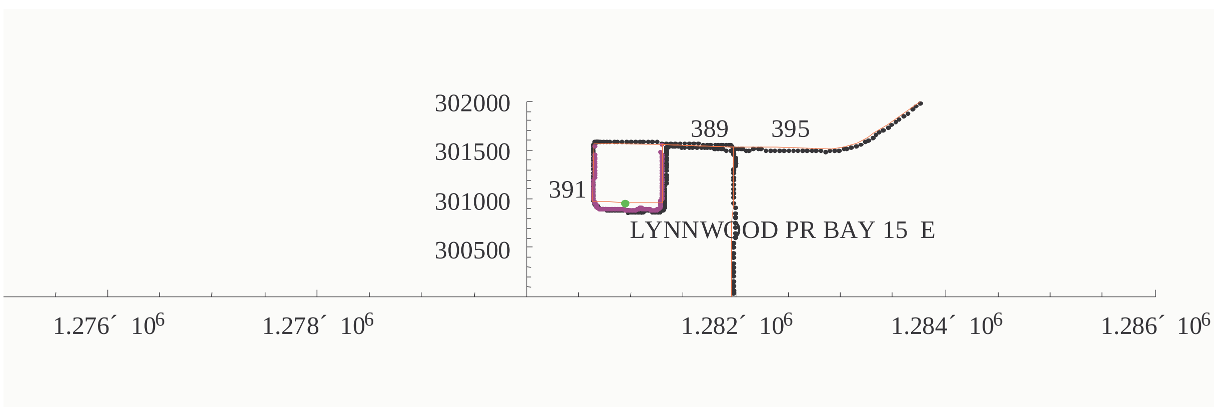

Figure 15: Roadway and GPS reporting.
Vehicle 4, CT/ST, Block 704, Trip 1, AM

Figure 16 shows the spatial deviations for the second trip. The first large spike in delta distance around time = 430 is illustrated in the xy- plot of Figure 17. It appears the vehicle took a short cut which was not part of the pattern.

DRAFT

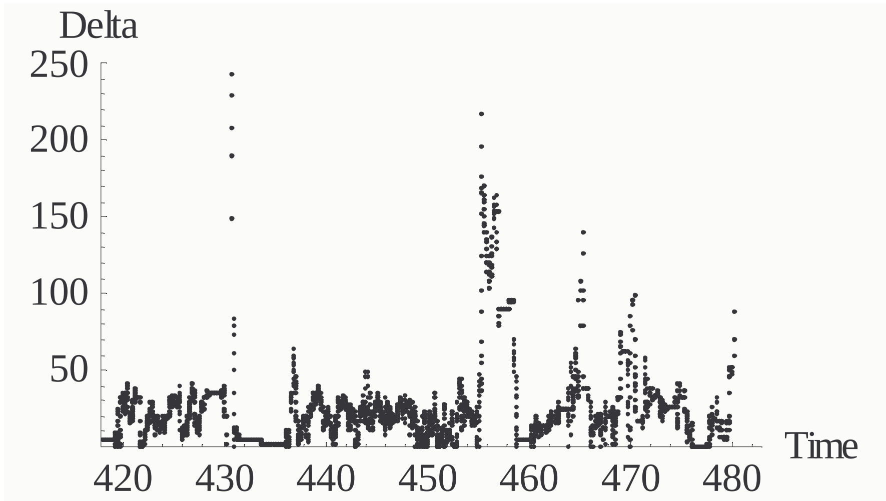

Figure 16: Deviation of GPS-reported location and roadway centerline map.
Vehicle 4, CT/ST, Block 704, Trip 2, AM

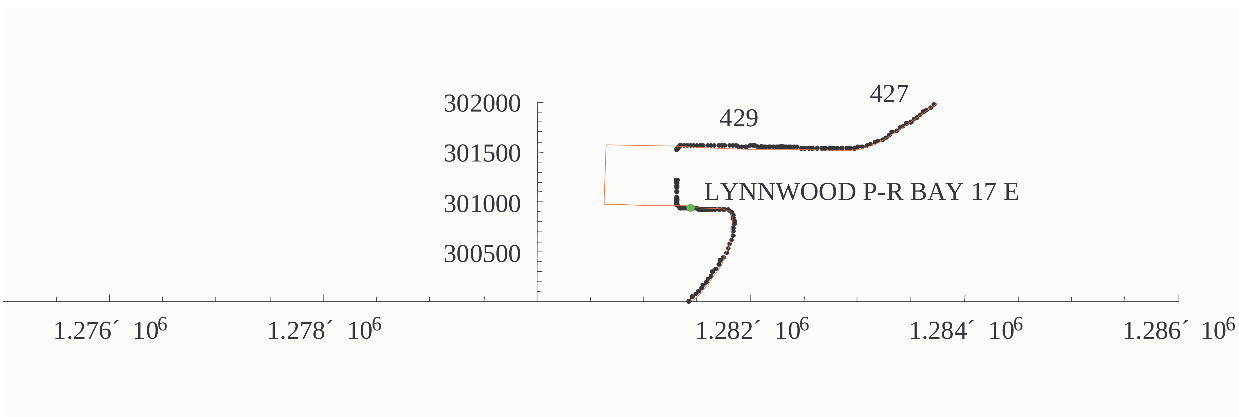

Figure 17: Roadway and GPS reporting.
Vehicle 4, CT/ST, Block 704, Trip 2, AM

Figure 18 show the spatial deviations for trip 3. The distance bias of 100 feet just after time = 390 (6:30) is also illustrated in the xy-plot of Figure 25. The large delta distance around time = 1040 is similar in character to that for vehicle 2 on its afternoon trip at time = 1010.

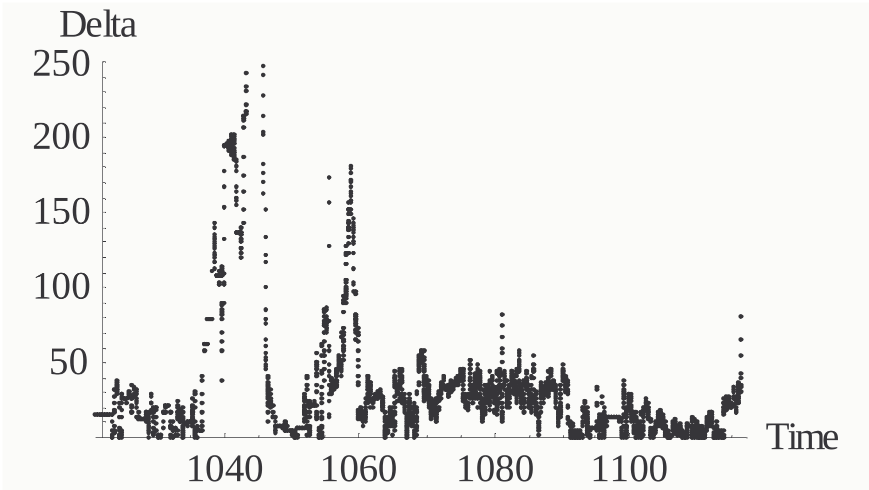

Figure 18: Deviation of GPS-reported location and roadway centerline map.
Vehicle 4, CT/ST, Block 704, Trip 4, PM

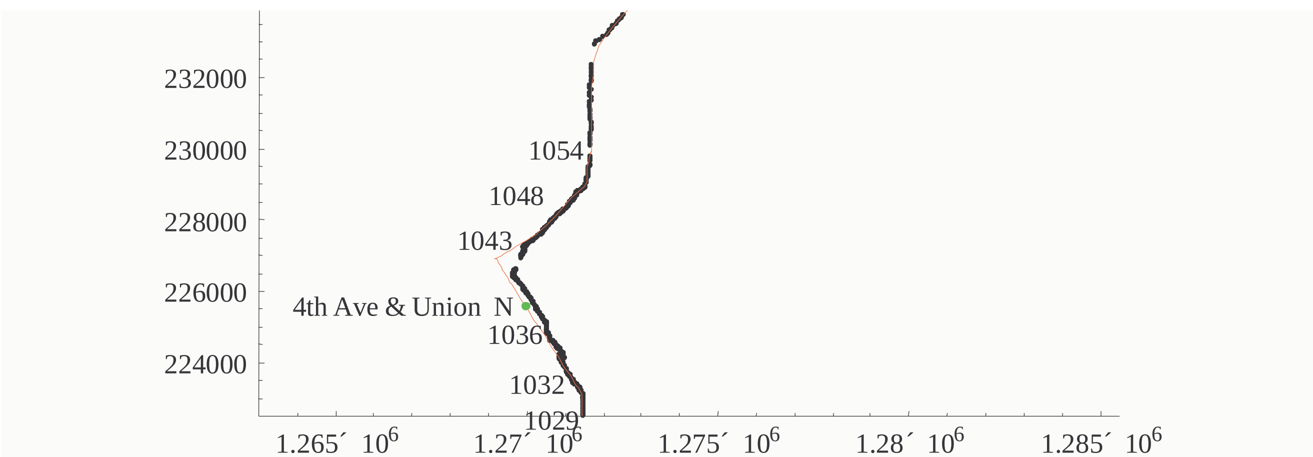

Figure 19: Roadway and GPS reporting.
Vehicle 4, CT/ST, Block 704, Trip 4, PM

Figure 20 shows a detail of the trajectory of trip 3 in the vicinity of the EastMont Park & Ride.

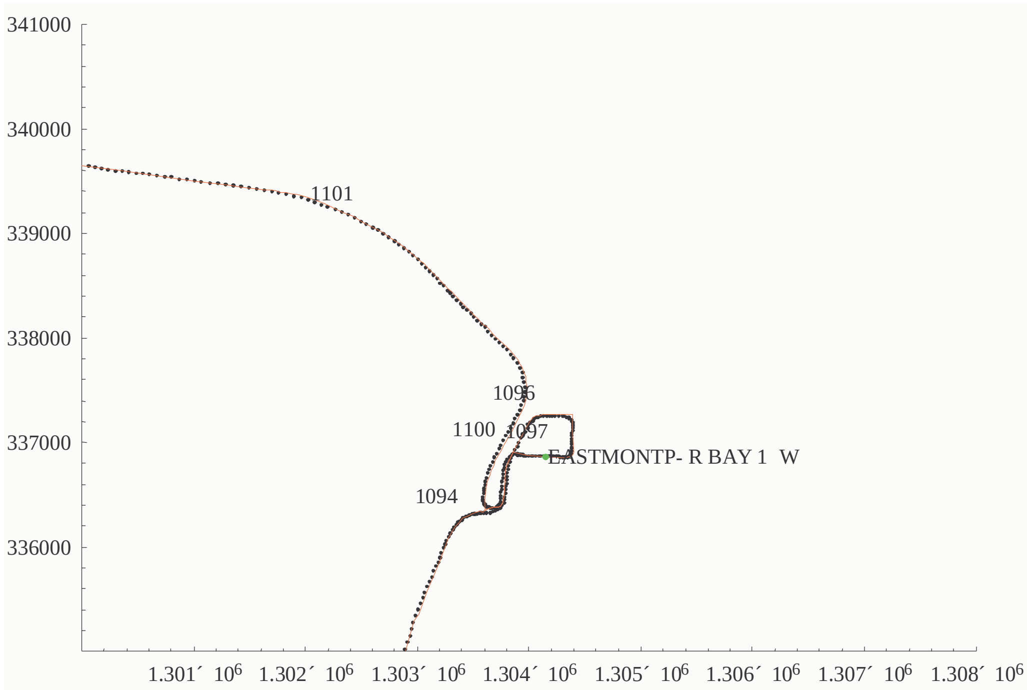

Figure 20: Roadway and GPS reporting.
Vehicle 4, CT/ST, Block 704, Trip 4, PM

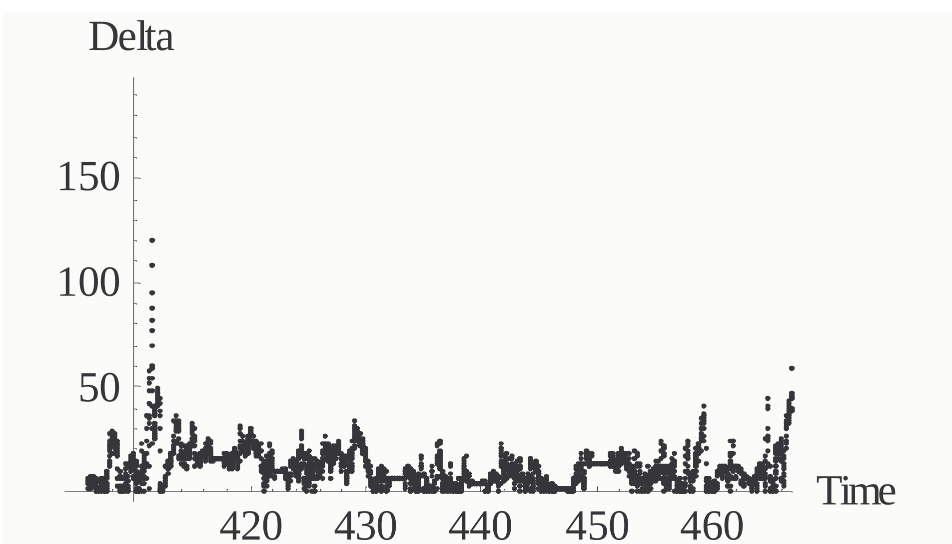

Figure 21: Deviation of GPS-reported location and roadway centerline map.
Vehicle 5, Sounder, Block 31, Trip 3, Northbound, AM

Figure 23 shows an xy-plot of the entire trip. Figure 22 shows a detail of the area in Sumner and Tacoma.

DRAFT

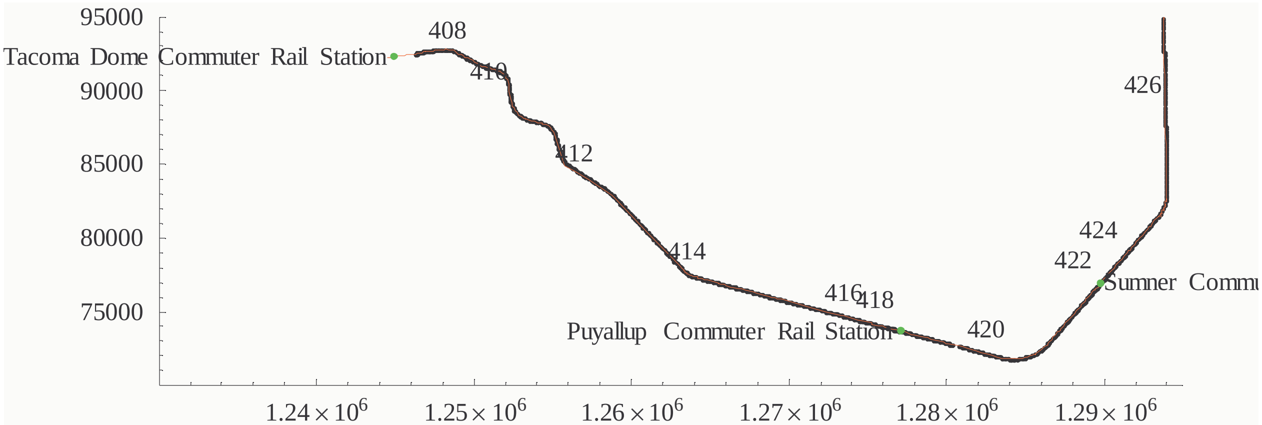

Figure 22: Roadway and GPS reporting – Tacoma/Sumner detail.
Vehicle 5, Sounder, Block 31, Trip 3, Northbound, AM

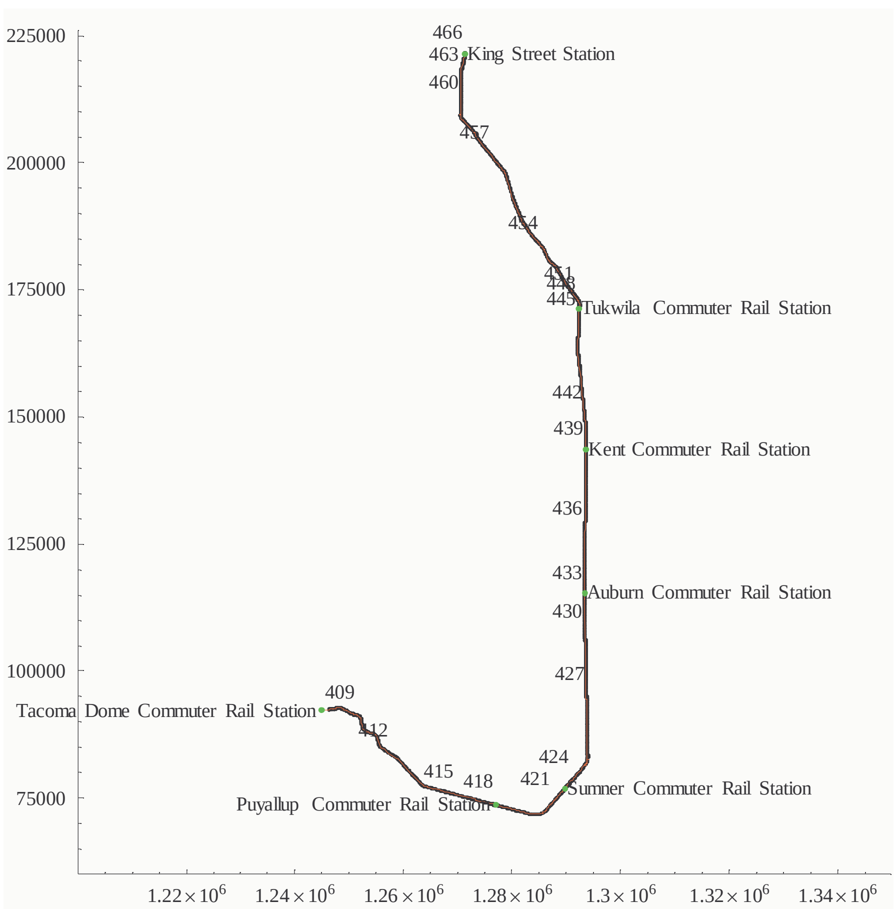

Figure 23: Roadway and GPS reporting – complete route.
Vehicle 5, Sounder, Block 31, Trip 3, Northbound, AM

DRAFT

Figure 24 shows the deviation for trip 33. There is a gap in the data near time = 1010. The reason for this is illustrated in the xy-plot of Figure 25. This figure shows that the vehicle takes a detour at the west end of 112th before continuing to its destination at the Parkland Transit Center. This deviation was sufficiently great that the tracker could not assign the vehicle to the schedule pattern.

