STATE OF CALIFORNIA DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR0003 (REV. 10/98)

1. REPORT NUMBER CA11-1223	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER

4. TITLE AND SUBTITLE

Development of an Integrated Adaptive Transit Signal Priority (ATSP) and Dynamic Passenger Information (DPI) System

5. REPORT DATE

August 2011

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)

Liping Zhang, Kun Zhou, Jingquan Li, Wei-Bin Zhang, Meng Li, Guoyuan Wu and Sonja Sun

8. PERFORMING ORGANIZATION REPORT NO.

UCB-ITS-PRR-2011-

9. PERFORMING ORGANIZATION NAME AND ADDRESS

California PATH Program
University of California, Berkeley
1357 .46th St.
Richmond, CA 94804

10. WORK UNIT NUMBER

11. CONTRACT OR GRANT NUMBER

65A0277

12. SPONSORING AGENCY AND ADDRESS

California Department of Transportation
Division of Research and Innovation
P.O. Box 942873, MS 83
Sacramento, CA 94273-0001

13. TYPE OF REPORT AND PERIOD COVERED

June 30, 2008 – March 31, 2011

14. SPONSORING AGENCY CODE

15. SUPPLEMENTAL NOTES

16. ABSTRACT

This document reports the development and implementation of an integrated Adaptive Transit Signal Priority (ATSP) and Dynamic Passenger Information (DPI) system, and extensive testing with a real-world operation setting on a 10-mile-long ATSP-enabled road segment along El Camino Real, between 42nd Avenue in San Mateo and Bayhill Road in San Bruno, and on two SamTrans (San Mateo County Transit District) revenue bus routes 390 and 391. The integrated ATSP/DPI system is built upon the existing Transit Automated Vehicle Location/Advanced Communications System (AVL/ACS) to provide transit buses with needed priority at prioritized intersections and to provide transit customers with real-time transit information. The extensive testing in the real-world setting provided valuable opportunities to discover and thus resolved a number of issues that might have prevented the system from achieving high integrate-ability and reliability in the future deployment on a large public scale.

17. KEY WORDS

Dynamic Passenger Information, Transit Signal Priority, Transit Advanced Communications System

18. DISTRIBUTION STATEMENT

No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161

19. SECURITY CLASSIFICATION (of this report) Unclassified	20. NUMBER OF PAGES	21. PRICE

Reproduction of completed page authorized

Development of an Integrated Adaptive Transit Signal Priority (ATSP) and Dynamic Passenger Information (DPI) System

Final Report for Contract TA65A0277

Prepared by:

California PATH
University of California, Berkeley
and
California Department of Transportation
in Collaboration with
San Mateo County Transit District

August 2011

DISCLAIMER STATEMENT

This document is disseminated in the interest of information exchange. The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California or the Federal Highway Administration. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement by the Department of any product described herein.

For individuals with sensory disabilities, this document is available in Braille, large print, audiocassette, or compact disk. To obtain a copy of this document in one of these alternate formats, please contact: the Division of Research and Innovation, MS-83, California Department of Transportation, P.O. Box 942873, Sacramento, CA 94273-0001.

Acknowledgements

This work was performed by the California PATH Program at the University of California at Berkeley in cooperation with the State of California Business, Transportation and Housing Agency, Department of Transportation (Caltrans). The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California.

The authors thank Larry Orcutt, Nancy Chinlund and Greg Larson of Caltrans' Division of Research and Innovation, Scott Sauer of Caltrans' Division of Mass Transportation, James Lau and Kai Leung of Caltrans's Division of Traffic Operations, Min Yee Lee, Harris Zaw and David Man of Caltrans District 4, and Roi Kingon of San Mateo County Transit District for their support and advice during the project, and Scott Johnston and Mark Miller of PATH for their technical contributions.

Author List

University of California, Berkeley:
Liping Zhang
Kun Zhou
Jingquan Li
Wei-Bin Zhang
Meng Li[1]
Guoyuan Wu[2]

Division of Research and Innovation, Caltrans:
Sonja Sun

[1] Now at Tsinghua University

[2] Now at University of California at Riverside

Executive Summary

The transit industry has applied various advanced Intelligent Transportation System (ITS) technologies to improve the efficiency of transit operations and to provide passengers more reliable transit service in order to better serve their riders. Among these ITS technologies, the Automatic Vehicle Location and Advanced Communication System (AVL/ACS), Transit Signal Priority (TSP) and Dynamic Passenger Information (DPI) are widely deployed technologies.

TSP and DPI share common critical elements. Both require an onboard processor, vehicle location information, communication links between vehicle and infrastructure and capability of predicting bus arrival time at fixed geographic locations (signalized intersection and bus-stops). AVL/ACS systems, emerged since the late 1980s and 1990s primarily for fleet operation management, provide the technical capabilities required by TSP and DPI. Today's bus AVL/ACS systems include both the core location tracking capabilities using Global Positioning System (GPS) and information exchange capabilities between fleet vehicles and dispatch. Using AVL/ACS onboard GPS receiver and processor, the functionality of prediction of bus arrival time can be added to the AVL/ACS systems. Despite the fact that these three systems share both hardware and software and that AVL/ACS already has the critical elements in place, they are separately implemented. As a result, multiple GPS units, computer hardware and communication links need to be installed on a single bus. Each of these systems costs the transit agencies millions of dollars. The disintegration not only causes redundant capital investments and costs for operation and maintenance, but also prevents transit from getting full benefits enabled by the ITS technologies.

One critical technical challenge of integrating TSP/DPI with AVL/ACS is how to interface with the existing ACS as a communication backbone for facilitating TSP requests. The current transit ACS technologies use a center-polling communication protocol. The ACS server initiates and polls one bus at a time for every bus in the fleet by turn and the polled bus responds with a message containing its location, schedule and route adherence status data. With this center-polling protocol, the polling and reporting rate from individual bus is usually between 1 and 2 minutes, which is considerably low to serve TSP purposes. In addition, with the current AVL/ACS, a bus is communicating with dispatch not directly via intersection infrastructure.

More detailed investigation on of the ACS communication protocol revealed that the regular periodic bus location polling only utilized 60 to 70 percent of ACS channel capacity. The remaining 30 to 40% capacity, although can also be used for bus polling, is reserved to meet other operation and management needs and is usually underutilized. Need-based dynamic polling that utilizes the available channel capacity for additional bus location updates could accommodate the communication needs for TSP.

The research is a continuation of two previous PATH research projects on the development and implementation of an adaptive transit signal priority (ATSP) system. California PATH, in collaboration with the California Department of Transportation (Caltrans) and San Mateo County Transit District (SamTrans) developed and field tested a centralized adaptive transit signal priority (ATSP) system with the architecture following National Transportation Communications for ITS Protocol (NTCIP) 1211 scenario 3. Although the field operational tests demonstrated the positive impacts of ATSP on transit operation, the tests were conducted without using SamTrans' existing AVL/ACS system but with second-by-second bus GPS location data for the generation of TSP requests.

This research project was conducted to develop an integrated ATSP/DPI system that utilizes the existing hardware and communication link offered by AVL/ACS. The eventual goal is to have one integrated system to support the purposes of fleet management, TSP and DPI. The hypothesis was that the integration approach is not only cost effective but can also improve the performance of dynamic passenger information. When integrated with ATSP, DPI processes share the information regarding ATSP operation status leading to better predicted bus arrival times at bus-stops.