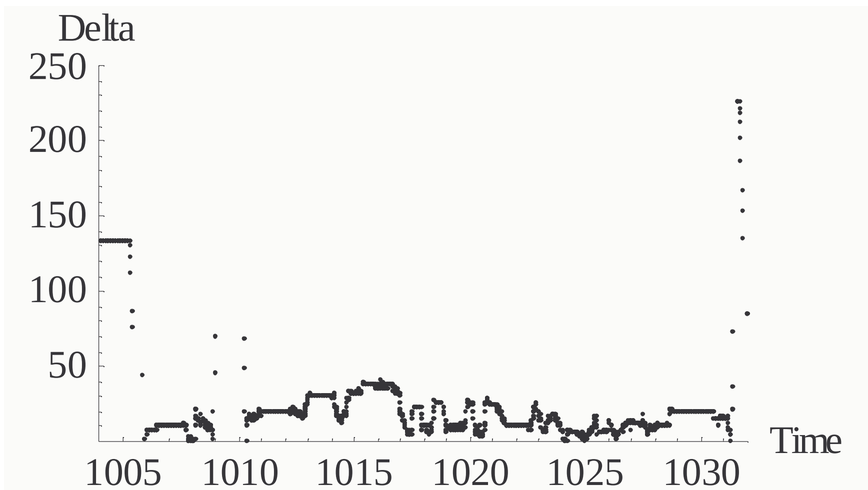

Figure 24: Deviation of GPS-reported location and roadway centerline map.
Vehicle 1, PT, Block 68, Trip 33, PM

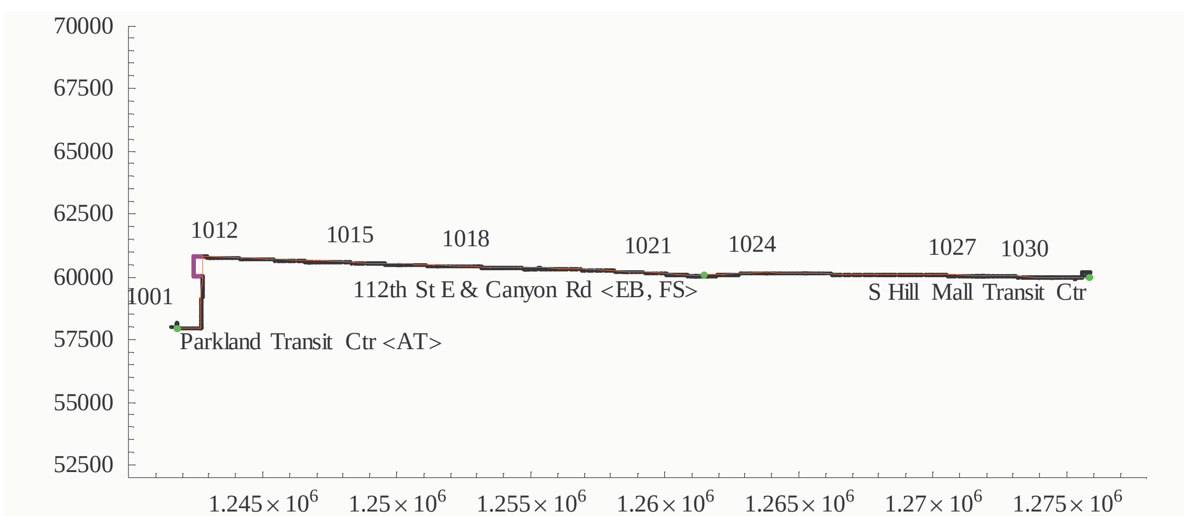

Figure 25: Roadway and GPS reporting.
Vehicle 1, PT, Block 68, Trip 33, PM

Figure 26 shows the deviation for trip 34. The large delta distance at the end of the trip is illustrated in the xy-plot of Figure 27. The vehicle deviates from the schedule pattern just before its destination at 1601 39th Ave NE.

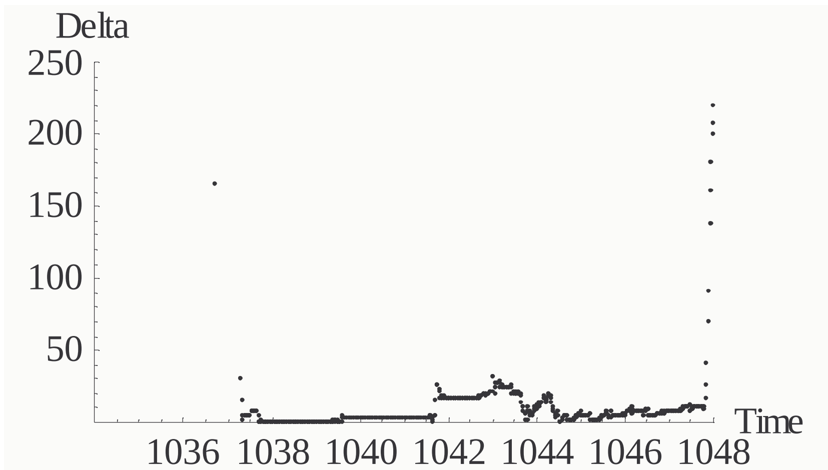

Figure 26: Deviation of GPS-reported location and roadway centerline map.
Vehicle 1, PT, Block 68, Trip 34, PM

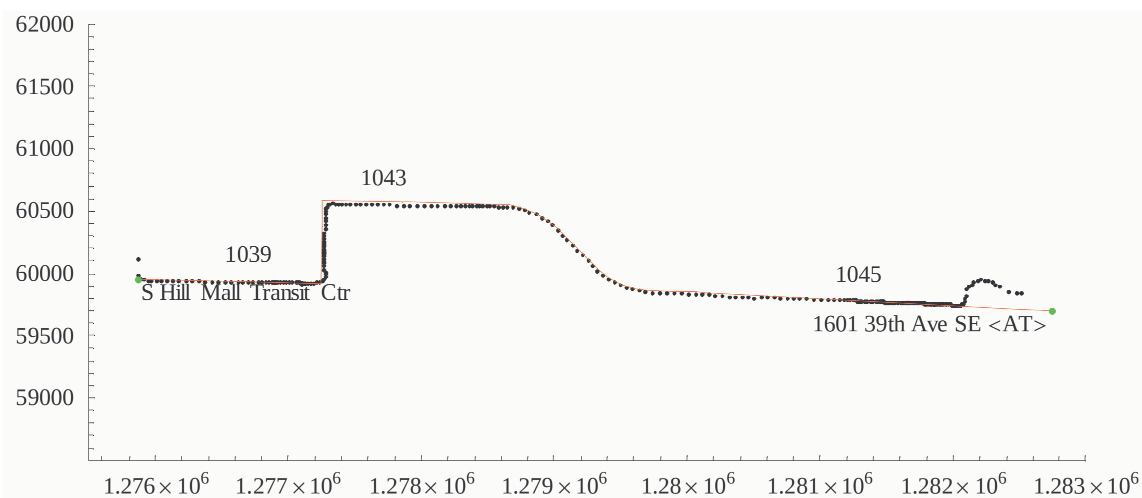

Figure 27: Roadway and GPS reporting.
Vehicle 1, PT, Block 68, Trip 34, PM

Figure 28 shows deviation for trip 36. The regions where there are large delta distances are illustrated in the xy-plot of Figure 29. The vehicle deviates from the schedule pattern at the S Hill Mall Transit Center and again just before its destination at 1601 39th Ave NE.

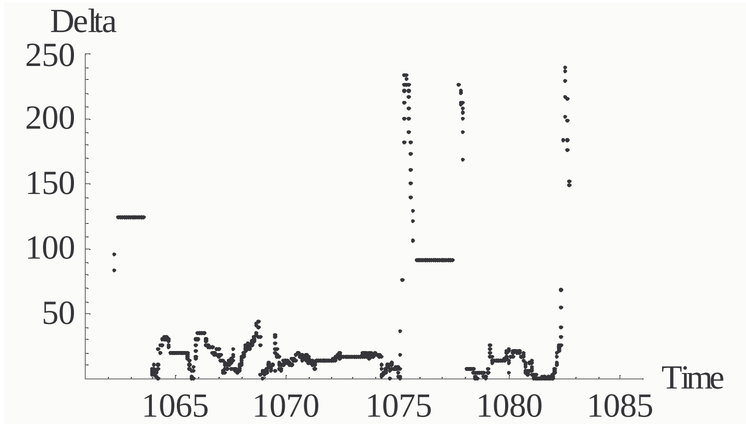

Figure 28: Deviation of GPS-reported location and roadway centerline map.
Vehicle 1, PT, Block 68, Trip 36, PM

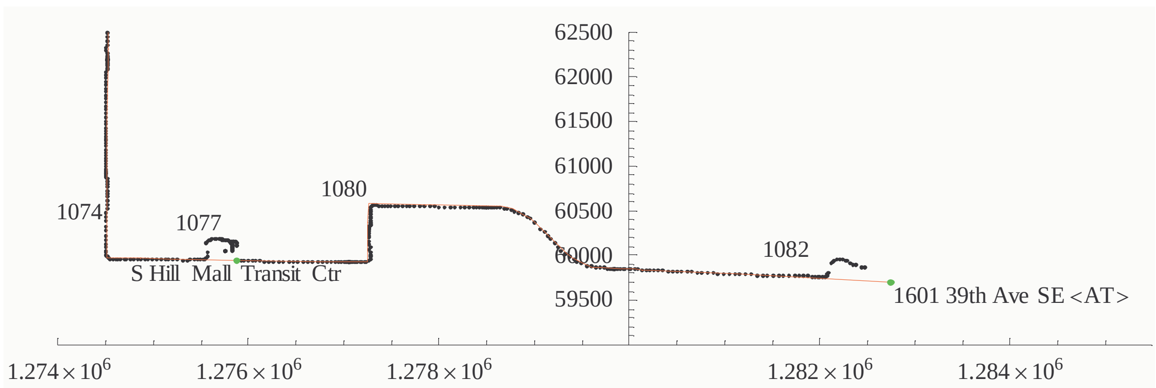

Figure 29: Roadway and GPS reporting.
Vehicle 1, PT, Block 68, Trip 36, PM

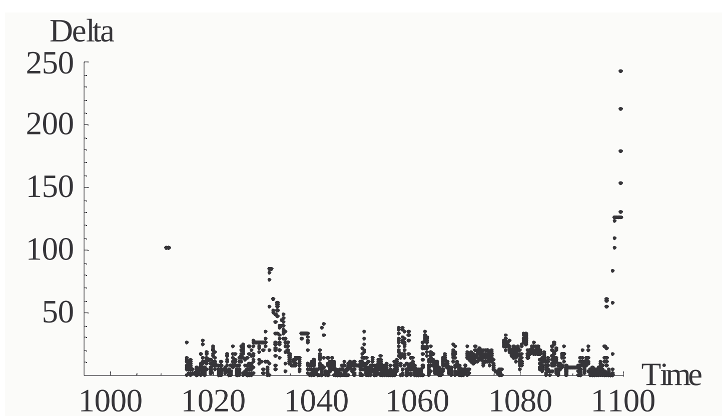

Figure 30: Deviation of GPS-reported location and roadway centerline map.
Vehicle 3, PT/ST, Block 222, Trip 1, PM

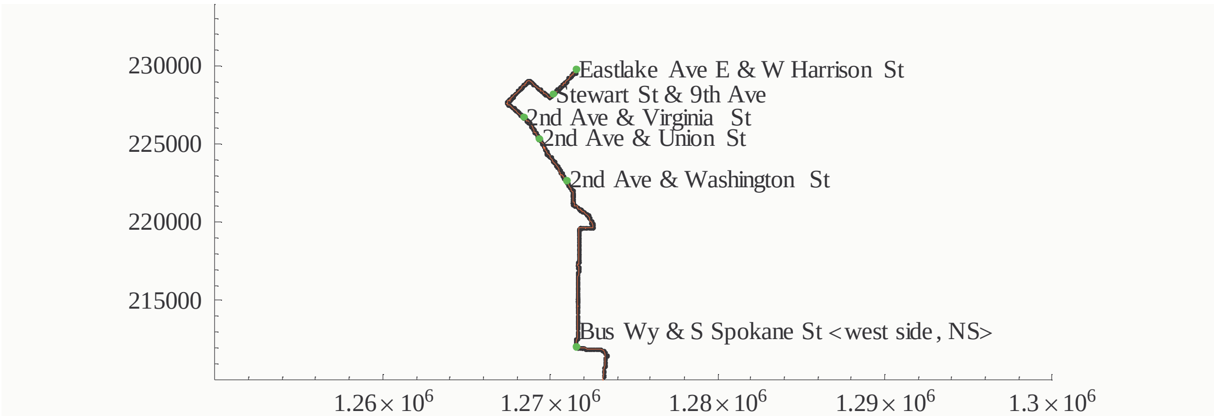

Figure 31: Roadway and GPS reporting – detail in Seattle.
Vehicle 3, PT/ST, Block 222, Trip 1, PM

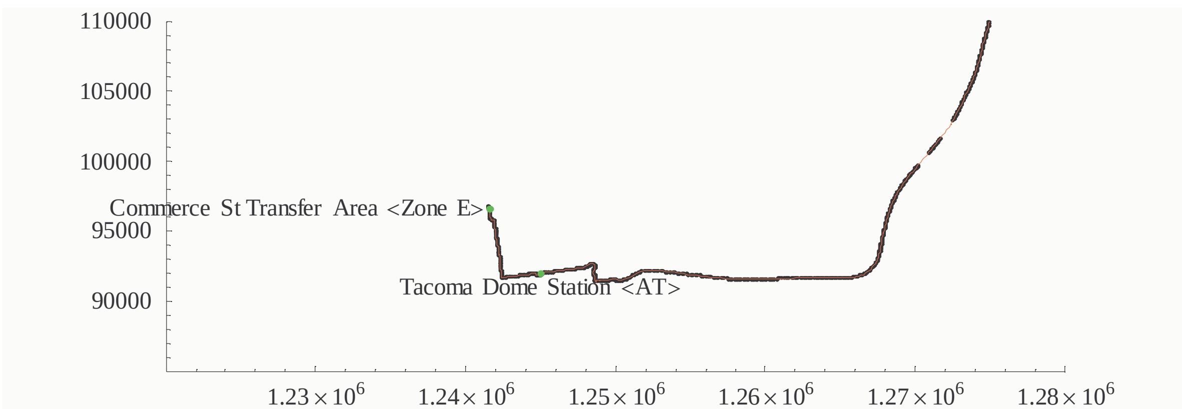

Figure 32: Roadway and GPS reporting – detail in Tacoma.
Vehicle 3, PT/ST, Block 222, Trip 1, PM

5.7.1 On-time Performance

On-time performance in this test has two aspects. The first is the answer to the classic question of did the vehicle depart the time-point at the time specified in the schedule? The second question applicable here, is did the predictions made in advance of the departure at the chosen event accurately reflect the departure time?

We first present the schedule adherence of the vehicles used in the GPS evaluation above, see Table 3. We then present some overall prediction performance figures using a large portion of the total data set recorded during the demonstration.

There are two schedules adherence plots for each of the cases identified in Table 3. The first plot in the pair is a space-time plot that has the schedule as a line and the GPS data from the vehicles as points where a point to the right of the line is a late vehicle and to the left of the line is an early vehicle. The second plot in the pair is a time series of the schedule deviation.

Figure 33 is a time-series plot of distance into trip for vehicle 2 on the AM trips for 4/2/03. Distance is measured in feet and time in minutes after midnight. (The first trip starts 301 minutes after midnight or 5:01 AM.) The red dots correspond to schedule events (time points and schedule times). The blue lines connecting the events represent the scheduled trips. The black line is actually the sequence of pairs (time, distance) where distance was computed by the tracker from vehicle position reports. They are very closely spaced since the vehicle was reporting at a rate of about 1 Hz (once per second). At the top of each curve are two numbers. The top number is the sequence number of the trip and the bottom number is the trip pattern identifier.

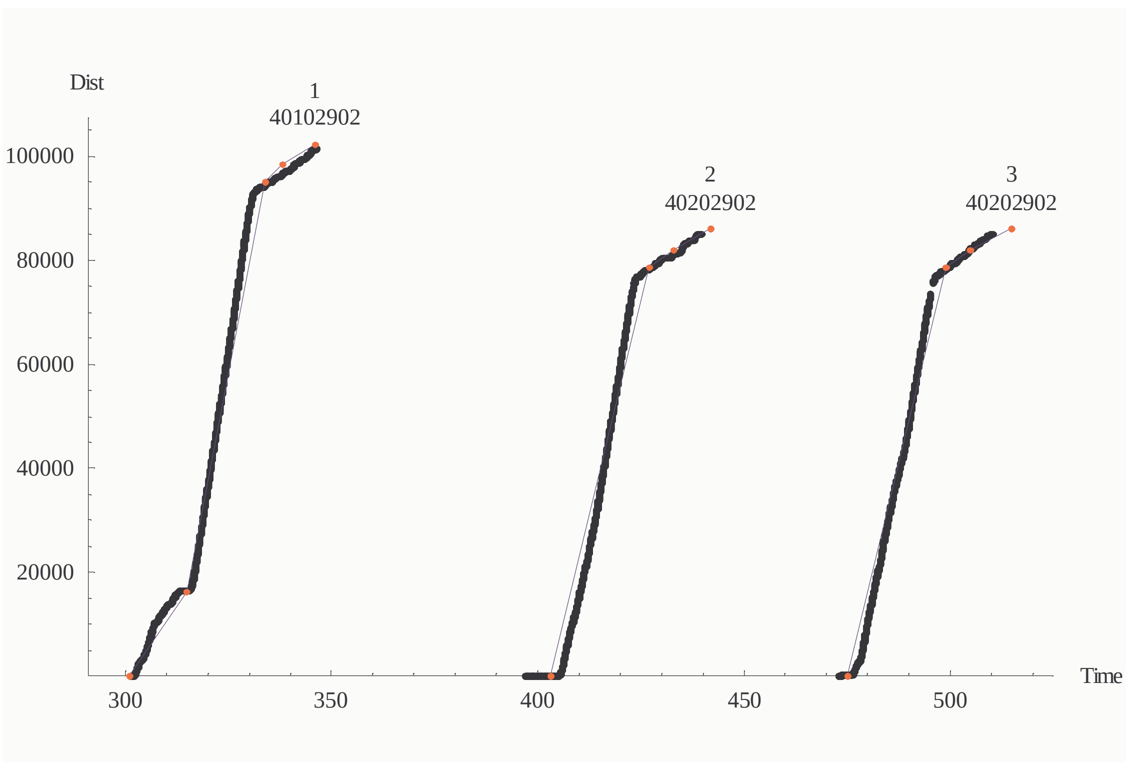

Figure 33: Schedule adherence - line is schedule, points are vehicle location in space and time.

Figure 34 is a time-series plot of interpolated temporal schedule deviation for trip 1. The vehicle starts its trip (164th st SW & Spruce Way West) just a little bit early and departs Lynnwood Park & Ride at 315 (5:15), on time. Similarly, it departs Stewart St & 9th at 334 (5:34), on time. It is then a little bit late for the remaining time points.

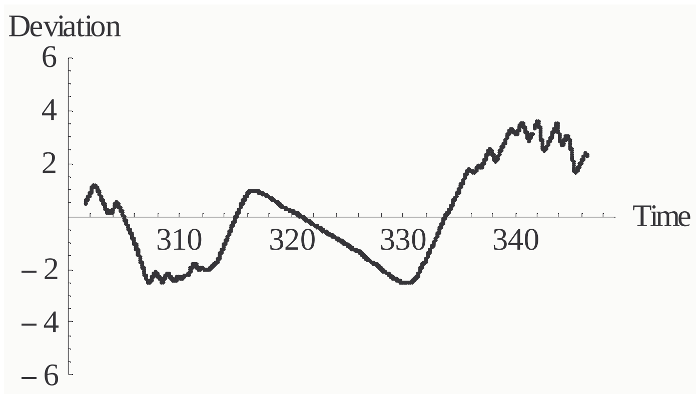

Figure 34: Schedule deviation, trip 1.