Under the this project, an open system architecture has been developed to build integrated ATSP/DPI upon existing AVL/ACS, with layered database structures and processing modules, and various interfaces for data collection and information dissemination. An integrated, flexible and scalable database has also been developed to support the open architecture ITS applications. A bus arrival time prediction model has added transit operation factors in the prediction algorithm such that it can serve both short-term prediction for ATSP and long-term prediction for DPI. Web APIs have been developed to deliver passenger dynamic information in various ways, including pre-trip planning, information querying by smart phones and information display at bus-stops.

The critical link for integrating these technologies is to enable the ACS communication to carry out inserted polls for the creation of priority requests. Emulated dynamic polling strategies were designed, implemented and tested in the laboratory environment and in the field. Due to the lack of access to ACS processing software, an ACS emulation approach was adopted for testing the capability of existing ACS in supporting the integrated ATSP/DPI and the resulting performance of ATSP and DPI. Fifteen SamTrans buses were instrumented with cell-phone-based AVL systems to record and send second-by-second bus location data to the ATSP/DPI central processor, as input to the ACS emulation platform. It is worth pointing out that the field testing system is not an actual integrated system in the sense that dynamic polling requests were handled by "emulated" ACS rather than the existing ACS. However, findings from the field testing provide a fair assessment on the capability of utilizing the existing AVL/ACS for ATSP and DPI, and of the benefits due to the integration approach.

The ATSP system has been implemented on a 50-intersection road segment along El Camino Real, from the intersection with 41st Avenue in the City of San Mateo and the intersection with Bayhill Road in the City of San Bruno. Field testing was conducted for SamTrans bus routes 390 and 391. The ATSP corridor is the central part of routes 390 and 391. Outcomes from the evaluation study using eight-week "after" and four-week "before" filed operational data showed very positive results.

Field testing shows the emulated dynamic polling under ACS emulation provided necessary additional bus location updates for the needs of ATSP without overwhelming the channel capacity of the existing ACS. On average about 120 polls were dynamically inserted along the 10-mile-long ATSP corridor for the average travel period of 55 minutes. The average time gap between inserted polls was 27 seconds. Ninety-seven percent (97%) of inserted polls were processed without the collision of multiple competing polling requests. Even when a request collision did happen, the delay in responding to inserted polling request was within 4 seconds.

ATSP under emulated dynamic polling worked as expected. The positive impacts of ATSP on transit operation are shown in several aspects. In comparison with the scenario of ATSP off, when ATSP was on,

- Bus trip travel time was reduced by 4.4% for southbound trips and by 1.4% for northbound trips;
- Bus total intersection delay was reduced by 19.4% for southbound trips and by 9.2% for northbound trips;
- Number of stops at prioritized intersections was reduced by 5.9% southbound and by 4.1% northbound;
- Bus running speed was increased by 4.3% southbound and by 1.5% northbound.

All these changes are statistically significant at the 5% level. Moreover, the percentage of buses running over 5 minutes behind the schedule was reduced by 27%. The evaluation results also confirmed that the time-point holding phenomena due to TSP. With TSP savings at intersections, the bus tended to wait longer at the time-points. The average dwelling time per time-point per trip was increased by 22.5% for southbound and by 11% for northbound. The additional waiting time at the time points cancelled the time saved at the intersection by TSP. Excluding the dwelling time at time-points for both the "before" and "after" scenarios, bus trip travel time could be reduced by 5.6% for southbound and by 2.4% for northbound.

Dynamic passenger information based on the integrated ATSP/DPI has been displayed to the public at Millbrae Station kiosks. Prediction error of bus arrive time was within +/- 2 minutes when the bus was 40 minutes away. SamTrans personnel has been site observing multiple times and found the information "is remarkably accurate". Integrating DPI with ATSP improved the performance of DPI, as the inserted polls for ATSP purpose contributed for better prediction results. The absolute error mean was reduced by 16% in the section covering four time-points within the ATSP corridor.

The results from the field operational tests demonstrate that ACS can potentially accommodate the ATSP communication needs with smart channel management such as dynamic polling, and that integrated ATSP/DPI benefits DPI with more bus location updates becoming available. It is recommended that transit agencies adopt the integration approach when AVL/ACS systems are deployed or updated by implementing dynamic polling strategies in order to facilitate cost effective integration of ATSP and DPI functions with ACS.

Table of Contents

Table of Figures

List of Tables

1. Introduction

The 2009 Urban Mobility Report (UMR) (Schrank & Lomax, 2009) shows that the congestion problem has been getting worse over the last two decades in America's 439 urban areas, as the increase in traffic volume has exceeded the increase in available roadway capacity. The UMR report highlights the role of public transportation in congestion relief and estimates the reduction in congestion cost provided by public transportation:

> *"If public transportation service had been discontinued and the riders traveled in private vehicles in 2007, the 439 urban areas would have suffered an additional 646 million hours of delay and consumed 398 million more gallons of fuel, 40% more than a decade ago. The value of the additional travel delay and fuel that would have been consumed if there were no public transportation service would be an additional $13.7 billion, a 16% increase over current levels in the 439 urban areas."*

Using information from the National Transit Database on America's 81 urban areas in year 2002, Harford (Harford, 2006) estimated an average of $0.47 congestion savings per passenger mile traveled by public transit. The congestion saving estimates approximately match with the finding concluded from an evaluation study covering Australasian, European and North American (Aftabuzzaman, Currie, & Sarvi, 2010).

Growing congestion in U.S. transportation systems is one of the largest threats to the U.S. economy and to the quality of life of Americans. In 2006, the U.S. Department of Transportation (DOT) identifies urban congestion reduction as one of the federal government's top priorities.

A crucial element in achieving the goal of congestion relief is transit improvement through the application of Intelligent Transportation Systems (ITS) to make transit a more attractive choice for travelers, encouraging mode shift and ultimately helping to reduce congestion ((US DOT, 2006)). Among the transit ITS technologies deployed in the United States, Transit Signal Priority (TSP) and Dynamic Passenger Information (DPI) have been wanted by a majority of transit agencies.

When a highway is congested, even small reductions in traffic volume can significantly increase travel speed. While in general both traffic congestion intensity and transit service increase with city size, comparisons between cities indicate that total congestion delay tends to decrease with the provision of good transit service (Litman, 2011). In addition, the total congestion delay declines in cities with an increase in quality transit mileage (Winston & Langer, 2006).

Despite the benefits to use transit, car drivers are reluctant to make the switch. Transit travelers' perceptions and satisfaction of waiting, transfer and transit travel times contribute greatly to their decisions whether or not to take transit. One of the barriers to use transit as an alternative mode is car drivers' distorted perceptions of transit service quality, which have considerable influence on their choice-sets. Van Exel and Rietveld (van Exel & Rietveld, 2010) found that car drivers' perceptions of transit travel time many times deviate substantially from real travel times. The ratio of perceived transit travel time to the car travel time was inversely related with both the car travel time and respondents' experience with transit on the same trip – the smaller the car travel time, the greater the distorted perception of transit travel time; and drivers with transit experience had more accurate perceptions of transit travel time. They also found that providing better information to car drivers about objective transit travel times will lead to a large proportion of car drivers including transit in their travel choice sets. Through a series of focus group studies, Kenyon and Lyons (Kenyon & Lyons, 2003) also reported that regular users of transit held more positive opinions of transit services and infrequent users had more negative perceptions. They concluded that presenting information to habitual travelers about the cost, travel time, comfort and convenience of travel alternatives could challenge previous perceptions of the utility of non-car modes and lead to a mode change. Another project conducted by California PATH, the Networked Traveler transit and smart parking as part of the Safetrip-21 California field testing, has also demonstrated an integrated real-time traveler information tool that helped to encourage mode shift by providing better real-time information (Zhang, et al., 2011).

It is commonly understood that there is great potential for transit to attract more riders (and therefore reduce congestions) by improving the quality and usability of transit service for more competitive travel time and better user perceptions. Time-competitive transit also reduced commuters' stress and created a less negative mood when compared with commuting by car (Wener & Evans, 2003), and allowed passengers to productively use their time (Lyons, Jain, & Holley, 2007).

To date, various deployment cases of Transit Signal Priority (TSP) systems have demonstrated that TSP is effective in improving transit service quality. The benefits of TSP include reduced intersection delay and travel time, improved schedule adherence and travel time reliability, which lead to increased transit quality of service and improved customers' satisfaction. More detailed case studies and TSP benefits assessments can be found in (Smith & Hemily, 2005).

Providing dynamic passenger information (DPI) removes one of the largest barriers to use transit. The timely and accurate information keeps travelers informed about the next bus arrival time, connection schedule, and the total trip time, enabling them to feel certainty about the transit travel, thereby attracting choice riders.

In a survey study of Northern California commuters, Abdel-Aty et al. (ABDEL-ATY, KITAMURA, & JOVANIS, 1996) found that about 38 percent of the non-transit users might consider transit if appropriate transit information were easily available. Since more than 80 percent of survey respondents did not currently use transit, this indicates a promising effect of DPI systems in encouraging transit ridership if the desired information is provided.

Many transit systems now offer dynamic passenger information. This information provides many benefits. It makes waits more tolerable and in situations with multi-route options, passengers use the information for en-route travel decisions (Turnbull & Pratt, 2003). Evidence showed that this information increased passengers' satisfaction and attracted riders (TRB, 2003b):

- Ridership increased 6% in Brussels, Belgium, 5% in Liverpool, the United Kingdom, and 3% in Turin, Italy to the implementation of DPI at bus stops.
- User surveys indicate high satisfaction rates to the DPI information, 90% in Belgium, 98% in Glasgow, UK, 87% in Liverpool, UK, and 90% in Turin, Italy. As a result, transit service is perceived as more reliable and a majority of users say that they would be

encouraged to use transit more often because of the DPI systems.

- The London Bus Countdown system that displays bus arrival time information on electronic signs at bus stops reduced the perceived waiting time from 11.9 minutes to 8.6 minutes. Sixty-five percent of passengers felt that they waited a shorter time even though the actual waiting time did not change significantly. Eighty-nine percent of users state that waiting itself is more acceptable at stops equipped with the Countdown system. Sixty percent of respondents claimed their attitude toward bus travel had improved as a result of the system.

In a survey study of Seattle-area bus riders who use OneBusAway to access real-time bus arrival information, Ferris et al. (Ferris, Watkins, & Borning, 2011) found that 92 percent of respondents rated that they are either somewhat more satisfied or much more satisfied with transit, ninety-one percent reported spending less time waiting, and users intend to use transit more often on a weekly basis. In addition, 78% of respondents claimed they were more likely to walk to a stop on a different (fast) route as a result of using OneBusAway.

The PATH Networked Traveler study exhibited similar promising results. More than 75% of the users felt that the Path2go arrival information helped them reduce their waiting time at transit stations. On the other hand, less than 10% of the users held opposite viewpoints. About 54% of the users felt that using the applications would consider transit as a more viable commuting choice and about half of the users still felt unsure or decided not to switch (Zhang, et al., 2011). An independent survey held by Science Applications International Corporation also showed the effectiveness of the integrated real-time information. Thirty-two percent of the respondents indicated that Path2go made them more likely to choose an alternative mode (Jasper, Armstrong, Golembiewski, & Miller, 2011).

These studies provided evidences that once the barrier of uncertainty about transit travel (waiting time, trip time, etc.) is removed, customers will be encouraged to use transit more often and choice riders would be more likely to consider transit as an alternative mode of travel.

TSP and DPI are the two promising ITS technologies that can effectively address the major issues with the current transit service. When TSP and DPI are further integrated, transit agencies and travelers would benefit more from this integration, on which this report is exactly focused.

1.1. Overview of ATSP System

PATH, in collaboration with the California Department of Transportation (Caltrans) and San Mateo County Transit District (SamTrans), has developed and field tested a centralized Adaptive TSP (ATSP) system. The term “adaptive” refers to the priority request generation and execution processes are dynamically responding to both bus movement and current traffic conditions. This subsection provides a brief overview of the ATSP system. Interested readers are referred to the full report entitled “Field Operation Tests of Adaptive Transit Signal Priority Systems” (PATH Report, 2010).

The primary objective of developing ATSP system is to overcome the following limitations of the conventional TSP systems thereby providing greater benefits to transit and fewer impacts on non-transit traffic. In conventional TSP systems:

- A priority request is usually made upon the detection of bus presence within the detection zone, without predicted information about when the bus will arrive at the intersection, and the signal controller's response depends on the signal phase at the time the request is made. This leads to either inefficient TSP as not-necessary TSP will be granted if the bus were far from the intersection (can pass the intersection in normal green without TSP) or not-enough TSP if the bus were close to the intersection (waiting in the queue).
- Priority treatments do not effectively consider the impacts to auto traffic in the decision making. For early green treatment, a uniform predetermined cut-off ratio is applied to all conflict phases prior to the bus phase, resulting unbalanced traffic impacts.
- Priority decision making only considers single intersection not bus arrival times at adjacent downstream intersections. As a result, priority granted at the upstream intersection often caused additional bus delay and stops at downstream intersections thereby achieving no net benefit for transit.

The ATSP system addressed these limitations by

- adding a software module to provide predicted bus arrival time at the intersection for more informed priority decision making; and
- making priority decision with the consideration of multi-intersection effect and impacts on different traffic movements at the intersection.

The ATSP system makes priority decisions at two levels. At the multi-intersection level, the objective is to minimize the weighted number of stops and delays at successive intersections for transit buses, using predicted bus arrival time, traffic progression speed, and constraints on the level of priority that can be granted at intersections as input. At this level, the decision making is to ensure every priority granted is effective. The output of the multi-intersection level optimization is the green start time (for early green treatment) or green end time (for green extension treatment) on bus approach at each intersection. Balanced traffic impacts on general traffic are considered at the single intersection level. High quality signal status data and traffic count data available from the centralized traffic signal control systems are used to assess the delays would have on different traffic movements. The objective at single intersection level is to minimize the person delay and the output is the set of force-off points to be granted by the local controller.

To illustrate the effectiveness of the ATSP system, its performance was compared with that of the two distributed TSP systems deployed in the San Francisco Bay Area, namely the 3M Opticom TSP system by Alameda-Contra Costa Transit District (AC Transit) (PATH Report, 2008) and the loop-transponder-based TSP system by Santa Clara Valley Transportation Authority (VTA) (PATH Report, 2006). The comparison was based on field operational test data.

AC Transit's Opticom TSP system uses infrared-based communications between a bus and an individual traffic controller to detect the approach of a transit bus. The primary components of this system are an emitter mounted on the front of a bus, an optical detector mounted on the signal head, and a phase selectors installed in the roadside controller cabinet. When activated, the bus emitter sends a frequency coded optical message that identifies the bus to the detector. The detector then sends a signal to the signal controller via the phase selector to request TSP operations. The signal is dropped when the bus has cleared the intersection, and a check-out call is placed to terminate an on-going TSP execution.

VTA's loop-transponder-based TSP system utilizes the existing inductive loops for transit vehicle detection and priority request generation. The primary components of this system are a coded transponder attached to the underside of a bus, an antenna-based vehicle detection system integrated into a loop detector, and a receiver located inside the signal controller cabinet. Buses equipped with transponders are detected when traveling over the loop detectors and the sensor unit in the controller cabinet transmits the bus identification number and travel direction to the traffic controller to request signal priority. The existing advance loops are used for the initialization of TSP request and the stopline loops are used for checking-out an on-going TSP execution.