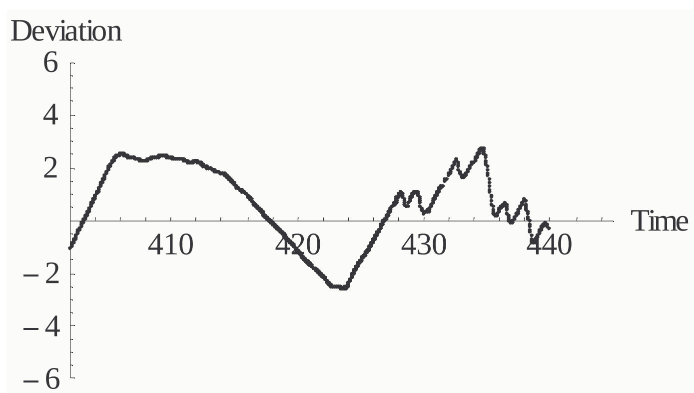

Figure 35: Schedule deviation, trip 2.

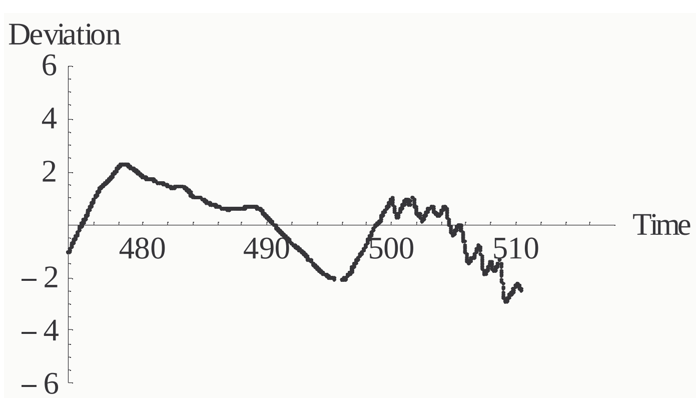

Figure 36: Schedule deviation, trip 3.

Figure 37 shows the PM trip for vehicle 2.

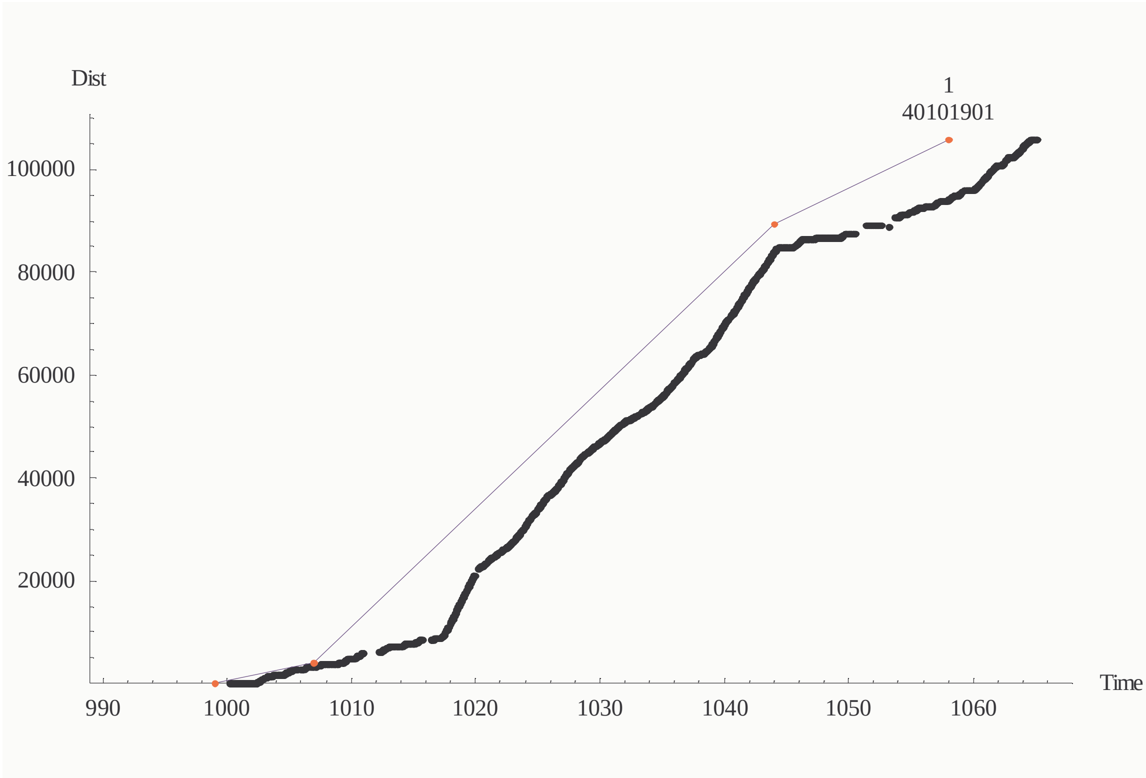

Figure 37: Schedule adherence - line is schedule, points are vehicle location in space and time.

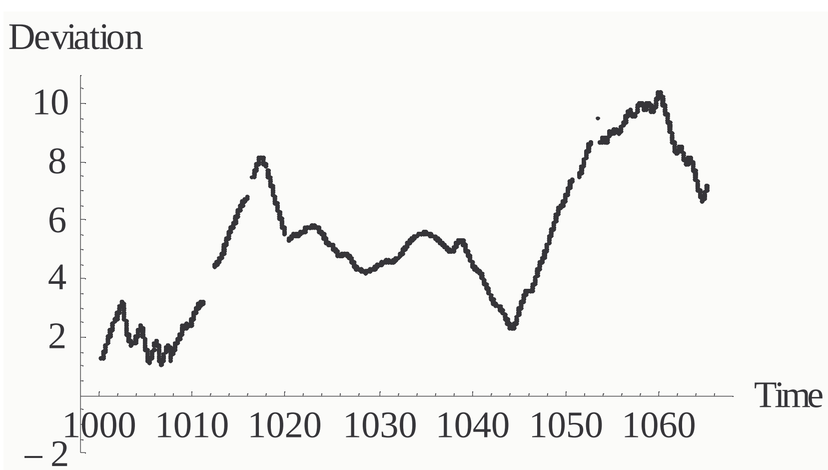

Figure 38: Schedule deviation.

Figure 39 shows the two morning trips for vehicle 4 on block 704 for the same day.

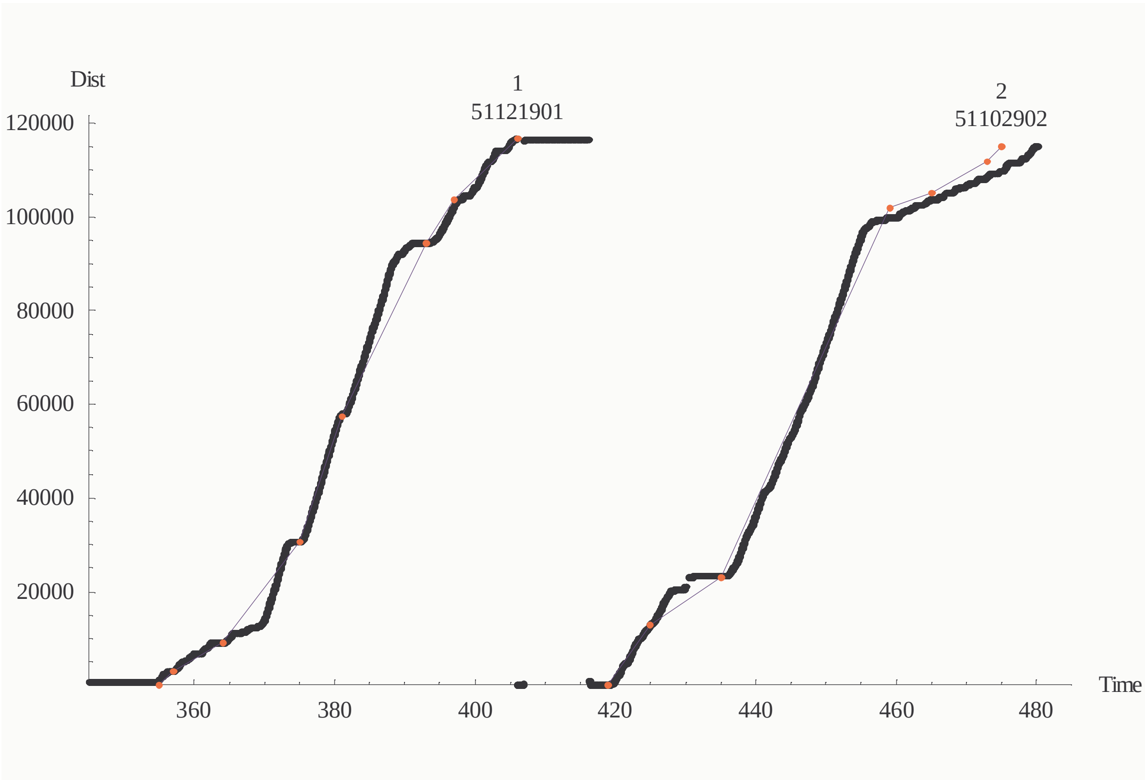

Figure 39: Schedule adherence - line is schedule, points are vehicle location in space and time.

Figure 40 shows the temporal deviations for trip1.

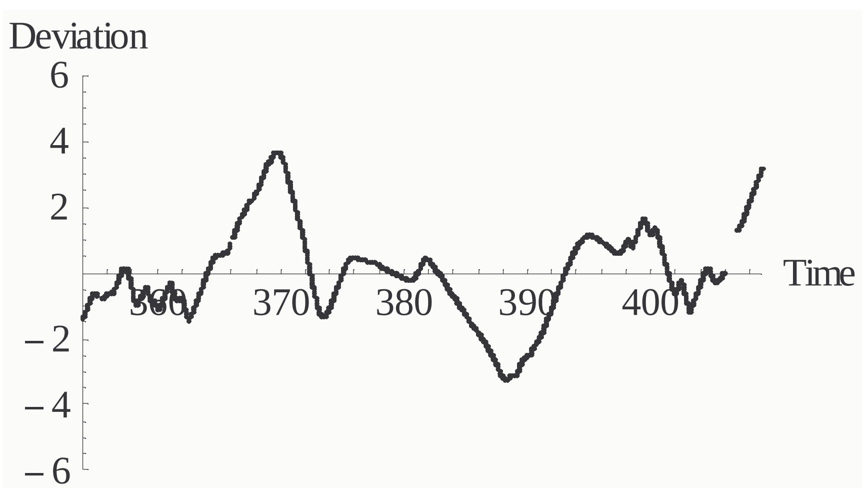

Figure 40: Schedule deviation, trip 1.

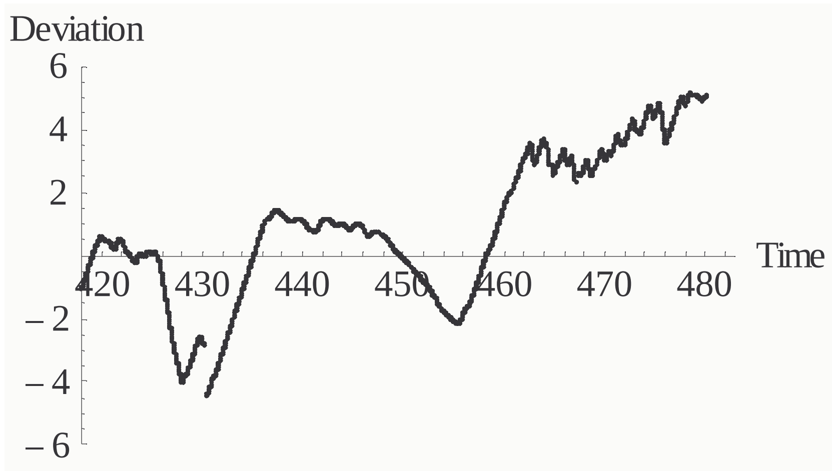

Figure 41: Schedule deviation, trip 2.

Figure 42 shows the two afternoon trips for vehicle 4 on block 704 for the same day.

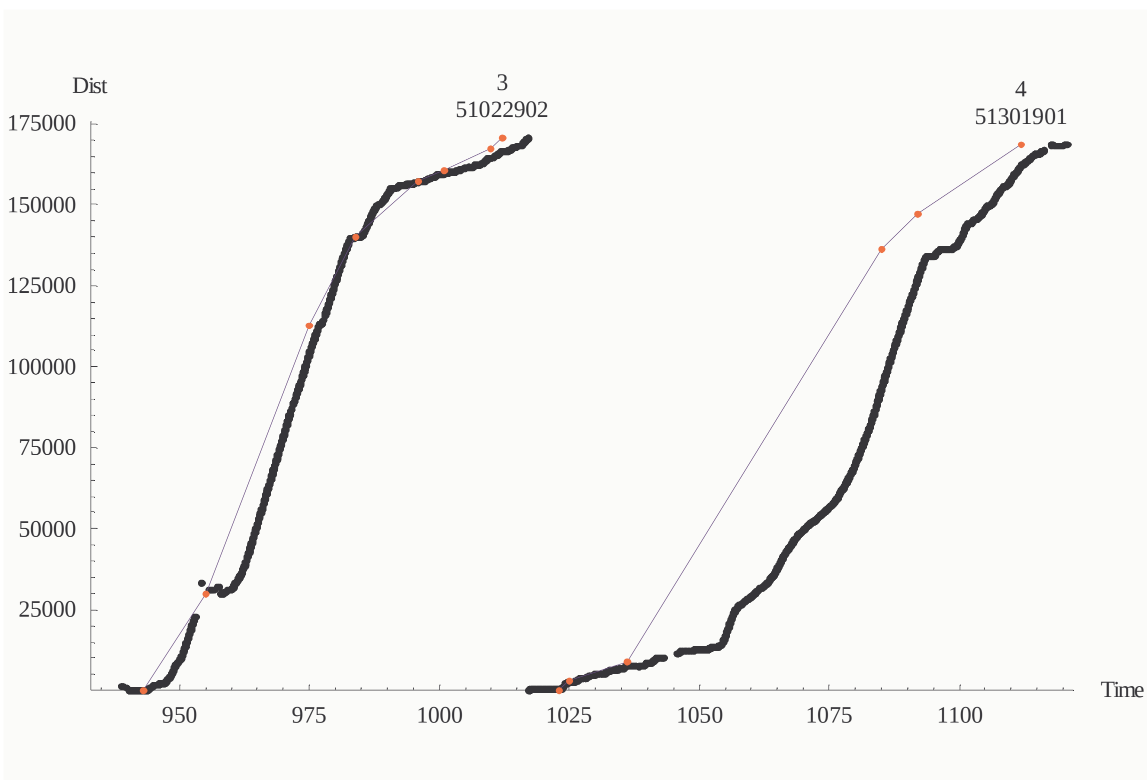

Figure 42: Schedule adherence - line is schedule, points are vehicle location in space and time.

Figure 43 shows the temporal deviations for trip 3. The large delta distance around time 1040 is similar in character to that for vehicle 2 on its afternoon trip at time = 1010.

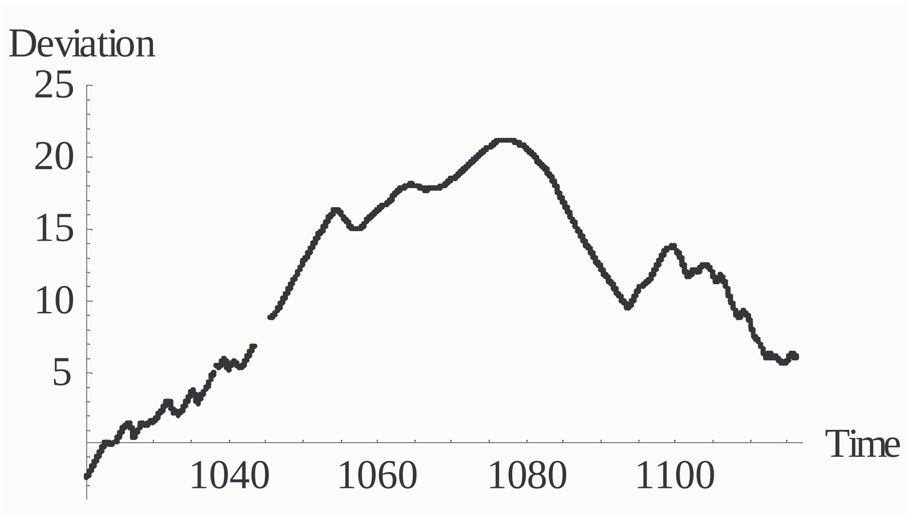

Figure 43: Schedule deviation.

5.7.2 Performance of Predictors

We present the performance of the prediction apparatus for MyBus in a series of plots that display the prediction at times before the arrival at various locations. The plots show the deviation of the prediction from the actual departure time as a function of time. For example, thirty minutes before the vehicle arrives at Lynwood Park and Ride the predictor component estimates it will be 5 minutes late, but the vehicle actually leaves 3 minutes late and so that the prediction error is two minutes.

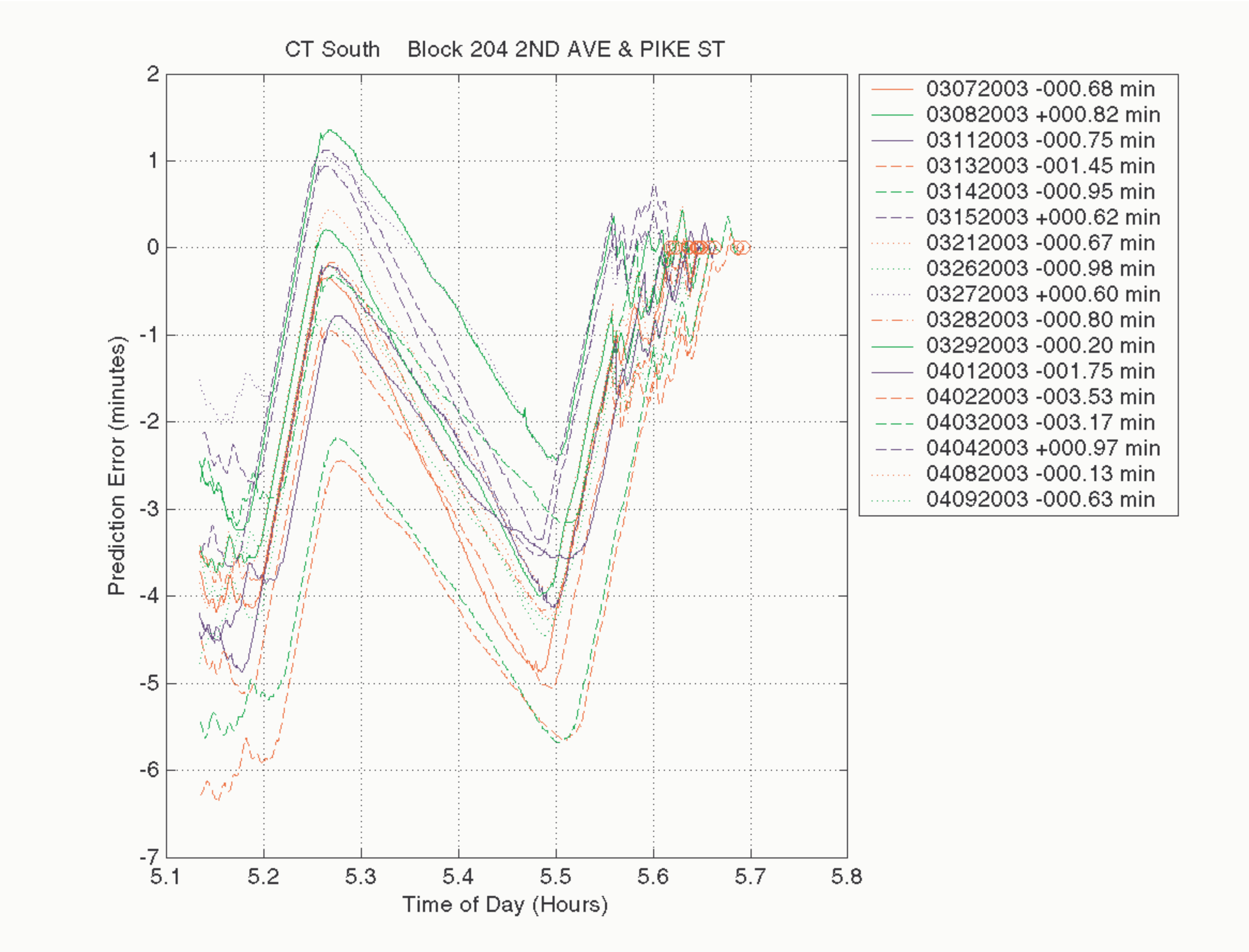

Figure 44: Prediction error as a function of time for Community Transit Block 204, the first line in Table 3.

Date code and actual schedule deviation is shown at the right. Red circle is absolute arrival time. For example, on the 4th of April, 04042003, it arrived 3.17 minutes late.

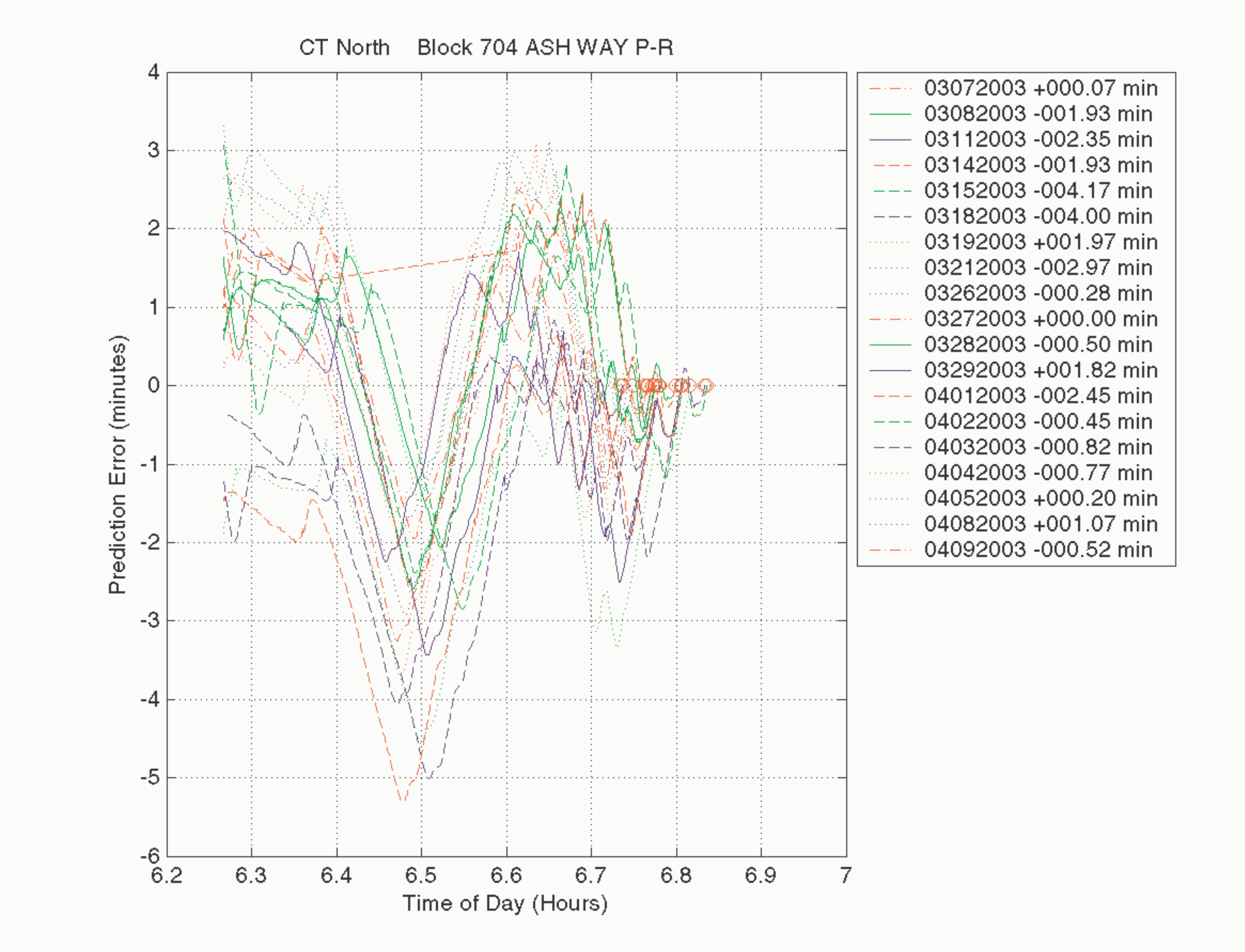

Figure 45: Prediction error for the Ash Way Park and Ride as a function of time for Community Transit Block 704, the third line in Table 3.

Date code and actual schedule deviation is shown at the right. Red circle is absolute arrival time. For example, on the 18th of March, 03182003, it arrived 4 minutes late.

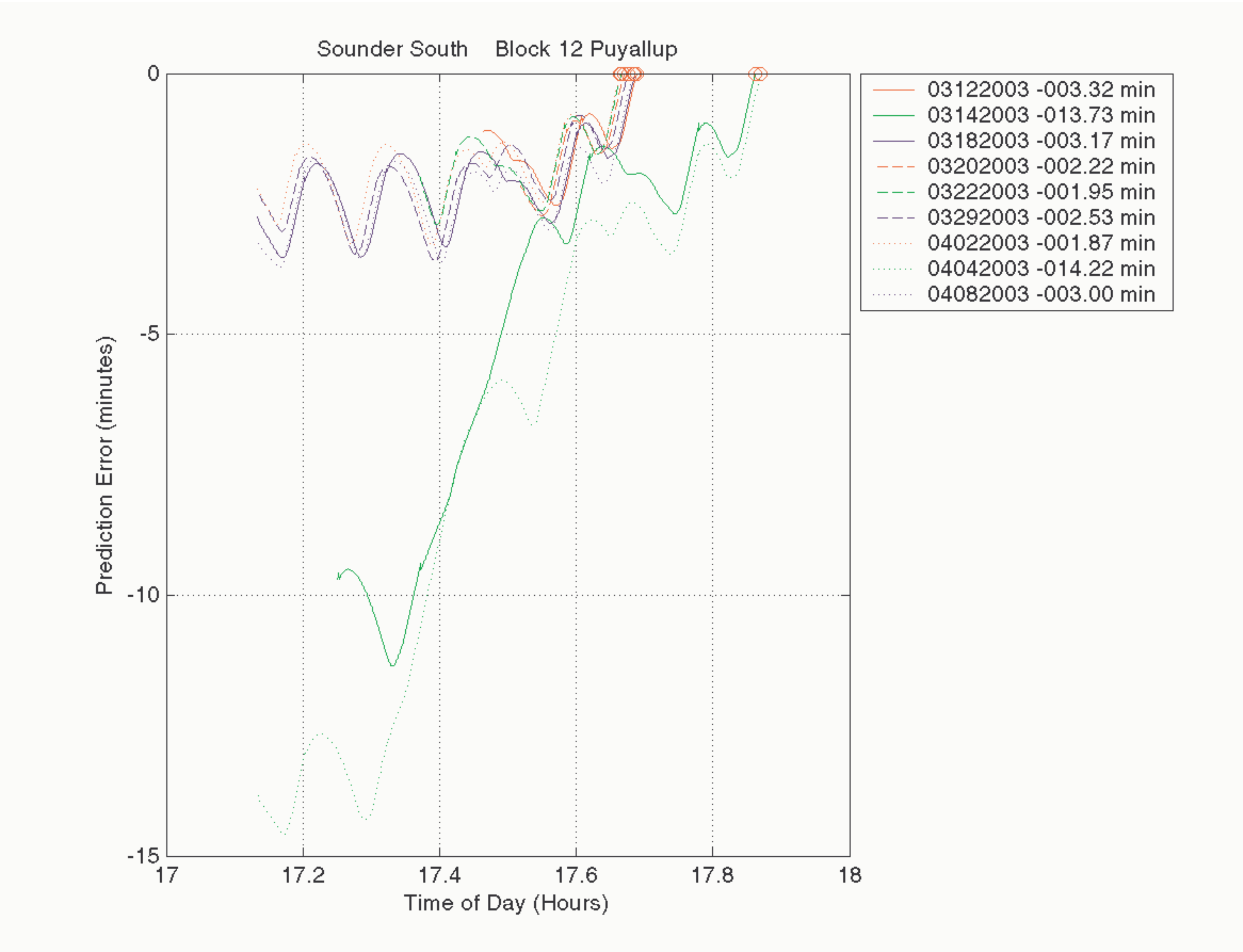

Figure 46: Prediction error as a function of time for the Sounder Block 12.

Date code and actual schedule deviation is shown at the right. Red circle is absolute arrival time. For example, on the 4th of April, 04042003, it arrived 14.22 minutes late.

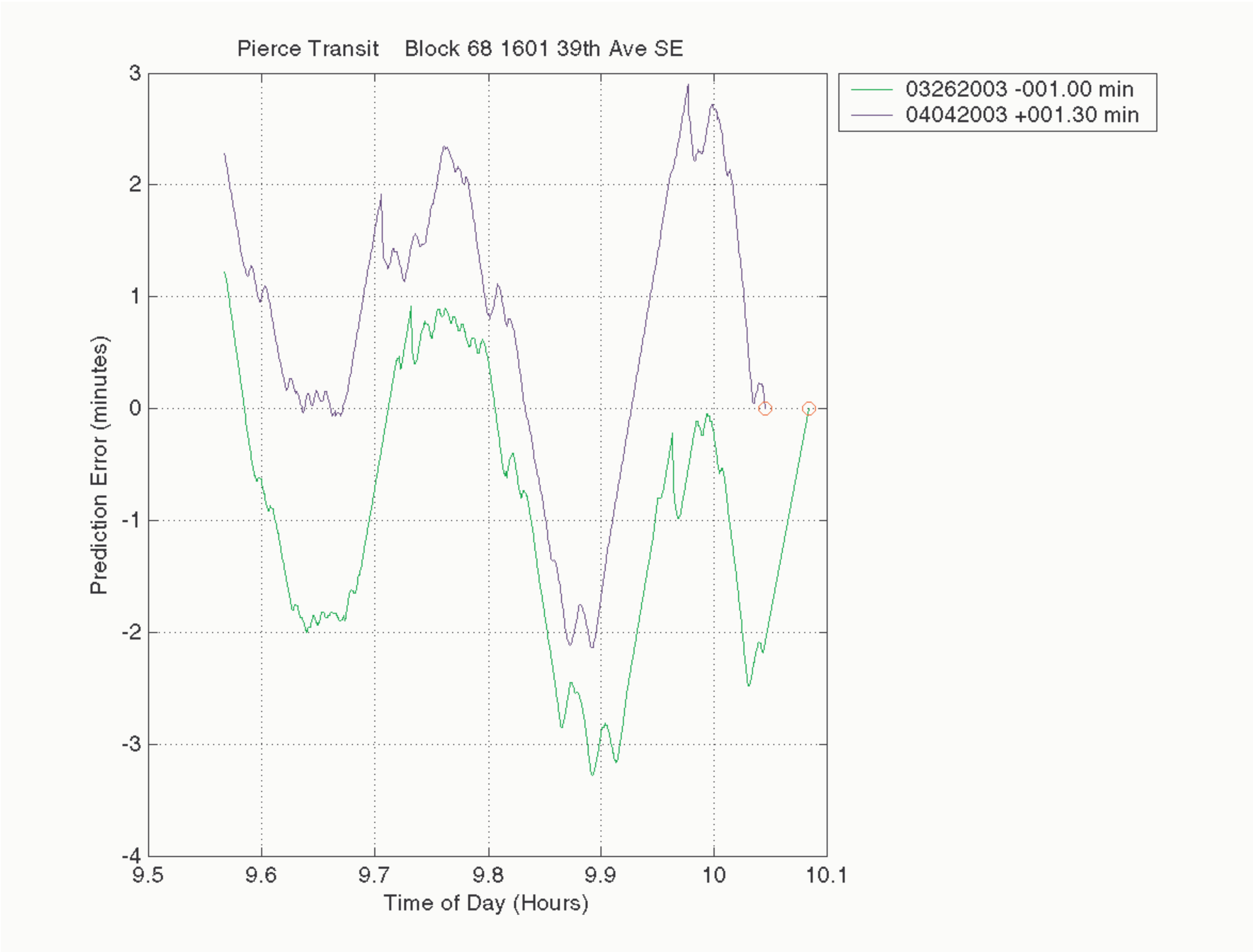

Figure 47: Prediction error as a function of time for Pierce Transit, Block 68, at the 39th Avenue South East timepoint.

Date code and actual schedule deviation is shown at the right. Red circle is absolute arrival time. For example, on the 4th of April, 04042003, it arrived 1.3 minutes early.

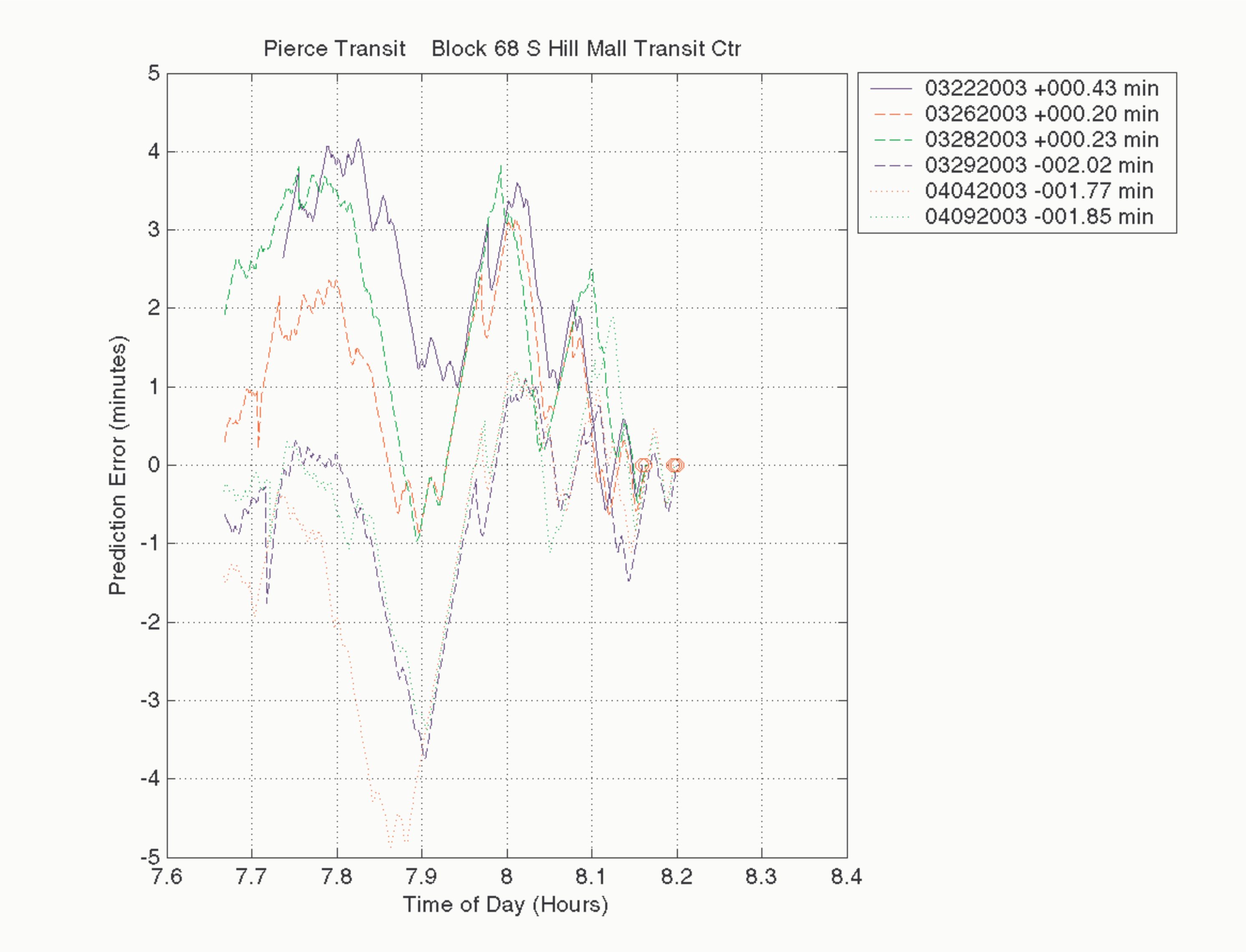

Figure 48: Prediction error as a function of time for Pierse Transit, Block 12, at the South Hills Mall Transit Center.

Date code and actual schedule deviation is shown at the right. Red circle is absolute arrival time. For example, on the 4th of April, 04042003, it arrived 1.77 minutes late.