Both the Opticom TSP system and the loop-transponder-based TSP system use the same SCP logic developed by Caltrans, i.e., when a bus is detected with a red light all conflicted green phases will be uniformly truncated by 20% until the signal returns green on bus approach; and when a bus is detected with a green light the green phase will be held until received a "check-out" request or reached the maximum allowable green extension interval.

SamTrans' ATSP system adopts NTCIP-1211/TCIP Scenario 3 as the system architecture, where the primary components (i.e., the priority request generator and priority request server) are physically located at the Traffic Management Center (TMC). Figure 1-1 illustrates the ATSP system architecture design. The Orbital System is SamTrans' existing Automatic Vehicle Location/Advanced Communication Systems (AVL/ACS) and the TMC is for Caltrans District 4.

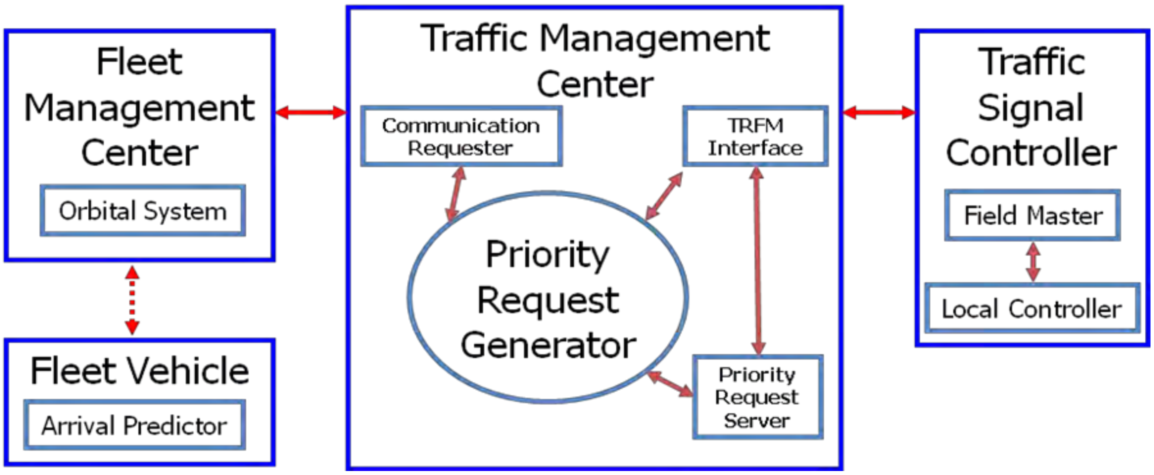


Figure 1-1 ATSP System Architecture (PATH Report, 2010)

Although the ATSP system was designed to utilize the existing bus AVL/ACS system for TSP, the field operational test was conducted without this integration, due to the low bus polling rate from the existing AVL/ACS system and the lack of access to AVL/ACS software. Figure 1-2 shows the actual system architecture used for the previous field operational tests. Portable cell-phone-based AVL systems were installed on 15 SamTrans buses to receive and send second-by-second GPS data to the core ATSP processor located at Caltrans District 4. The ATSP processor receives bus GPS data and high resolution signal status and loop data, predicts bus arrival times at prioritized intersections, estimates traffic intersection delay, makes balanced trade-off between transit needs and traffic needs, and sends priority timing (i.e., force-off points) to individual traffic controllers for execution.

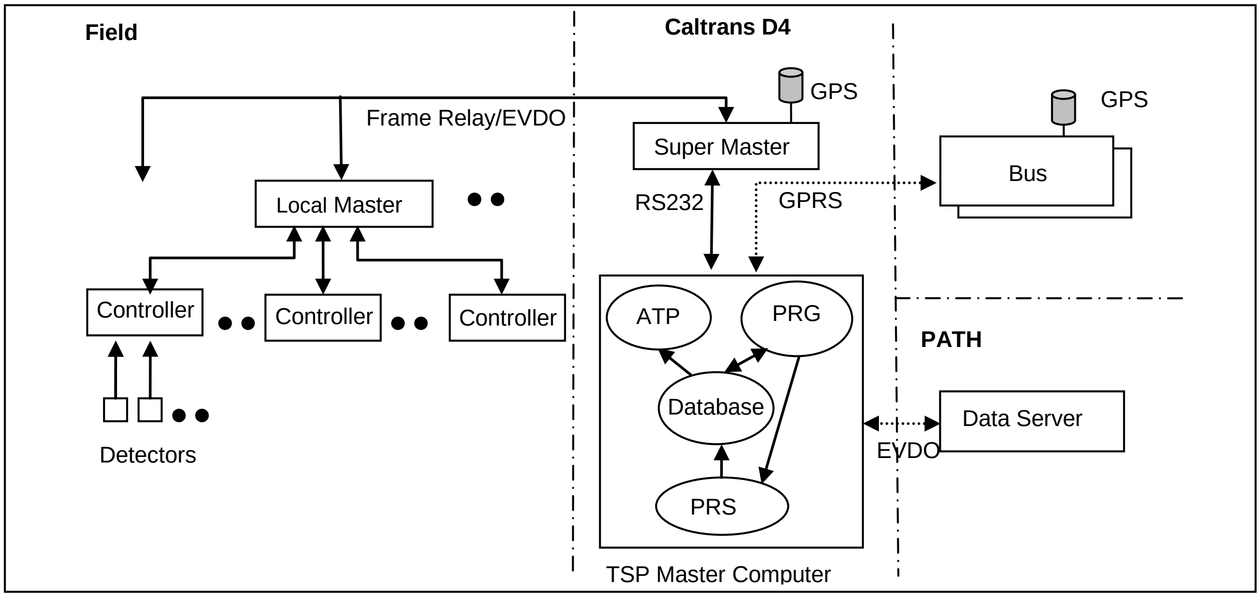


Figure 1-2 ATSP System Architecture for Previous Field Operational Tests (PATH Report, 2010)

AC Transit TSP system is operating on the San Pablo corridor in the Alameda County; VTA TSP system is on El Camino Real corridor in Santa Clara County; and SamTrans ATSP system is on El Camino Real corridor in San Mateo County. Although the traffic signal control systems along the TSP corridors for both AC Transit and VTA are centralized, the two TSP systems are single-intersection based (or distributed), due to the constraints of the deployed commercial products.

Figure 1-3 compares the TSP benefits on transit for the three TSP systems. The selected Measures of Effectiveness (MOEs) include trip travel time, total intersection delay, average delay per stop at signals, number of stops at signals, and average bus running time. For each compared MOE on the horizontal axis, the vertical axis is the percentage change from the “without TSP” scenario to the “with TSP” scenario. As shown in Figure 1-3, all three TSP systems achieved net savings for transit, while the ATSP system had the most benefits. It is worthy pointing out that the ATSP system was applied for local bus routes while the other two TSP systems were for Bus Rapid Transit (BRT) routes. Compared with BRT buses, local buses have lower average travel speed than general traffic and made more frequent passenger loading/boarding activities. TSP net savings on local buses would be considerably smaller than that on BRT buses if the same TSP technology were applied on the same transit route. The gains over the conventional TSP systems would be larger if the ATSP system were applied on BRT routes.