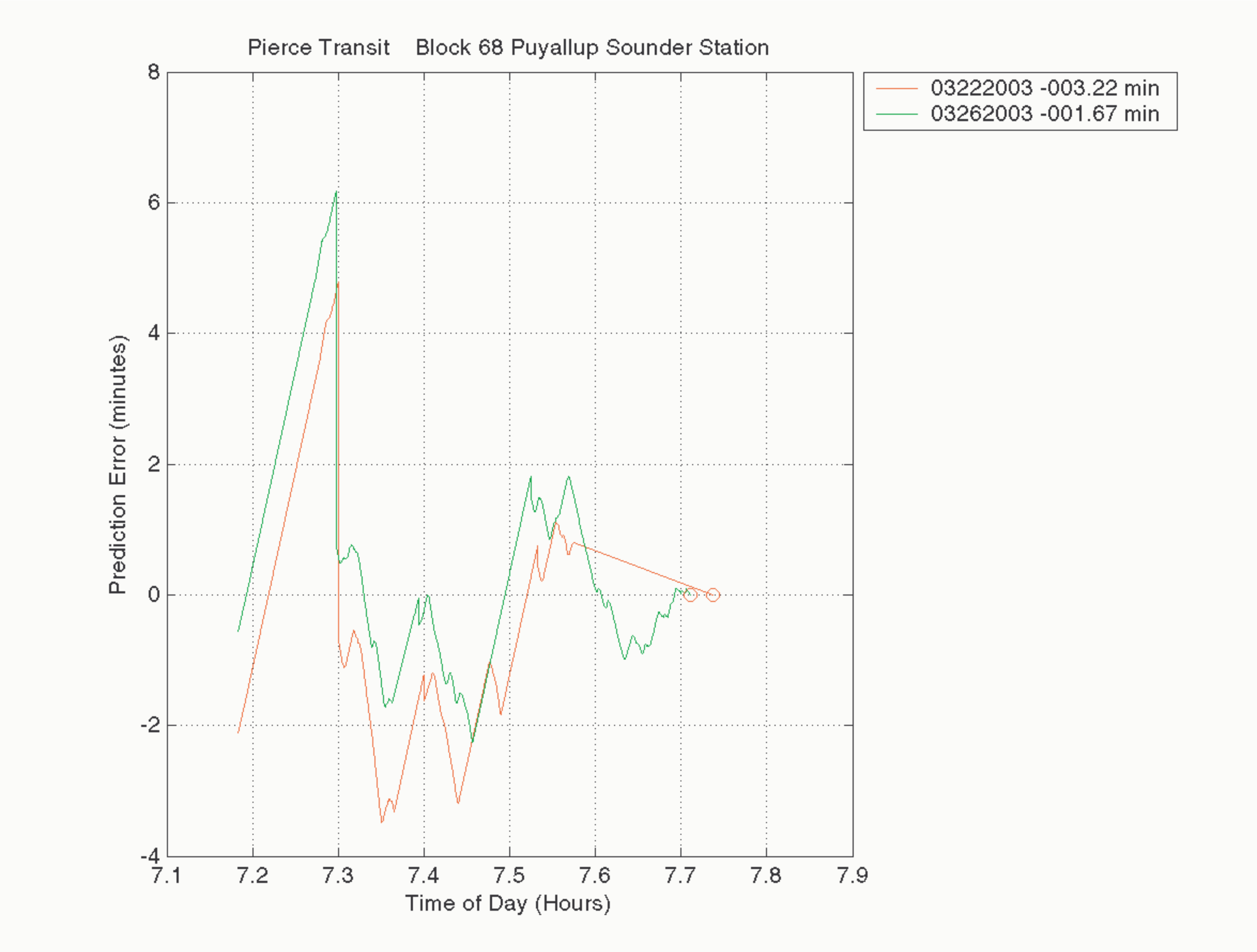

Figure 49: Prediction error as a function of time for Pierse Transit, Block 68, at the South Hills Mall Transit Center.

Date code and actual schedule deviation is shown at the right. Red circle is absolute arrival time. For example, on the 4th of April, 04042003, it arrived 1.77 minutes late.

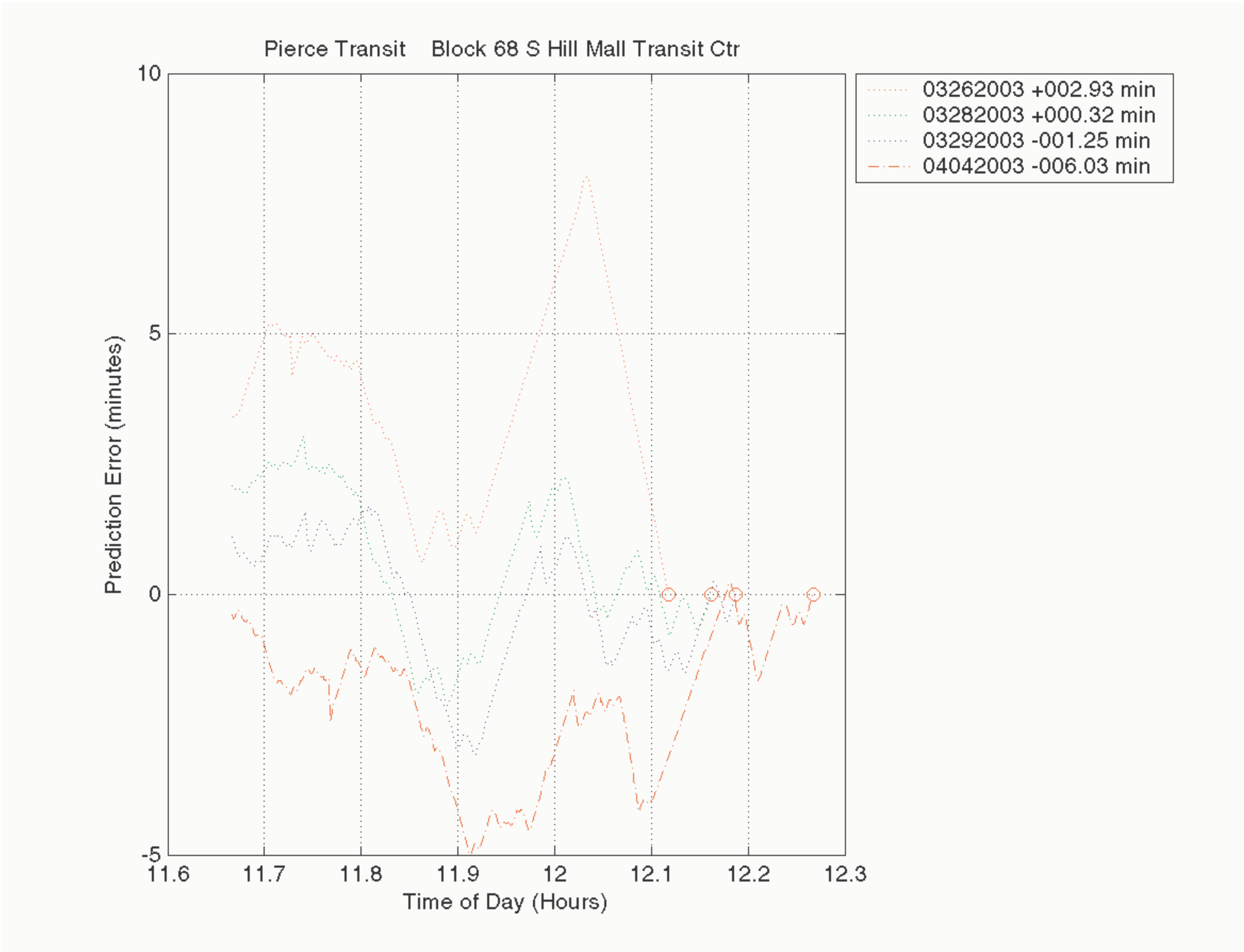

Figure 50: Prediction error as a function of time for Pierse Transit, Block 68, at the South Hills Mall Transit Center.

Date code and actual schedule deviation is shown at the right. Red circle is absolute arrival time. For example, on the 4th of April, 04042003, it arrived 6.03 minutes late.

DRAFT

5.8 Analysis of Sounder and Pierce Service Interaction at Puyallup Station

The transit vehicles travel in both space and time. An example of the spatial trajectories for the Sounder train and the Pierce 410/411 bus service is shown in Figure 51. The Sounder trajectory is at the top, and the Pierce service meets the train service at the Puyallup station, which is at location (0,0). In this project, only one train, on one trip north in the morning and one trip south in the afternoon, is actually tracked. However, the Pierce 410/411 service operates all day.

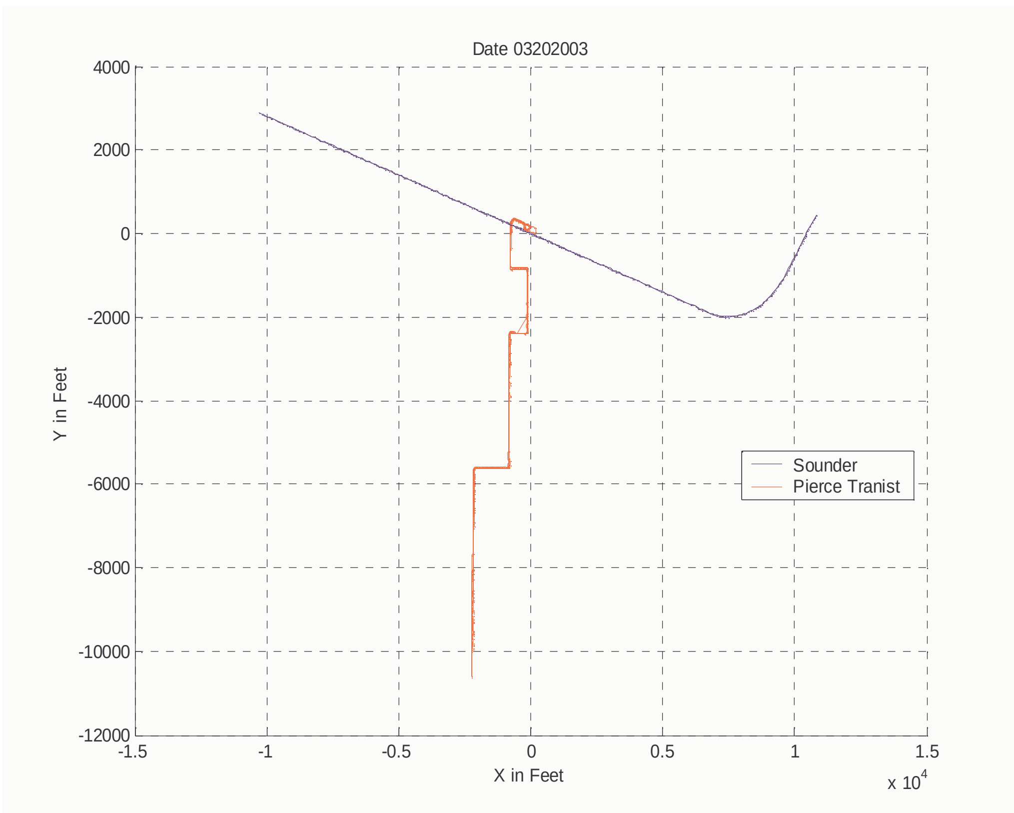

Figure 51: Spatial trajectories for Sounder and Pierce 410/411 at Puyallup station.

Figure 52 displays the train and bus service with time in the vertical dimension so that the sequential bus trips are shown over the course of the day. The train is shown traveling past the Puyallup station once in the morning and once in the afternoon.

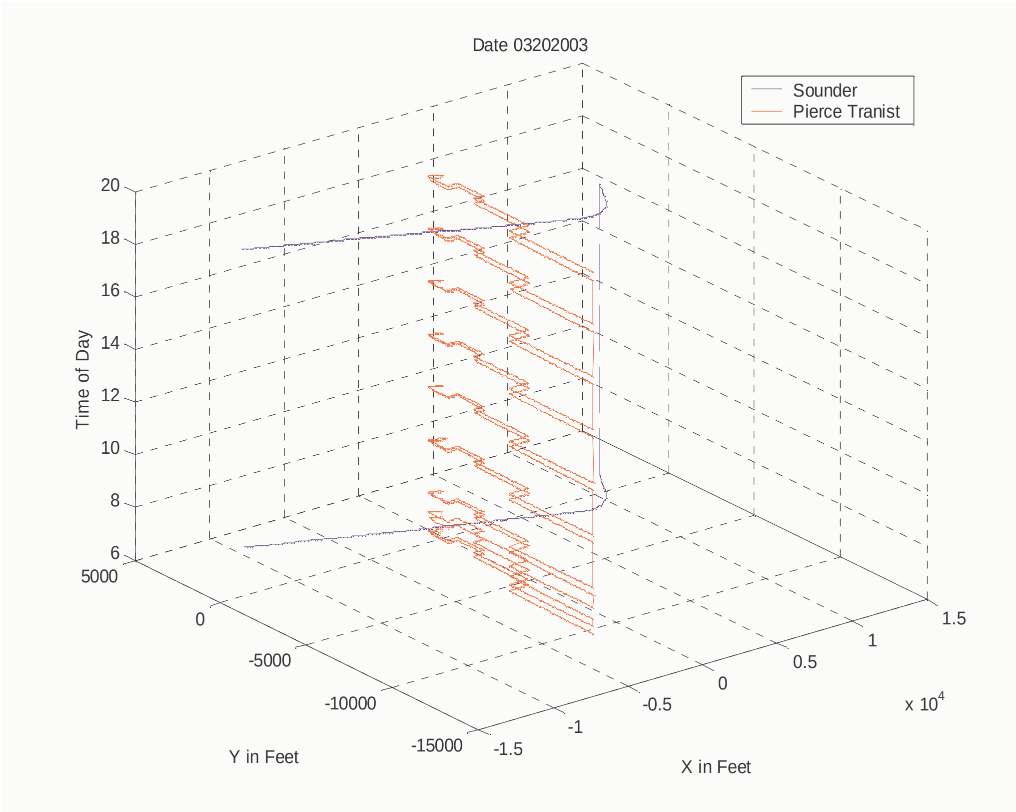

Figure 52: Sounder and 410/411 services over the course of the day, with time shown in the vertical dimension.

This behavior is observed over the life of the project and is shown for the days of March 12, 14, 15, 19, 20, 21, 22 and 25 in Figures 53-60. From these figures, it is somewhat difficult to identify the relationship between the two scheduled services.

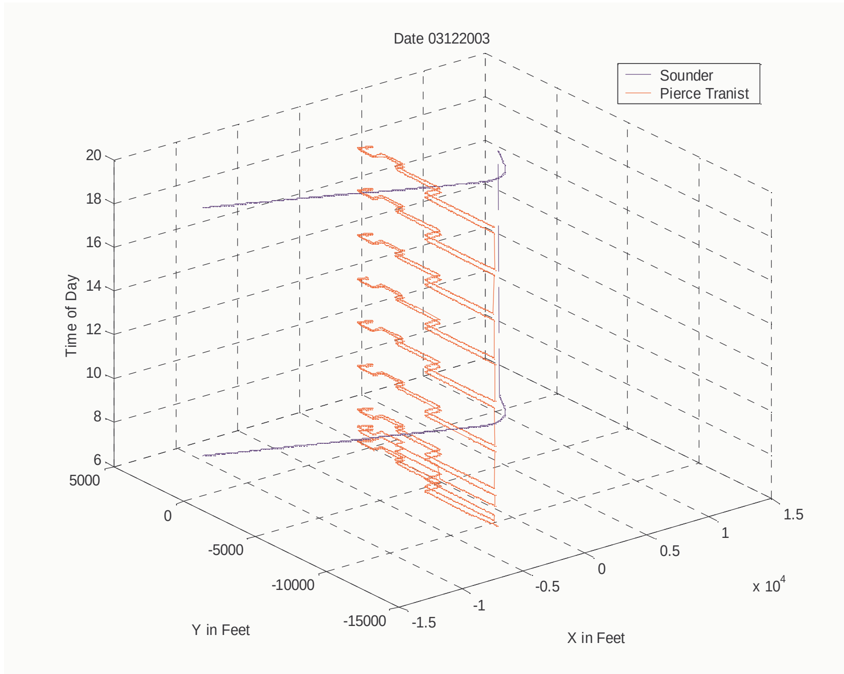

Figure 53: Sounder and 410/411 services over the course of the day, March 12, 2003.

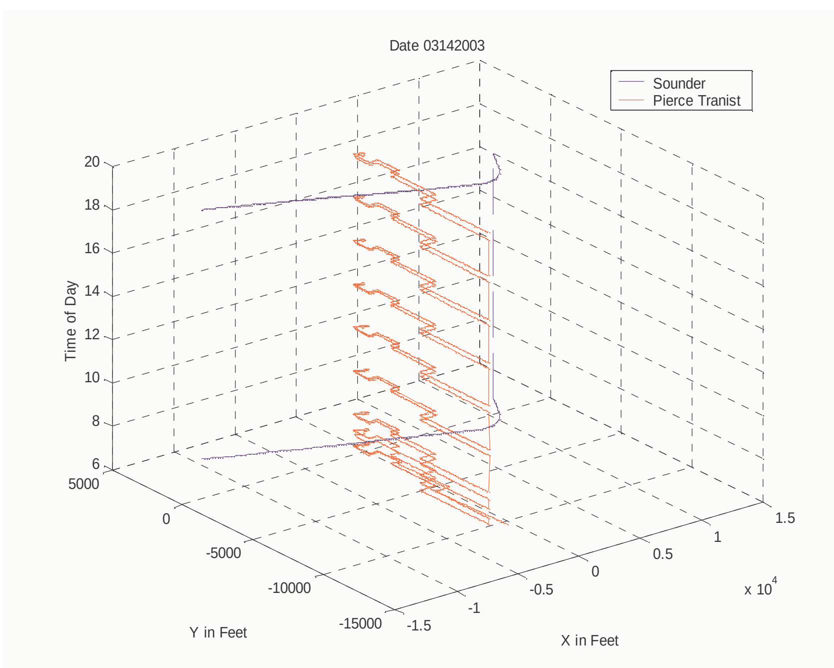

Figure 54: Sounder and 410/411 services over the course of the day, March 14, 2003.

DRAFT

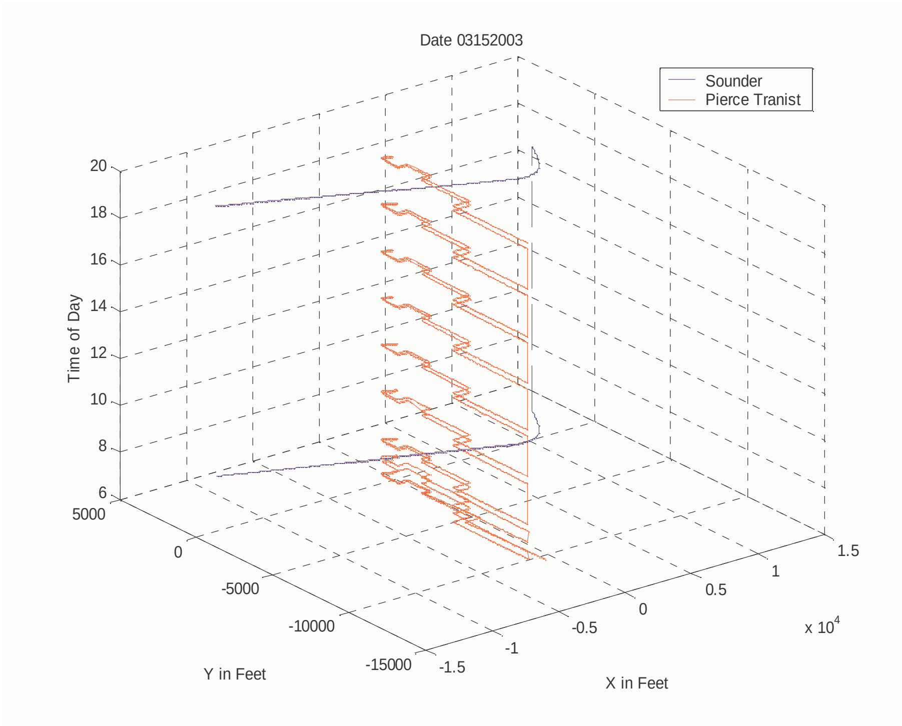

Figure 55: Sounder and 410/411 services over the course of the day, March 15, 2003.

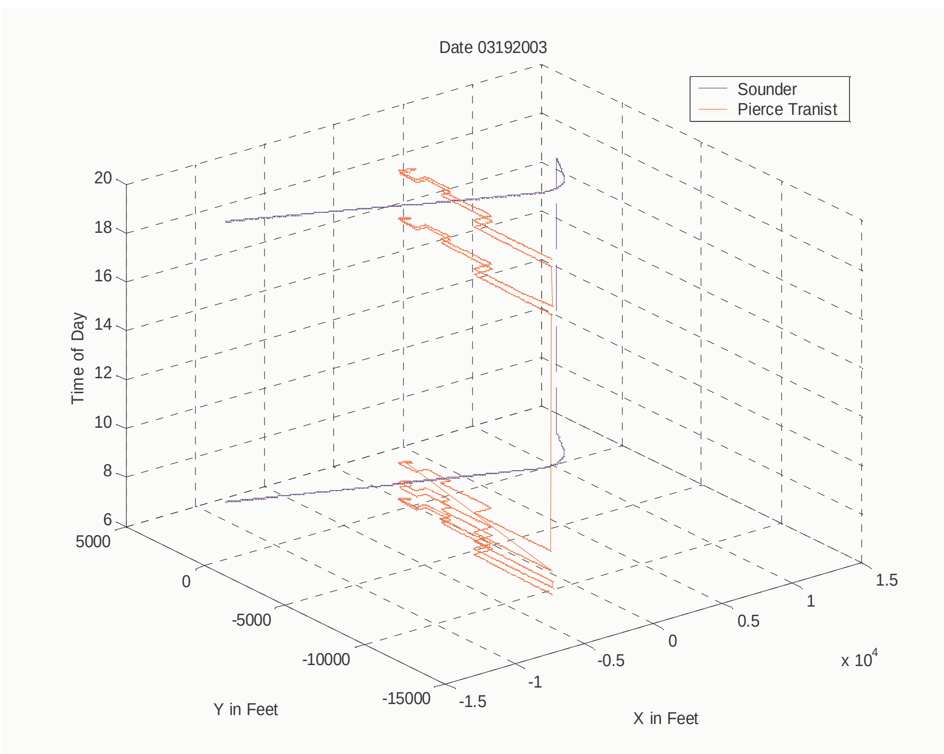

Figure 56: Sounder and 410/411 services over the course of the day, March 19, 2003.

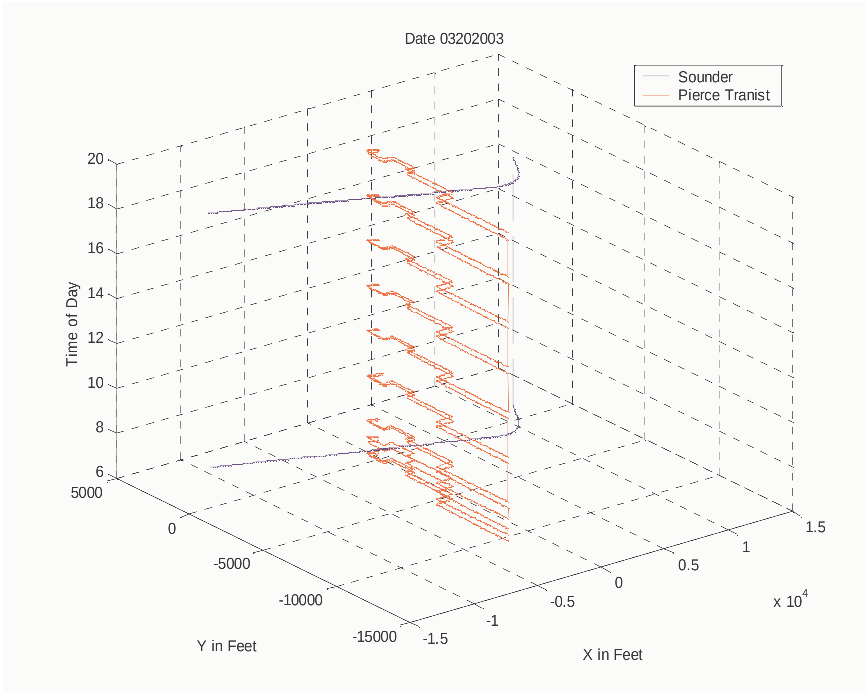

Figure 57: Sounder and 410/411 services over the course of the day, March 20, 2003.

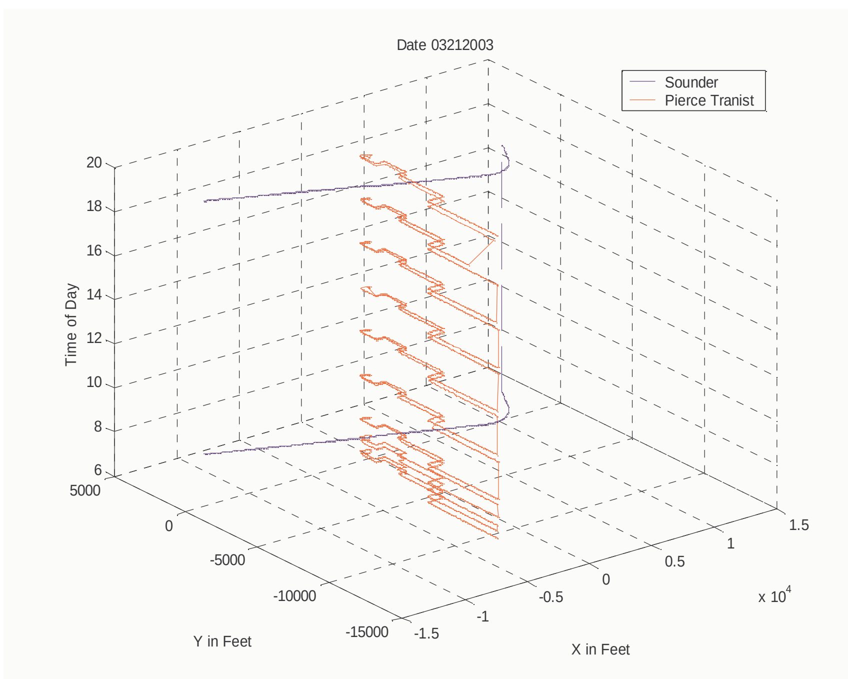

Figure 58: Sounder and 410/411 services over the course of the day, March 21, 2003.

DRAFT

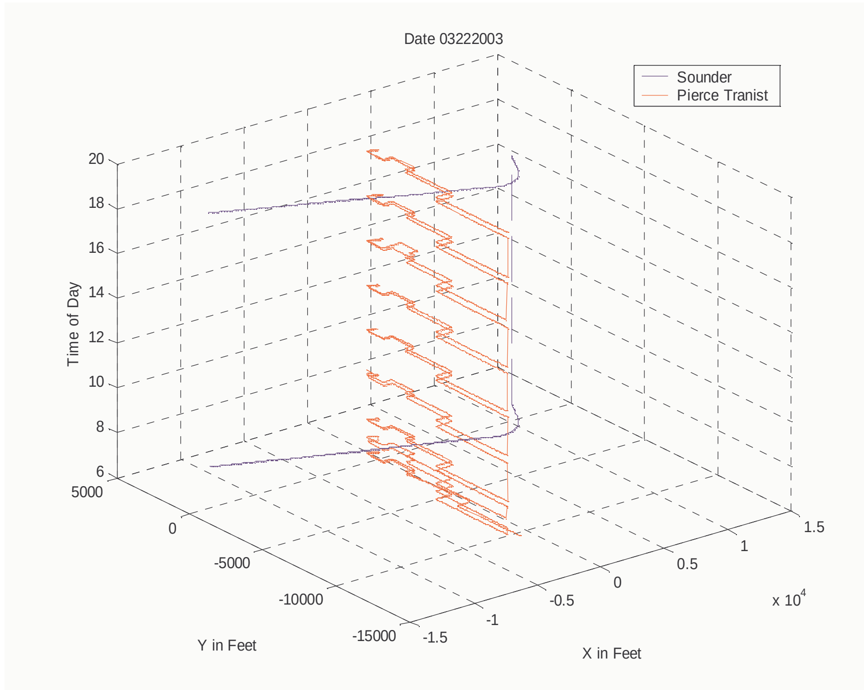

Figure 59: Sounder and 410/411 services over the course of the day, March 22, 2003.

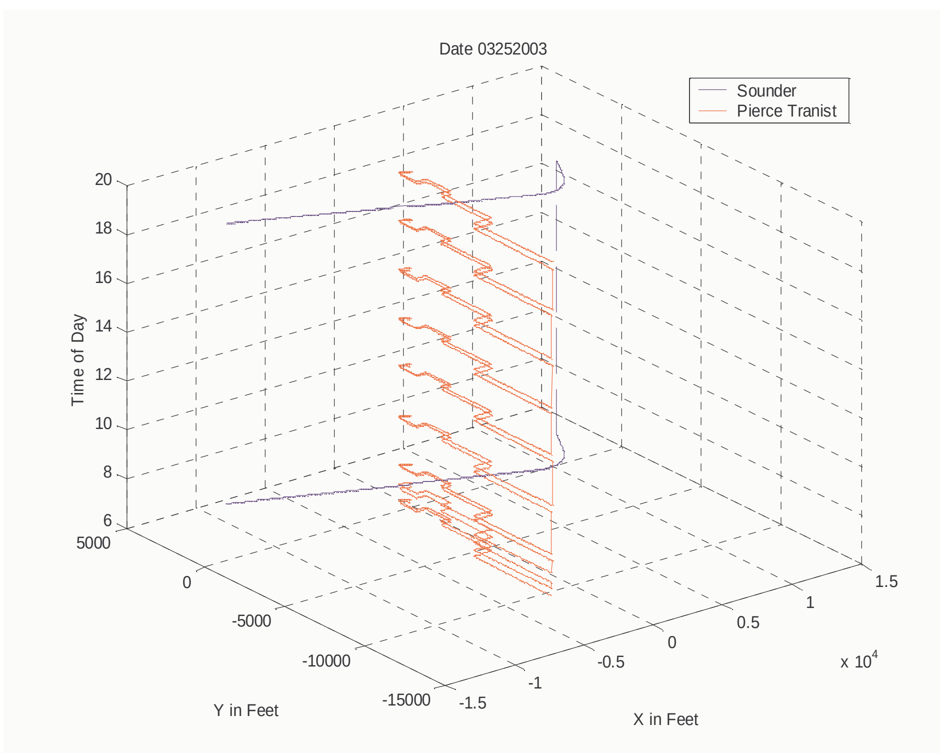

Figure 60: Sounder and 410/411 services over the course of the day, March 25, 2003.

DRAFT

To examine the relationship between the multiple modes in detail, the distance from the Puyallup station is plotted verses time for both agencies over the course of the day. Figure 61 shows the Sounder (in dashed blue lines) and the 410/411 service (in solid red lines) for March 12, 2003. The vehicles arrive at the Puyallup station at the minima in the trajectories, and the scheduled arrival times for the Sounder at the Puyallup station are shown as tick marks on the horizontal time axis. On this day, the Sounder was recorded for the first train of the day and arrived nearly on time. Passengers trying to use the multi modal service toward Seattle would have needed to catch the first run of the 410/411 in order to catch the Sounder, and those traveling from Tacoma, and connecting to the bus service in Puyallup, would need to wait about 30 minutes after leaving the train for the next bus to arrive. In the afternoon, the Sounder arrives on time, and the eighth trip in the 410/411 bus service meets the train and departs shortly afterward.

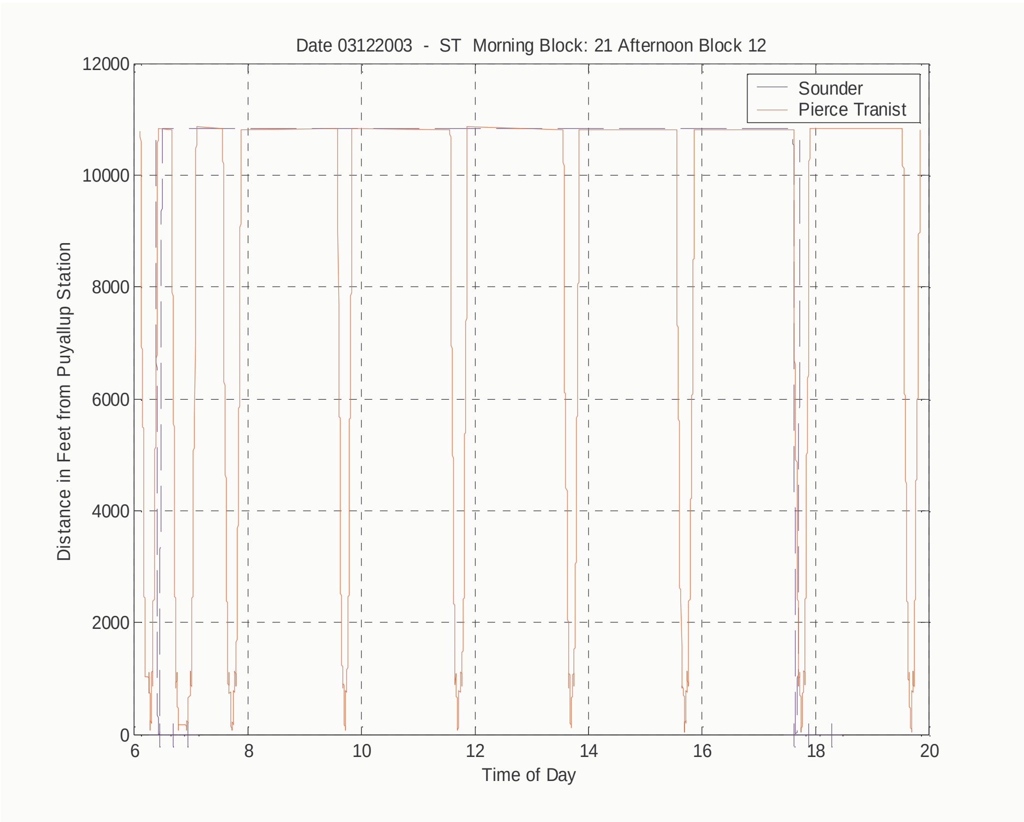

Figure 61: Distance from Puyallup station for Sounder and 410/411 over the course of the day, March 12, 2003.

The passenger car the GPS unit was attached to was associated with different trains on sequential days. The interaction between the transit service provided by the train and the bus that had the GPS unit on that day are shown for the days of March 14, 15, 19, 20, 21, 22 and 25 in Figures 62-68.

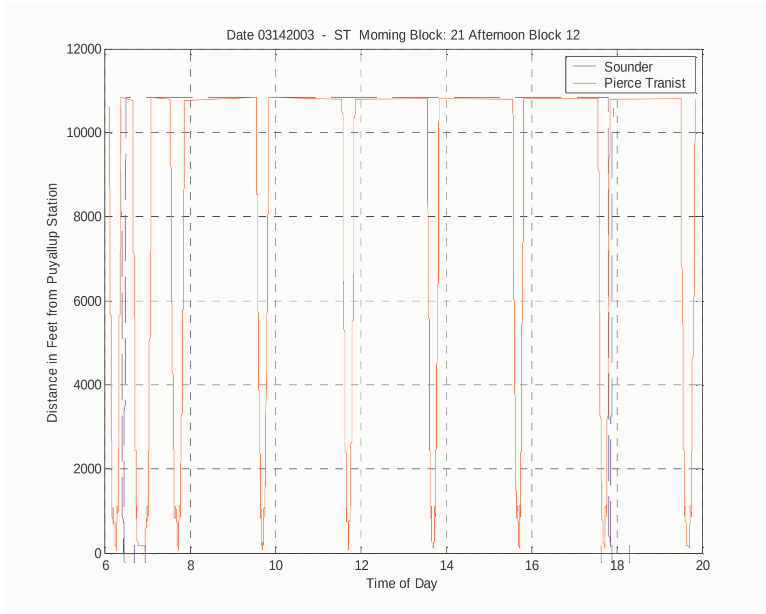

Figure 62: Distance from Puyallup station for Sounder and 410/411 over the course of the day, March 14, 2003.

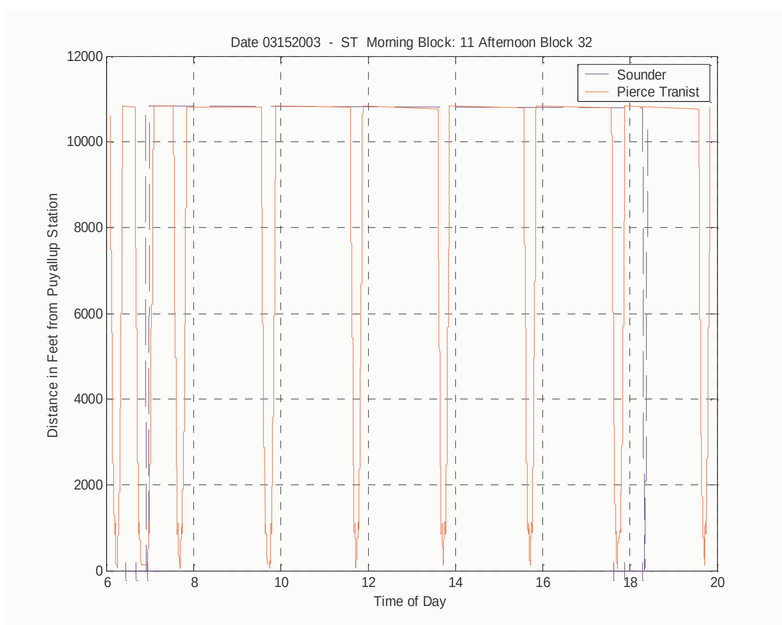

Figure 63: Distance from Puyallup station for Sounder and 410/411 over the course of the day,
March 15, 2003.

DRAFT

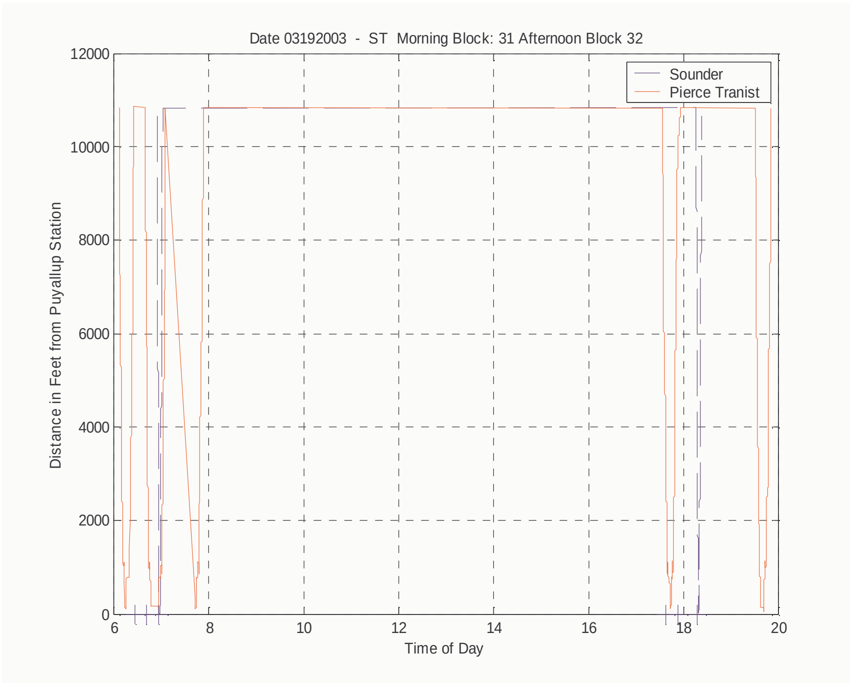

Figure 64: Distance from Puyallup station for Sounder and 410/411 over the course of the day, March 19, 2003.