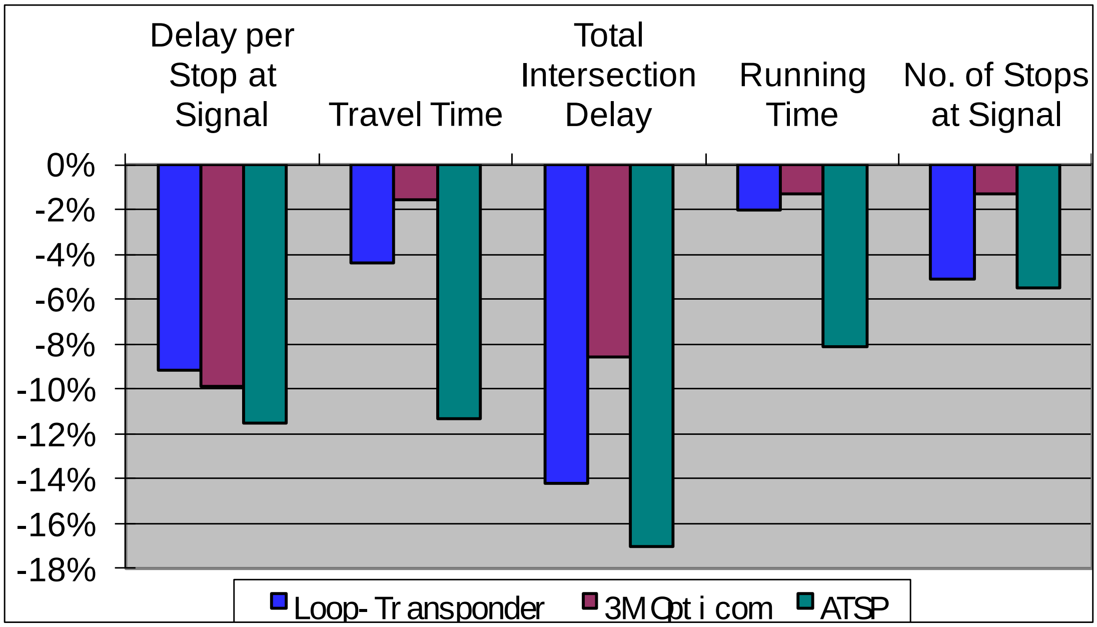


Figure 1-3 Comparison of TSP Benefits

Figure 1-4 compares TSP impacts on traffic for the three TSP systems. All three TSP system provided positive impacts on traffic that go along with the buses and negative impacts on cross-street traffic. The ATSP system, while achieving the most transit benefits, had the fewest negative impacts on the cross-street traffic (Minor Phase Traffic Delay), due to the consideration of traffic impact on the priority decision making.

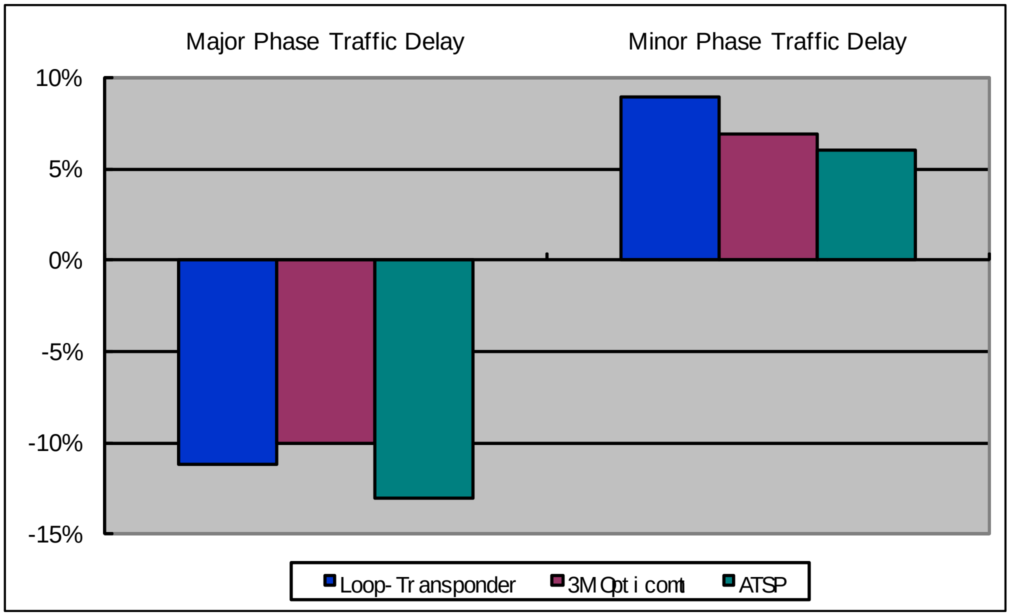


Figure 1-4 Comparison of TSP Impacts on Traffic

A natural extension for the ATSP system is to integrate ATSP and DPI with the existing AVL/ACS, which is the topic of the next subsection.

1.2. Needs and Feasibilities of an Integrated ATSP and DPI system

TSP and DPI have common key features: both require vehicle location information, a communication link between vehicle and infrastructure and capabilities of prediction of bus arrival at fixed geographic locations, i.e., bus stops and intersections. In the late 1980s and 1990s, AVL/ACS systems have been implemented for transit application in the US in order to better monitor and control operations (TRB, 2003a). Although the primary objective of these deployments is to increase operational efficiency, it was recognized that data from the AVL/ACS system could be used to provide real-time predictions of bus arrival at intersections or bus stops, providing the technical capabilities required by TSP and DPI. However, current commercially available TSP and DPI products do not use this information. Instead, when a transit agency wants transit operation management, signal priority, Dynamic Traveler Information (DPI), they will have to go to different vendors, resulting in disintegrated systems and a significant waste of resources. As a result, multiple GPS units, computer hardware and communication links need to be installed on a single bus. Each of these systems costs the transit agencies millions of dollars.

Although both TSP and DPI aimed to improve the level of customer service, they are responsible to different agencies – the local or municipal traffic agency for TSP and the transit agency for DPI. For TSP operations, a bus communicates its location/presence to the local traffic signal controller to request signal priority, and it is up to the signal control to decide how to respond the priority request and grant the priority. For generating DPI, the bus communicates its in-service information (route, direction, run #, location and speed, and schedule adherence) to the transit management center, and the transit management center provides the processed DPI to the customers. As a result, while many transit agencies have implemented both TSP and DPI technologies on their fleets, they are two different commercial products, and most of time, provided by different vendors.

In addition, transit agencies have already widely deployed fleet management technologies before TSP and DPI emerged as crucial ITS technologies – the third commercial product on the transit fleets. The fleet management refers to the processes of planning, supervising, and optimizing the delivery of transit services, and to the maintenance of vehicles providing those services. Computer Aided Dispatch, Automatic Vehicle Location and Advanced Communication (CAD/AVL/CAS) systems, simplified as bus AVL/ACS systems in this report, are examples of technologies that facilitate the management of transit fleet operations, provide up-to-date information on vehicle locations to assist transit dispatchers, and inform travelers of route status (FTA, 2006). By the late 1990s, agencies were generally adopting AVL systems using Global Positioning System (GPS), which became fully operational in 1995.

Multiple ITS systems deployed on the same transit fleet are not integrated systems. Each of these systems costs the transit agencies millions of dollars. As a result of the non-integration, multiple GPS receivers and antennas, computer hardware and communication links need to be instrumented on every single bus, with redundant hardware providing similar functionalities. The nonintegrated systems cause transit agencies not to get the full benefits of its investments in ITS due to unnecessary and redundant use of resources and high ongoing maintenance costs on multiples systems.

There is a clear need from the transit agencies to save their operational and equipment costs. They have expressed strong interest in integrating DPI and TSP with existing transit AVL/ACS systems that most transit agencies have already deployed. A close look at the AVL/ACS, DPI and TSP technologies will also reveal the feasibility of this desired integration.