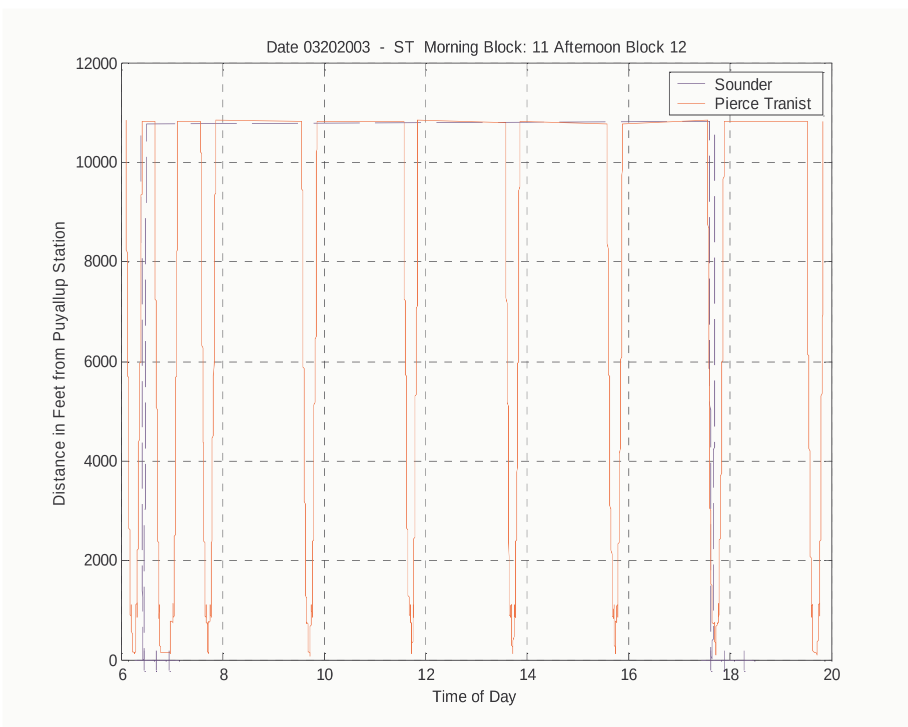

Figure 65: Distance from Puyallup station for Sounder and 410/411 over the course of the day,
March 20, 2003.

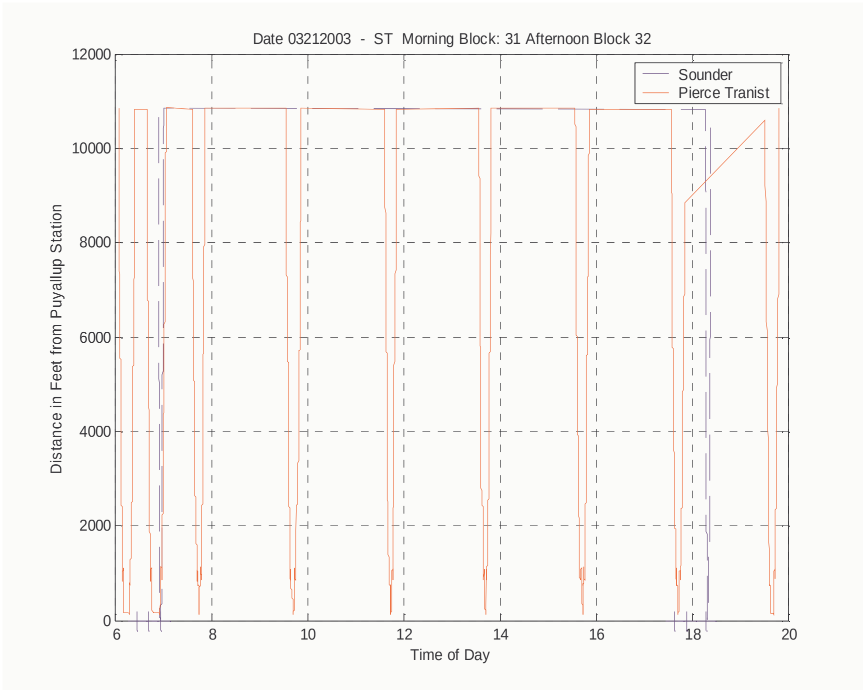

Figure 66: Distance from Puyallup station for Sounder and 410/411 over the course of the day, March 21, 2003.

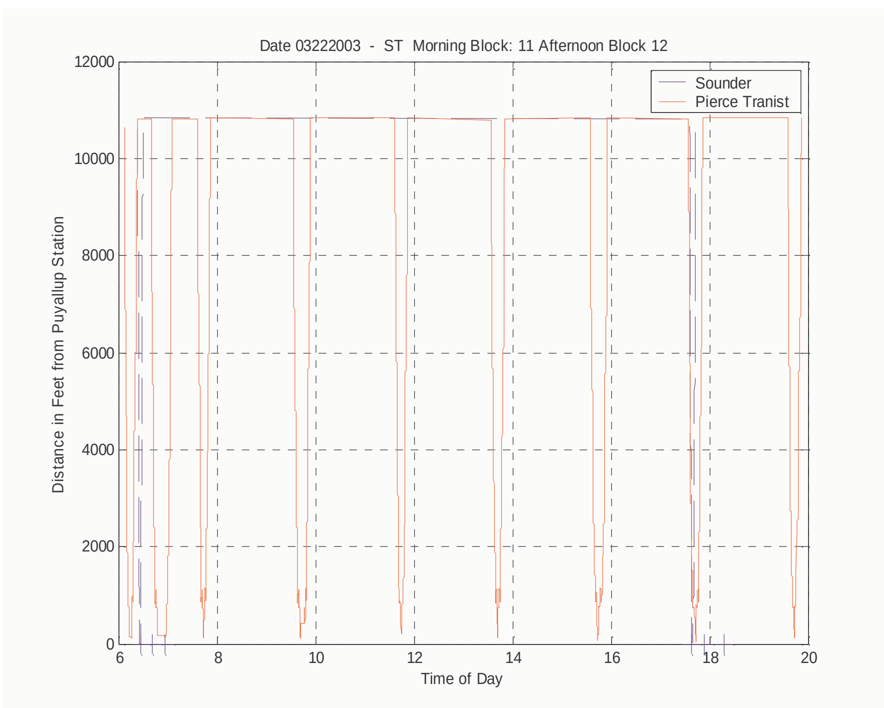

Figure 67: Distance from Puyallup station for Sounder and 410/411 over the course of the day, March 22, 2003.

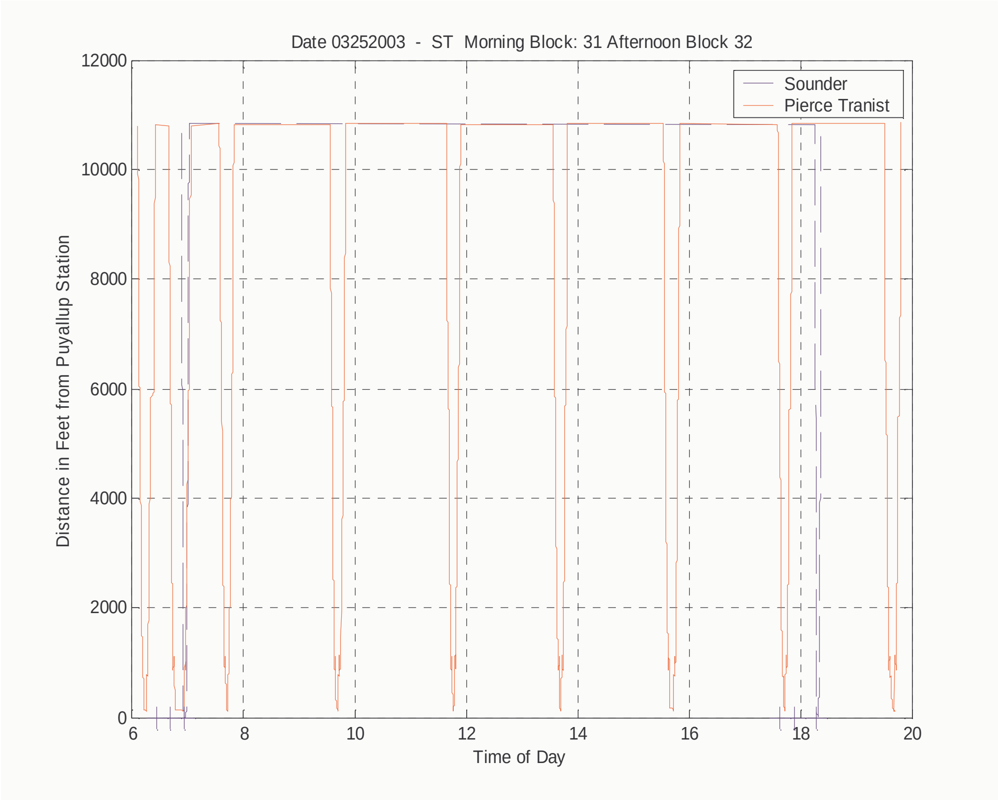

Figure 68: Distance from Puyallup station for Sounder and 410/411 over the course of the day, March 25, 2003.

Figure 69 shows another detailed example of the relationship between the multiple modes. The proximity of the 410/411 bus to the sounder station is shown by the solid red lines with the layover at the station clearly seen at the minima of each of the curves. On March 21, 2003, the GPS-equipped passenger car serviced the third train of the day, shown as a dashed blue line in Figure 69, and interacted with the second trip of the morning 410/411 service. The bus arrives at the station a few minutes before the third train of the day leaves the station. It is noteworthy that the bus service precedes the scheduled departure of the first train by some minutes. If all the vehicles were GPS equipped, it would be possible to use results like those shown in Figure 69 to adjust schedules to improve the multi-modal service.

DRAFT

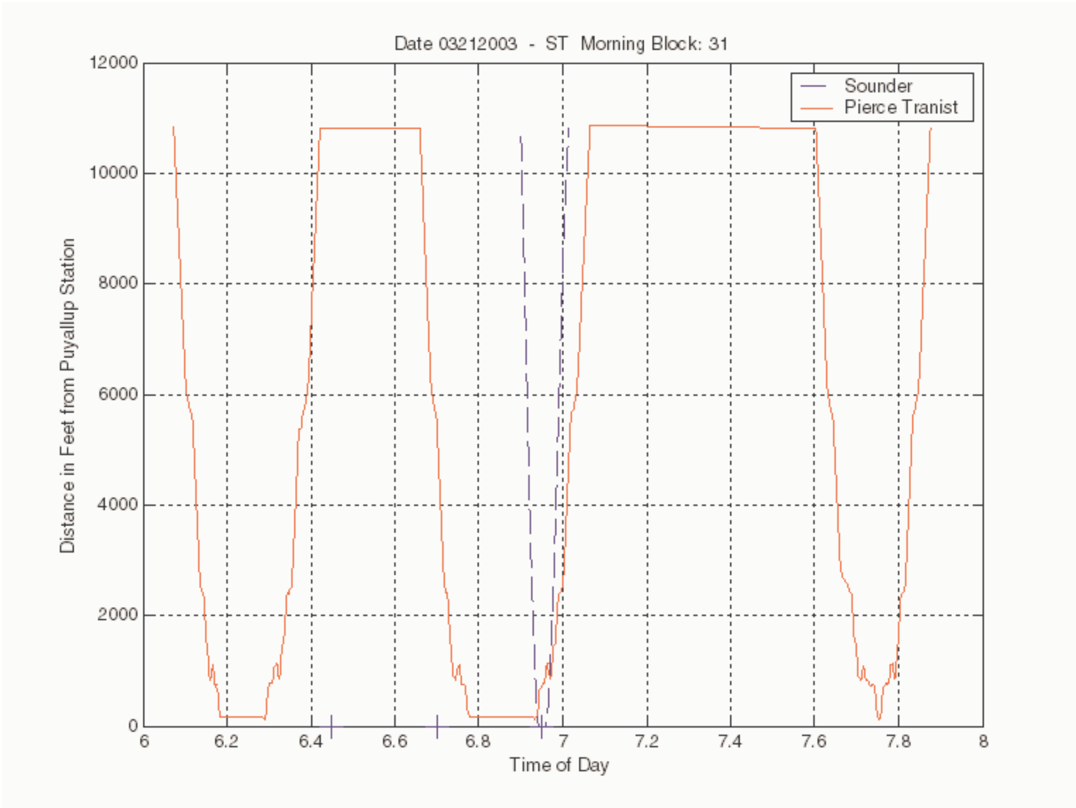

Figure 69: Distance from Puyallup station for Sounder and 410/411 for morning trips, March 21, 2003.

Recording the tracking data, like that just presented, for all of the service from all of the agencies would allow for insights into the changes in the schedule structure that would provide better connections for the multi-modal traveler. In this project, the uncertainty associated with the deployment of the GPS set on one of three trains complicates the analysis and any operational system should have GPS sets on all the trains.

5.9 Technical Conclusions

Creating a multi-modal version of our publicly available transit information applications is both technically and politically challenging. The four agencies who provide data to this effort are at very different stages in the development of the use of real-time data. Obtaining usable schedule information, both spatial and temporal, can be a challenging activity in such an environment. However, in a relatively short time we constructed multi-modal applications useful to all four agencies. This was facilitated by the pre-existing component architecture that allowed us to reuse components and plug together new applications incrementally.

The GPS-based tracking implementation proved to be effective in most cases, with the exception of some downtown Seattle locations. Since most of Sound Transit's deployed service operates in areas outside the downtown core, a GPS-based AVL solution would work for most service. In the case of downtown service, the addition of dead reckoning equipment would improve the position estimates.

Coordination between the spatial and temporal schedule efforts and any future AVL effort will need to improve if the AVL system is to be successful. For example, in this project we created much of the spatial information for the Pierce Transit schedule using the GPS information from the vehicles. A second example is the trip assignment for the Sounder train. The trip assignment of the car that had the GPS receiver mounted on it was not known in advance, and we were forced to guess the trip assignment, not always successfully. This can easily be remedied by both coordination with the train operations and the addition of GPS receivers to all trains.

References

[1] University of Washington ITS Research Program, “Busview,” http://www.its.washington.edu/projects/busview overview.html.

[2] “Mybus.org: Bus Arrival Predictions web application,” http://www.mybus.org.

[3] “Transit Vehicles as Traffic Probes,” http://www.its.washington.edu/transit-probes/.

[4] D.J. Dailey, M.P. Haselkorn, and D. Meyers, “A Structured Approach to Developing Real-Time, Distributed Network Applications for ITS Deployment,” ITS Journal, vol. 3, no. 3, pp. 163—180, 1996.

[5] D.J. Dailey, S. Mclean, F.W. Cathey, and D. Meyers, “Self Describing Data Transfer Model in Intelligent Transportation Systems Applications,” IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 4, pp. 293—300, 2002.

[6] D.J. Dailey, S.D. Mclean, F.W. Cathey, and Z. Wall, “Transit Vehicle Arrival Prediction: An Algorithm and a Large Scale Implementation,” in Proceedings of the Transportation Research Board Annual Meeting, Jan. 2001.

[7] “ITS Backbone toolkit,” http://www.its.washington.edu/backbone/.

[8] F.W. Cathey and D.J. Dailey, “A Prescription for Transit Arrival/Departure Prediction using Automatic Vehicle Location Data,” Transportation Research C, 2003.

[9] J. E. Stern, State Plane Coordinate System of 1983, NOAA Manual NOS NGS 5, January 1989.

[10] Institute of Transportation Engineers, “Transit Communications Interface Profiles,” http://www.tcip.org, accessed Feb.

Appendix A: Standard Transit Schema

This document describes the tables used by ITS-UW to represent transit agency data. This schema is not part of TCIP, but it is influenced by TCIP. The concepts and terms used come from TCIP whenever possible.

Constants tables

event_type_constants
Referenced by: time_point_events, stop_point_events
Identifies: an event as one of four types:

- passing - the bus goes through a point without a planned layover
- arrival - the bus begins a planned layover at a point
- departure - the bus leaves a point ending a planned layover
- virutal. - events that were not part of the transit agencies original data set

```
CREATE TABLE event_type_constants
( id INTEGER NOT NULL PRIMARY KEY,  name VARCHAR(100) NOT NULL
UNIQUE);
INSERT INTO event_type_constants (id, name) VALUES (3, 'Sign Change');
INSERT INTO event_type_constants (id, name) VALUES (8, 'Begin Layover');
INSERT INTO event_type_constants (id, name) VALUES (9, 'End Layover');
INSERT INTO event_type_constants (id, name) VALUES (10, 'Begin Trip');
INSERT INTO event_type_constants (id, name) VALUES (11, 'End Trip');
INSERT INTO event_type_constants (id, name) VALUES (12, 'Begin Deadhead');
INSERT INTO event_type_constants (id, name) VALUES (13, 'End Deadhead');
INSERT INTO event_type_constants (id, name) VALUES (150, 'Passing Timepoint');
INSERT INTO event_type_constants (id, name) VALUES (151, 'Passing Timepoint
(Interpolated)');
```

route_direction_constants
Referenced by: patterns
Identifies: the direction the bus travels.

```
CREATE TABLE route_direction_constants
(id INTEGER NOT NULL PRIMARY KEY,  name VARCHAR(100) NOT NULL
UNIQUE);
INSERT INTO route_direction_constants (id, name) VALUES (1, 'N');
INSERT INTO route_direction_constants (id, name) VALUES (2, 'S');
INSERT INTO route_direction_constants (id, name) VALUES (3, 'E');
INSERT INTO route_direction_constants (id, name) VALUES (4, 'W');
INSERT INTO route_direction_constants (id, name) VALUES (5, 'NE');
INSERT INTO route_direction_constants (id, name) VALUES (6, 'NW');
INSERT INTO route_direction_constants (id, name) VALUES (7, 'SE');
```

INSERT INTO route_direction_constants (id, name) VALUES (8, 'SW');
INSERT INTO route_direction_constants (id, name) VALUES (9, 'Inbound');
INSERT INTO route_direction_constants (id, name) VALUES (10, 'Outbound');
INSERT INTO route_direction_constants (id, name) VALUES (11, 'Circular');

service_type_constants
Referenced by: trips
Identifies: a trip as either regular, express, or non-revenue. Deadhead trips have a service_type of non-revenue.

CREATE TABLE service_type_constants
(id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL UNIQUE);
INSERT INTO service_type_constants (id, name) VALUES (1, 'regular');
INSERT INTO service_type_constants (id, name) VALUES (2, 'express');
INSERT INTO service_type_constants (id, name) VALUES (8, 'non-revenue');

trip_type_constants
Referenced by: trips
Identifies: revenue or non-revenue trips. Deadhead trips have a trip_type of non-revenue.

CREATE TABLE trip_type_constants
(id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL UNIQUE);
INSERT INTO trip_type_constants (id, name) VALUES (1, 'revenue');
INSERT INTO trip_type_constants (id, name) VALUES (2, 'deadhead');

day_type_constants
Referenced by: blocks
Identifies: type of day, which can be as specific as Sunday or as general as weekend

CREATE TABLE day_type_constants
(id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL UNIQUE);
INSERT INTO day_type_constants (id, name) VALUES (1, 'Sunday');
INSERT INTO day_type_constants (id, name) VALUES (2, 'Monday');
INSERT INTO day_type_constants (id, name) VALUES (3, 'Tuesday');
INSERT INTO day_type_constants (id, name) VALUES (4, 'Wednesday');
INSERT INTO day_type_constants (id, name) VALUES (5, 'Thursday');
INSERT INTO day_type_constants (id, name) VALUES (6, 'Friday');
INSERT INTO day_type_constants (id, name) VALUES (7, 'Saturday');
INSERT INTO day_type_constants (id, name) VALUES (8, 'Holiday');
INSERT INTO day_type_constants (id, name) VALUES (9, 'Weekday');
INSERT INTO day_type_constants (id, name) VALUES (10, 'Weekend');
INSERT INTO day_type_constants (id, name) VALUES (11, 'Weekday, school closed');
INSERT INTO day_type_constants (id, name) VALUES (255, 'Unspecified');

condition_type_constants

Referenced by: trips
Identifies: Condintions that affect a trip.