The core bus AVL/ACS system is defined as (TRB, 2008):

> *"the central software used by dispatchers for operations management that periodically receives real-time updates on fleet vehicle locations. In most modern AVL systems this involves an onboard computer with an integrated Global Positioning System receiver and mobile data communications capability."*

The three ITS systems (the fleet management, the TSP and the DPI systems) have common key elements and features:

- an onboard GPS receiver and antenna for vehicle location positioning;
- a communication link, for example a radio and antenna, between a bus and infrastructure, either the transit management center or the local traffic controllers, for bus location updating; and
- an onboard computer to host either the AVL software for fleet management and/or the bus arrival time prediction software for TSP or DPI.

These three systems can certainly be integrated upon the existing bus AVL/ACS systems, with a single onboard GPS receiver and antenna, a single communication link, and a single onboard computer. Vehicle positioning data from bus AVL can be used to generate predictions of bus arrival times at signalized intersections and at bus stops and predicted arrival times can be uploaded to infrastructure through bus ACS, providing capabilities required by TSP and DPI.

The National Transportation Communications for ITS Protocol (NTCIP) standard 1211, *Object Definitions for Signal Control and Prioritization (SCP)* (AASHTO/ITE/NEMA, 2008), was developed to establish standards for use in implementing TSP applications within traffic signal systems. It addresses four SCP scenarios and use cases that can be used to provide a logical architecture for implementation of TSP. Based on NTCIP 1211 standard, the Transit Communications Interface Profiles (TCIP) Standard recommends five SCP scenarios for the implementation of TSP in different transit and traffic operating environments (FTA, 2008). Of those five scenarios, three are for centralized TSP systems and two for distributed TSP systems. Most deployed TSP systems in the U.S. are distributed systems, where additional onboard and roadside devices are required to establish the communications link between a bus and an individual traffic signal controller. A centralized TSP architecture is desired for the integration approach, as with bus AVL/ACS system a bus is communicating with the transit management center not directly with a traffic signal controller. With center-to-center communications, priority requests can be communicated from buses to the transit management center, then be forwarded to the traffic management center, and finally be granted at the local traffic signal controllers. Given the advances in ITS and communication technologies, more and more traffic agencies have upgraded their systems to be centralized, particularly in response to the needs of arterial performance measure and integrated multi-corridor management.

The integrated approach is the most cost effective move to include the TSP and DPI features on the existing bus AVL/ACS systems. It not only minimizes the redundancy and saves transit agencies' investments and maintenance costs but also can provide greater efficiencies.

- The integrated system allows the three ITS technologies to use the same set of data in seamless fashion and avoid the burdens on transit technology personnel of integrating data that are spread throughout various applications in different agency departments, leading to a more supportable system. For example, while the transit agency owns the GIS database of its transit routes and stops, the GIS information about locations of signalized intersections and signal timings are owned by the traffic agency. The integrated data source will help transit agencies to more easily identify the operational problems and the corresponding ITS technologies to address the problems.
- Intersection delay accounts for about 20 to 30 percent of total bus trip time. TSP is capable to reduce this delay thereby producing more reliable travel time between bus stops. The integrated approach allows the DPI technologies to utilize TSP information in generating more accurate arrival time predictions at downstream bus stops.
- TSP operation, in general, requires more frequent bus location updates than DPI. The additional available bus location update will allow the DPI to provide more timely updates of arrival times at bus stops.

1.3. Technical Challenges of Integrating ATSP and DPI

The ATSP system communication is already center-to-center, therefore the aforementioned requirement of center-to-center communications for the integration of ATSP/DPI with AVL/ACS would be easier to implement. Furthermore, ATSP already provides the key features for DPI, so extending the ATSP system to include DPI features and to develop an integrated ATSP/DPI system with the existing bus AVL/ACS system becomes more logical and natural.

While there is great potential in integrating ATSP and DPI upon the existing bus AVL/ACS systems, technical challenges exist. A major technical challenge on the development of an integrated ATSP/DPI system is to overcome the low bus polling rate from AVL/ACS system to support ATSP operation. Under the bus AVL/ACS system, although the onboard component - Advanced Mobile Data Terminal (AMDT), continuously determines bus location and tracks schedule and route adherence in real-time, the data about location, schedule and route adherence status and other status are only transmitted to ACS server at the Operations Control Center (OCC) – dispatch, on a frequent periodic basis (polling rate). Bus AVL/ACS technologies use center-polling communication protocol for radio data, where the center (ACS server) performs carrier sense and collision detection to make sure that the carrier is available and issues a polling command and each node (bus) can only communicate with the center when polled. The center polls one bus at a time and the polling rate refers to the time interval that all vehicles in the fleet are polled. The number of radio data channels needed for a bus AVL/ACS system depends on factors including the capacity of the mobile data communications system, the fleet size, polling rate, and the efficiency of the polling algorithm used in a particular AVL/ACS system. Data channel for polling periodic bus location and status reports usually utilizes 60 to 70 percent of the channel capacity of current AVL/ACS systems. The remaining 30 to 40 percent of channel capacity is used as a contention channel for meeting other application needs, such as voice communications and emergency data messages to dispatch. For SamTrans AVL/ACS, there are six time slots per second, four time slots are used as data channel and the other two as contention channels. The contention channel can also be used for polling bus location and status reports but the current AVL/ACS systems do not utilize it for bus polling. As radio channels have a low data transmission rate, commonly at 9.6 kbps, the polling rate for periodic location and status reports on any particular bus is about 1 to 2 minutes. This low polling rate makes it not practical to use existing polling algorithms for the generation of transit signal priority requests, as a bus could travel across a couple of intersections within one polling cycle or is already waiting in the traffic queue at an intersection or loading/boarding passengers at a bus stop when polled.

To overcome this low polling rate challenge, there is a need to modify the AVL/ACS polling algorithm to support the ATSP features. Ideally, changing the frequency-based polling to need-based polling could resolve the low polling rate issue. Buses that need signal priority get higher polling rates while other buses in the fleet get lower polling rates but still can meet an agency's requirements on fleet management applications. However, this would require significant changes on the existing polling algorithm. A more practical approach is to extend the existing polling algorithm to include a parallel, need-based dynamic polling algorithm for inserting additional polls that are carrying out through contention channels to serve transit signal priority purposes.

In the previous project on the development of ATSP system (PATH Report, 2010), a dynamic polling algorithm was proposed based on the existing ACS communication protocol. The dynamic polling algorithm includes the following steps:

1) Each bus continuously estimates its time-to-arrival (T2A) at signalized intersections;
2) When polled (either a regular or an inserted poll), the predicted T2A along with location and status update are transmitted to dispatch;
3) The dispatch maintains a dynamic need-based polling schedule depending on received T2A and schedule adherence status. An inserted polling is carried out though contention channels when T2A countdowns to within a predetermined time window, which is desired for making priority request. Only behind schedule buses are considered for insert polls.

The dynamic polling algorithm was validated using simulation methods. Six-month operations data were collected for 15 SamTrans bus routes that travel along part of El Camino Real. In the simulation study the T2A time window for inserted polls is preset as 40 seconds. The frequency of the number of inserted polls exceeding the capacity of the contention channel was examined (i.e., more than 2 inserted polls within 1 second duration). Figure 1-5 illustrates the histogram of frequency of inserted polls in a typical day of bus operation and at any given 1-second interval. Over 76.5% of time there is no need for inserted polls; about 20.5% of time there are up to 2 inserted polls which can be served by a contention channel within the 1-second interval; only less than 3% of time there are more than two inserted polls and part of those have to be served by contention channel available in the following 1-second intervals.