CREATE TABLE condition_type_constants
(id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL UNIQUE);
INSERT INTO condition_type_constants (id, name) VALUES (1, 'Canceled if Snow Event');

coord_type_constants
Referenced by: measures
Identifies: What type of coordinates are used within the schema.

CREATE TABLE coord_type_constants
(id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL UNIQUE);
INSERT INTO coord_type_constants (id, name) VALUES (1, 'Geodetic');
INSERT INTO coord_type_constants (id, name) VALUES (2, 'State Plane');

unit_type_constants
Referenced by: measures
Identifies: Meaurement units, such as feet or meters.

CREATE TABLE unit_type_constants
(id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL UNIQUE);
INSERT INTO unit_type_constants (id, name) VALUES (1, 'feet');
INSERT INTO unit_type_constants (id, name) VALUES (2, 'meters');

status_type_constants
Referenced by: avl_data
Identifies: status codes derived from AVL system

CREATE TABLE status_type_constants
(id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL UNIQUE);
INSERT INTO status_type_constants (id, name) VALUES (1, 'OK');
INSERT INTO status_type_constants (id, name) VALUES (2, 'Off Route');
INSERT INTO status_type_constants (id, name) VALUES (3, 'Bad Odometer');
INSERT INTO status_type_constants (id, name) VALUES (4, 'Bad Data');

Data Tables

time_points
Referenced by: time_point_intervals, time_point_events

CREATE TABLE time_points (
  time_point_id INTEGER NOT NULL PRIMARY KEY,

```
  time_point_designator VARCHAR(16) NOT NULL,
  time_point_name VARCHAR(80) NOT NULL,
  time_point_name_short VARCHAR(20),
  coord1 NUMERIC(15,6) not null,
  coord2 NUMERIC(15,6) not null
);
```

TCIP Description: A point at which time is measured to create trips.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
time_point_id	integer not null primary key	A number assigned by a transit agency to uniquely identify a location at which time is measured.	schddTimePointID	IDENS
time_point_designator	varchar(16) not null	An alpha-numeric identifier of a location at which time is measured.	schddTimePointDesignator	NAME8
time_point_name	varchar(80) not null	The name of a time point.	schddTimePointName	OCTET STRING
time_point_name_short	varchar(20)	A short name associated with a time point. This 4-character name supports existing legacy systems which rely on 4 characters to identify their time points.	schddTimePtNameShort	OCTET STRING
coord1, coord2	float	A location reference for a point feature.	spPointClass	N/A

stop_points
Referenced by: stop_point_events

```
CREATE TABLE stop_points (
  stop_point_id INTEGER NOT NULL PRIMARY KEY,
  stop_point_designator VARCHAR(16) NOT NULL UNIQUE,
  stop_point_name VARCHAR(80),
  stop_point_description VARCHAR(255),
  coord1 NUMERIC(15,6),
  coord2 NUMERIC(15,6)
);
```

TCIP Description: A point where public transportation customers board or alight from a transit vehicle in revenue service.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
stop_point_id	integer not null primary key	Identifies a point where public transportation customers board or alight from a transit vehicle in revenue service.	cptddStopPointID	IDENS
stop_point_designator	varchar(16) not null unique	ITS-UW Def: An alpha-numeric identifier for a stop point.		
stop_point_name	varchar(80)	A name of a point where public transportation customers board or alight from a PTV in revenue service.	cptddStopPointName	NAME
stop_point_description	varchar(256)	An expository description of a Stop Point.	cptddStopPointDescription	FOOTNOTE
coord1, coord2	float	A location reference for a point feature.	spPointClass	N/A

time_point_intervals
Referenced by: pattern_tpis

```
CREATE TABLE time_point_intervals (
  tpi_id INTEGER NOT NULL PRIMARY KEY,
  tpi_designator VARCHAR(16) NOT NULL UNIQUE,
  start_point_id INTEGER NOT NULL REFERENCES time_points,
  end_point_id INTEGER NOT NULL REFERENCES time_points,
  tpi_length NUMERIC(15,6) NOT NULL
);
```

TCIP Description: The segment between two timepoints.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
tpi_id	integer not null primary key	Identifies a unique path between two time points.	schddTimePointIntervalID	IDENS
tpi_designator	varchar(16) not null unique	An alpha-numeric identifier for a unique path between two time points.	schddTimePointIntervalDesginator	NAME8
start_point_id	integer not	See	schStartPointID	N/A

	null	schddTimePointID		
end_point_id	integer not null	See schddTimePointID	schEndPointID	N/A
tpi_length	float not null	ITS-UW Def: Length between start and end points.	N/A	N/A

tpi_shape_points
Referenced by: N/A

```
CREATE TABLE tpi_shape_points (
  tpi_id INTEGER NOT NULL REFERENCES time_point_intervals,
  sequence_in_tpi INTEGER NOT NULL,
  UNIQUE (tpi_id, sequence_in_tpi),
  distance_into_tpi NUMERIC(15,6) NOT NULL,
  coord1 NUMERIC(15,6),
  coord2 NUMERIC(15,6)
);
```

Description: Defines the spatial path of time point intervals.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
tpi_id	integer not null	Identifies a unique path between two time points.	schddTimePointIntervalID	IDENS
sequence_in_tpi	integer not null	ITS-UW Def: Enumerates shape points from 0 to n for each time point interval.	N/A	N/A
distance_into_tpi	float not null	ITS-UW Def: The distance from the beginning of the tpi.	N/A	N/A
coord1, coord2	float	A location reference for a point feature.	spPointClass	N/A

routes
Referenced by: patterns

```
CREATE TABLE routes (
  route_id INTEGER NOT NULL PRIMARY KEY,
  route_designator VARCHAR(16) NOT NULL UNIQUE,
  route_name VARCHAR(60)
);
```

TCIP Description: A collection of patterns in revenue service.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
route_id	integer not null primary key	Identifies a collection of patterns in a revenue service.	schddRouteID	IDENS
route_designator	varchar(16) not null unique	An alpha-numeric identifier of a collection of patterns in a revenue service.	schddRouteDesignator	NAME8
Route_name	varchar(60)	Names a collection of patterns in a revenue service.	schddRouteName	NAME

patterns
Referenced by: trips, pattern_tpis

```
CREATE TABLE patterns (
  pattern_id INTEGER NOT NULL PRIMARY KEY,
  pattern_designator VARCHAR(16) NOT NULL UNIQUE,
  pattern_name VARCHAR(60),
  route_direction INTEGER REFERENCES route_direction_constants,
  route_id INTEGER NOT NULL REFERENCES routes,
  pattern_length NUMERIC(15,6) NOT NULL
);
```

TCIP Description: One of multiple outer route segments served by a single transit route.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
pattern_id	integer not null primary key	A unique identifier assigned by a transit agency for a defined sequence of points, events and activation events along a variation of a route.	SCH_PatternID_id	IDENS
pattern_designator	varchar(16) not null unique	An alpha-numeric identifier for a defined sequence of points, events and activation events along a route.	schddPatternDesignator	NAME8
pattern_name	varchar(60)	The name for a defined sequence of points and events along a variation of a route that represents a physical path traversed by a transit vehicle in a	schddPatternName	

		network.		
route_direction	integer	A name which describes the direction of a route.	schddRouteDirectionName	UBYTE
route_id	integer not null	Identifies a collection of patterns in a revenue service.	schddRouteID	IDENS
pattern_length	float not null	ITS-UW Def: Length of the pattern.	N/A	N/A

blocks
Referenced by: trips

```
CREATE TABLE blocks (
  block_id INTEGER NOT NULL PRIMARY KEY,
  day_type INTEGER NOT NULL REFERENCES day_type_constants,
  block_length NUMERIC(15,6)
);
```

pattern_tpis
Referenced by: N/A

```
CREATE TABLE pattern_tpis (
  pattern_id INTEGER NOT NULL REFERENCES patterns,
  sequence_in_pattern INTEGER NOT NULL,
  UNIQUE (pattern_id,sequence_in_pattern),
  tpi_id INTEGER NOT NULL REFERENCES time_point_intervals
);
```

Description: Enumerates a sequence of time point intervals for each pattern.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
pattern_id	integer not null	A unique identifier assigned by a transit agency for a defined sequence of points, events and activation events along a variation of a route.	SCH_PatternID_id	IDENS
sequence_in_pattern	integer not null	ITS-UW Def: Enumerates time point intervals for each pattern from 0 to n.	N/A	N/A
tpi_id	integer not null	Identifies a unique path between two time points.	schddTimePointIntervalID	IDENS

trips
Referenced by: time_point_events, stop_point_events, avl_data

```
CREATE TABLE trips (
  trip_id INTEGER NOT NULL PRIMARY KEY,
  trip_designator VARCHAR(16) NOT NULL,
-- day_type now added to blocks table. this field now deprecated...
  day_type INTEGER NOT NULL REFERENCES day_type_constants,
  block_id INTEGER NOT NULL REFERENCES blocks,
  service_type INTEGER NOT NULL REFERENCES service_type_constants,
  trip_type INTEGER NOT NULL REFERENCES trip_type_constants,
  sequence_in_block INTEGER NOT NULL,
  UNIQUE( block_id, sequence_in_block ),
  distance_into_block NUMERIC(15,6) NOT NULL,
  pattern_id INTEGER REFERENCES patterns,
  destination VARCHAR(64),
  condition_type INTEGER REFERENCES condition_type_constants
);
```

Description: Describes all scheduled bus movements.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
Trip_id	ingeter not null primary key	A number assigned by a transit agency that uniquely identifies a one way operation of a transit vehicle between two terminus points on a route.	SCH_TripID_id	IDENS
Trip_designator	varchar(16) not null	ITS-UW Def: An alpha-numeric assigned by a transit agency that uniquely identifies a one way operation of a transit vehicle between two terminus points on a pattern.		
Day_type	integer not null	A type of day characterized by one or more properties which affect public transport operation.	schddDayType	UBYTE
Block_id	integer not null	A unique identifier within a day type which is used to associate a sequence of trips to a transit vehicle.	SCH_BlockID_id	IDENS
Service_type	integer not	Type of transit service	schddServiceTyp	UBYTE

	null	provided (either ‘regular’, ‘express’, or ‘non revenue’).	e	
Trip_type	integer not null	ITS-UW Def: Identifies a trip as ‘revenue’ or ‘deadhead’.	schddTripType	
Sequence_in_block	integer not null	ITS-UW Def: Enumerates trips from 1 to n for a each block.	N/A	N/A
Distance_into_block	float not null	ITS-UW Def: The distance from the start of the block of the beginning of the trip.		
Pattern_id	integer not null	A unique identifier assigned by a transit agency for a defined sequence of points, events and activation events along a variation of a route.	SCH_PatternID_id	IDENS
Destination	varchar(64) not null	ITS-UW Def: The destination text displayed on the front of the bus.	N/A	N/A
Condition_type	integer	ITS-UW Def: Identifies special conditions that affect a trip.		

time_point_events
Referenced by: N/A

```
CREATE TABLE time_point_events (
  time_point_id INTEGER NOT NULL REFERENCES time_points,
  event_time INTEGER NOT NULL,
  trip_id INTEGER NOT NULL REFERENCES trips,
  event_type INTEGER NOT NULL REFERENCES event_type_constants,
  distance_into_trip NUMERIC(15,6) NOT NULL,
  sequence_in_trip INTEGER NOT NULL,
  UNIQUE(trip_id, sequence_in_trip),
  UNIQUE(trip_id,time_point_id,event_time,event_type)
);
```

Description: Describes scheduled events for trips at time points.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
time_point_id	integer not null	A number assigned by a transit agency to	schddTimePointID	IDENS

		uniquely identify a location at which time is measured.		
event_time	integer not null	ITS-UW Def: Minutes after midnight time value for this event.	N/A	N/A
trip_id	integer not null	A number assigned by a transit agency that uniquely identifies a one way operation of a transit vehicle between two terminus points on a route.	SCH_TripID_id	IDENS
event_type	integer not null	ITS-UW Def: Identifies an event as ‘passing’, ‘arrival’, ‘departure’, or ‘virtual’.	N/A	N/A
distance_into_trip	float not null	ITS-UW Def: Distance value of where this event occurs from the beginning of the trip’s pattern. See the measures table for definition of units.	N/A	N/A

stop_point_events
Referenced by: N/A

```
CREATE TABLE stop_point_events (
  stop_point_id INTEGER NOT NULL REFERENCES stop_points,
  event_time INTEGER NOT NULL,
  trip_id INTEGER NOT NULL REFERENCES trips,
  event_type INTEGER NOT NULL REFERENCES event_type_constants,
  distance_into_trip NUMERIC(15,6) NOT NULL,
  sequence_in_trip INTEGER NOT NULL
);
```

Description: Describes schedlued events for trips at stop points.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
Stop_point_id	integer not null	Identifies a point where public transportation customers board or alight from a transit vehicle in revenue service.	cptddStopPointID	IDENS
Event_time	integer not null	ITS-UW Def: Minutes after midnight time value	N/A	N/A

		for this event.		
Trip_id	integer not null	A number assigned by a transit agency that uniquely identifies a one way operation of a transit vehicle between two terminus points on a route.	SCH_TripID_id	IDENS
Event_type	integer not null	ITS-UW Def: Identifies an event as ‘passing’, ‘arrival’, ‘departure’, or ‘virtual’.	N/A	N/A
Distance_into_trip	float not null	ITS-UW Def: Distance value of where this event occurs from the beginning of the trip’s pattern. See the measures table for definition of units.	N/A	N/A

time_table_version
Referenced by: N/A

```
CREATE TABLE time_table_version (
  version_id INTEGER NOT NULL PRIMARY KEY,
  version_name VARCHAR(64) NOT NULL,
  activation_date CHAR(18) NOT NULL,
  deactivation_date CHAR(18),
  agency_name VARCHAR(64) NOT NULL
);
```

Description: All the tables in this schema are for a paticular time table version. Each time table version will need its own instantiation of the tables in this schema. This table idenifies an instantiation of this schema by providing an activation date, deactivation dates, version id, version name, and the name of the transit agency.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
version_id	integer not null primary key	A unique number associated with the concepts, messages and parts of a schedule including all the messages associated with the scheduling business area.	schddTimeTable VersionID	IDENS
version_name	varchar(64) not null	Name of a time table, e.g., “summer.”	schddTimeTable VersionName	NAME8

activation_date	char(18) not null	The date a public transportation staff, service, facility or asset is placed in service, becomes operational or is registered. This may refer to a Stop Point, timetable, roster, operator logon, equipment installation, etc.	cptddActivationDate	DATE
deactivation_date	char(18)	The date a public transportation person, place or thing is taken out of service, decommissioned or deregistered. This may refer to an Stop Point, timetable, roster or transit employee logoff, etc.	cptddDeactivation Date	DATE
agency_name	varchar(64) not null	ITS-UW Def: The name of the transit agency.	N/A	N/A

measures
Referenced by: N/A

```
CREATE TABLE measures (
  coord_type INTEGER NOT NULL PRIMARY KEY REFERENCES
coord_type_constants,
  ref_system VARCHAR(64) NOT NULL,
  coord_units INTEGER NOT NULL REFERENCES unit_type_constants,
  length_units INTEGER NOT NULL REFERENCES unit_type_constants
);
```

Description: This table identifies the type of corrdinates and units used in other tables in this schema. Other tables in this schema, such as tpi_shape_points, have coord1 and coord2 columns. The coord_type column of this table explains what is meant by coord1 and coord2. There must be one and only row in this table for each instantiation of the schema. The length_units column of this table identifies what units are used in othere tables, for example the distance_into_trip column in the time_point_events table.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
coord_type	integer not null primary key	ITS-UW Def: What type of coordinates are used within the schema. If 'Geodetic' then coord1 is	N/A	N/A

		latitude and coord2 is longitude. If 'State Plane' then coord1 is x and coord2 is y.		
ref_system	varchar(64) not null	ITS-UW Def: Identifies the reference system that coordinates in this schema come from, for example "Washington North Zone".	N/A	N/A
coord_units	integer not null	ITS-UW Def: Identifies the units of coord1 and coord2 columns in other tables in this schema as either feet or meters.	N/A	N/A
lengh_units	integer not null	ITS-UW Def: Identifies the units used for measurements by other tables in this schema such as feet or meters.	N/A	N/A

avl_data
Referenced by: N/A

```
CREATE TABLE avl_data (
  vehicle_id INTEGER NOT NULL,
  data_time CHAR(18) NOT NULL,
  PRIMARY KEY (vehicle_id, data_time),
  trip_id INTEGER REFERENCES trips,
  distance_into_trip INTEGER NOT NULL,
  status_type INTEGER NOT NULL REFERENCES status_type_constants,
  coord1 NUMERIC(15,6),
  coord2 NUMERIC(15,6),
  sched_dev INTEGER,
  avl_source VARCHAR(255)
);
```

Description: AVL data archive table.

SQL Colum Name	SQL Type	TCIP Definition	TCIP Name	TCIP Type
vehicle_id	integer not null	A unique number assigned by the transit agency to each of their vehicles.	OB_J1587_VehicleIdentificationNumber	IDENS
data_time	char(18) not null	ITS-UW Def: Time value of AVL update.	N/A	N/A
trip_id	integer	A number assigned by a transit agency that	SCH_TripID_id	IDENS

		uniquely identifies a one way operation of a transit vehicle between two terminus points on a route.		
distance_into_trip	integer not null	ITS-UW Def: Distance value from the beginning of the trip's pattern. See the measures table for definition of units.	N/A	N/A
status_type	integer not null	ITS-UW Def: Identifies AVL status values.	N/A	N/A
coord1, coord2	float	A location reference for a point feature.	spPointClass	N/A
sched_dev	integer	ITS-UW Def: The number of positive or negative minutes ahead or behind schedule determined by the AVL system.	N/A	N/A
avl_source	varchar(256)	ITS-UW Def: Allows for storage of AVL data exactly as it was received. When status_type is 'Bad Data' this column could contain the AVL data that was not understood.		

day_types
Referenced by: N/A

```
CREATE TABLE day_types (
  day_id DATE NOT NULL,
  day_type INTEGER NOT NULL REFERENCES day_type_constants
);
```

Filling the Transit Schema

Tables that are referenced by other tables must be filled before they can be referenced. Below is the order for filling the tables in this schema.

DRAFT

Constant Tables

These tables must be filled before the Data Tables. The rows of these tables have been defined in this document. These tables do not reference each other and can be filled in any order.

Data Tables

Fill these tables in this order:

Core Tables

time_points
time_point_intervals
tpi_shape_points
routes
patterns
pattern_tpis
trips
time_point_events

Stop Point Tables

stop_points
stop_point_events

Schema Referenence Tables

time_table_version
measures