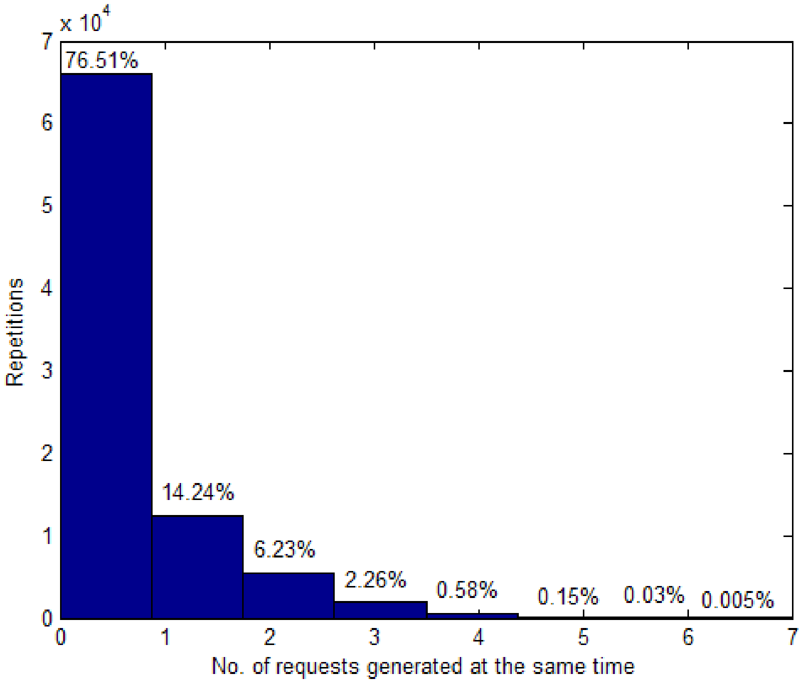


Figure 1-5 Histogram of Frequency of Inserted Polls

Assuming an inserted-polling request will be timed-out (dropped) if not being served within 20 seconds, Figure 1-6 shows the cumulative numbers of inserted poll generated and processed. The total number of dropped insert-polling requests is less than 20 for the whole day and for all 15 bus routes. The number of dropped insert-polling requests is 0 when only the two bus routes (SamTrans 390 and 391) traveling along the implemented TSP corridor on El Camino Real are considered.

The simulation study has demonstrated it is viable to implement the dynamic polling algorithm to serve ATSP feature upon the existing AVL/ACS systems. Realizing the dynamic polling algorithm for the integrated ATSP/DPI system with AVL/ACS remains one of the critical challenges for the integrated approach.

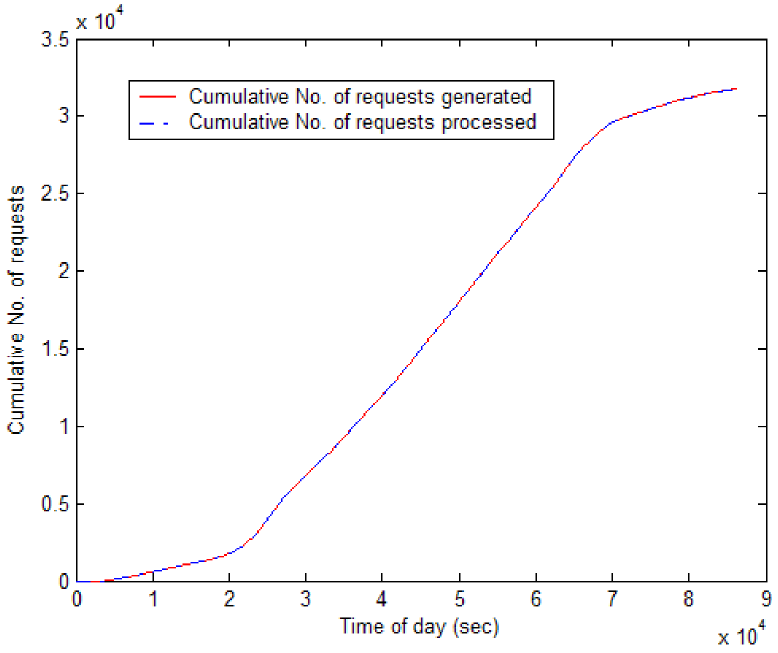


Figure 1-6 Cumulative Numbers of Inserted Poll Generated and Processed

Extending the capability of the bus arrival time (T2A) prediction algorithm for DPI presents another technical challenge for integrating ATSP and DPI. T2A for ATSP aims at short-term prediction (to the downstream signalized intersection) while T2A for DPI aims at long-term prediction (to all downstream bus-stops). Different prediction models would be required to meet the different needs for ATSP and DPI features.

An integrated ATSP/DPI system uses a wide range of data as inputs, including both historical and real-time transit and traffic operations data and digital road network data among others, and produces widespread outputs. Flexible and efficient data management and processing impose other technical challenges for integrating ATSP and DPI, which are discussed in the next subsection.

1.4. Project Objective and Integration Approaches

The objective of this project is to develop an integrated ATSP/DPI system on hardware and communications link already deployed by transit agencies. The primary purpose is to provide TSP at prioritized intersections and real-time transit information to transit customers using one integrated ATSP/DPI system.

1.4.1. Based on Existing Transit AVL/ACS System

The major aspect of the technical approach is to build the integrated system based on existing transit AVL/ACS systems. Among these AVL/ACS systems that have various different vendors and different technical specifications, we conducted research in this project to identify the requirements of the AVL/ACS for the ATSP/DPI integration. Based on these requirements, we developed the technologies to integrate these systems.

Among various AVL/ACS systems, the low bandwidth ACS system that was deployed at our project partner, SamTrans, is a typical representation, and has thereby been used as the basis of the project.

1.4.2. Adopting an Open Architecture

Transit technologies need to take a holistic approach that will address transit operation needs and will support integrated and sustainable transit ITS deployment. To accomplish this goal, an open, layered and scalable architecture is advantageous. Openness will allow a range of existing and new technologies to be integrated, and also define a set of protocols for information exchange between layers. The layered architecture structures functions at different levels, allowing the independent functions grouped into a predetermined category to support higher level functions or applications. Scalability enables the system to easily expand applications using the existing functions over time. The proposed architecture integrated AVL/ACS and DPI by modifying the existing architecture to include the following architectural layers:

- **Data collection layer** that includes either a central database or centrally managed distributed databases;
- **Data repository layer** that includes a shared data repository;
- **Process layer** that consolidates various common data processing functions and services needed to support various business applications;
- **Application layer** that reassembles existing business processes in operation, planning, maintenance, meeting SamTrans' current and future transit operation requirements; and
- **Interface layer** that enables interaction between operators and systems, between transit system and traffic control system and between travelers and systems.

1.4.3. Develop an Integrated Database

Archiving and managing the large set of data efficiently is very challenging since there are many different types of transportation data, including the traffic data, transit data, freeway data, underlying road network data, detector data, and application data. The data from these different sources is strongly related to each other. Moreover, the amount of traffic and transit data is substantial. Hence, our goal is to design an integrated, scalable and efficient ITS database.

In order to improve flexibility and reduce complexity, we decomposed the data management into several layers. The lowest layer is the underlying road network. The second layer is designed for the static traffic, transit, and parking data, while the dynamic traffic, transit and parking data is handled in the third layer. The fourth layer is designed for applications, such as transit signal priority, integrated corridor management, and dynamic passenger information system.

1.4.4. Develop Integrated Data Processing Algorithms for ATSP/DPI

A different requirement of the data processing by DPI and ATSP also imposes a need to develop an integrated data processing method for integration. Bus arrival time estimates to the DPI server aims at long-term prediction where the bus is far away from the bus stop. Arrival time estimates to the ATSP server aims at short-term prediction where the bus is within 20 to 30 seconds of arrival at the intersection. This difference posed different requirements on the inputs to and the outputs from the prediction algorithm. The regular 2-minute polling rate could meet the purpose of serving DPI but does not support ATSP operation. A dynamic polling algorithm needs to be developed and implemented at the operational level. Desirably, having the vendor of the existing bus AVL/ACS system to integrate the dynamic polling algorithm with its regular polling algorithm would achieve a fully operational and integrated system. It requires the vendor to modify its communication system and to integrate the prediction software module with the onboard components. In the lack of access to both onboard and center ACS software, an ACS polling emulator was developed to mimic the operations of the regular polling algorithm. Integrated with the dynamic polling algorithm, it served as a "pseudo" AVL/ACS system. Upon the validation through the field operational test, the developed dynamic polling algorithm will be ready and can be transferred to the vendor for the deployment on its existing system.

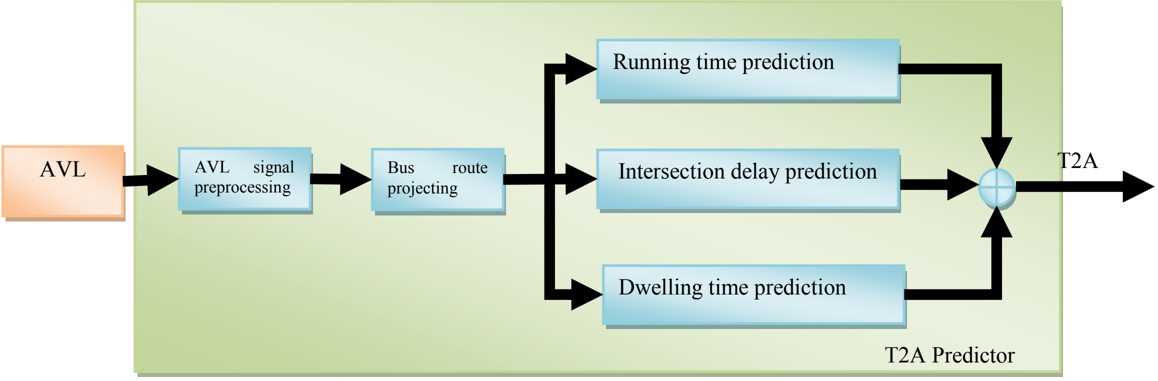


Figure 1-7 Structure of an Integrated Time-to-Arrival Estimator

Figure 1-7 shows the block diagram of a T2A predictor. As the operations characteristics vary throughout bus route, different models will be developed to achieve the best prediction accuracy.

1.4.5. Using Emulated Dynamic Polling to Validate the Viability of ATSP/DPI Integration

Emulated dynamic polling, instead of actual dynamic polling, was used in this project. The added function of actual dynamic polling would require making changes to the onboard software, which is not accessible by the project team at this time. Therefore an alternative approach of using emulated polling based on high frequency GPS is used. The dynamic polling algorithm is then implemented on the processing center instead of an onboard computer.

1.5. Overview of an Integrated ATSP/DPI System

Under this project, California PATH has implemented an integrated ATSP and DPI system with emulated dynamic polling under partnership with SamTrans. The system has the following features:

1) The system is implemented based on the SamTrans AVL/ACS system with emulated dynamic polling. There are 15 buses that each have an equipped portable AVL device to receive and send second-by-second GPS data; those are only used as (a) ground truth data for the purpose of performance evaluation but not operations and (b) emulation of ACS regular and dynamic inserted polling for ATSP. Second-by-second GPS is not required as part of the system for future deployment.

2) Dynamic passenger information is provided for all SamTrans bus routes regardless of whether the bus is running along the TSP arterial or not. The T2A prediction module for DPI is treated as running at the fleet management center. Inputs to DPI prediction module are the actual polling data received from bus AVL/ACS system.

3) Transit signal priority is provided to the 15 portable AVL equipped SamTrans buses when they are serving routes 390 and 391. The T2A prediction module for ATSP is hosted at the traffic management center (Caltrans District 4) and serves as the “pseudo” onboard prediction model. Inputs to the ATSP prediction model are the fused second-by-second GPS data and actual polling data received from bus AVL/ACS system (second-by-second GPS data do not include bus current running route, direction and schedule adherence information which are part of the ACS polling data). Predicted T2A information is not directly used for the generation of priority requests. Rather, it is discretely sampled by ACS regular and dynamic polling emulator at times either a regular poll or an inserted poll needs to be carried out.

   We note that the implementation of the dynamic polling algorithm requires certain modification of the onboard processing software of the ACS system. This modification was not doable due to the lack of access to the onboard software. Therefore second-by-second GPS data is used for emulation of dynamic polling for onboard processing.

4) Returns from the inserted polls are fed into the DPI prediction module as additional bus status updates.

In the next section we will describe in detail the design and implementation of the integrated system.

2. ATSP/DPI System Requirements and Architecture

2.1. Understanding Transit Operations and Environments

Successful integration of ATSP and DPI systems highly depends on the transit operations and environment. In this subsection, the details of transit operations for the SamTrans ATSP routes are analyzed to form the basis for integration.

There are two types of bus stops: non-time-point stops and time-points. The difference between them is that, at time-points, a transit vehicle can arrive before - but not leave earlier than (so called time-point holding) - the stated time as indicated in the route schedule. SamTrans routes 390 and 391 provide schedule-based transit services, where time-point holding discipline is applied.

Table 2-1 lists the number of stops, number of time-points and route length for SamTrans routes 390 and 391, which provide transit service along El Camino Real and cover the ATSP arterial.

Table 2-1Transit Stops and Time-Points

Transit Route	Length (miles)	Direction	Origin	Destination	Total No. of Transit Stops	No. of Time-Points
Route 390	27.2	NB	Palo Alto T.C.	Daly City BART	97	**11**
		SB	Daly City BART	Palo Alto T.C.	100	**11**
Route 391	30.3	NB	Redwood City T.C.	San Francisco	105	**13**
		SB	**San Francisco**	**Redwood City T.C.**	**105**	**13**

SamTrans operates multiple services (patterns) along a one-way transit route, with different origin-destination (O-D) pairs. For example, the weekday SamTrans 391 northbound service includes 4 patterns: 1) Redwood City Transit Center to San Francisco Transbay Terminal, 2) Redwood City Transit Center to Mission and Evergreen (Daly City), 3) Hillsdale Blvd at El Camino Real (San Mateo) to San Francisco Transbay Terminal, and 4) Millbrae Transit Center to San Francisco Transbay Terminal. The capability of the integrated ATSP/DPI system in identifying the on-going transit services is important for providing travelers accurate transit information as different services have different O-D pairs, stop patterns and schedules.

Buses share the roadways with general traffic. In the design of route schedule, the expected route travel time (for example, the 85-percentile traffic travel time) is combined with the slack time, leading to schedule stability. If the slack time is insufficient, transit vehicles are unlikely to catch the schedule when falling behind, thereby downgrading the service reliability. On the other hand, large slack time reduces service frequency and increases transit waiting time and travel time.

Figure 2-1 and Figure 2-2 illustrate the scheduled travel time to time-points for northbound and southbound SamTrans route 390 trips, respectively. They clearly show that route schedules match with the traffic patterns. The scheduled travel time to time-points is shorter in the early morning and late evening, when traffic is lighter, and is longer during the rush hours.

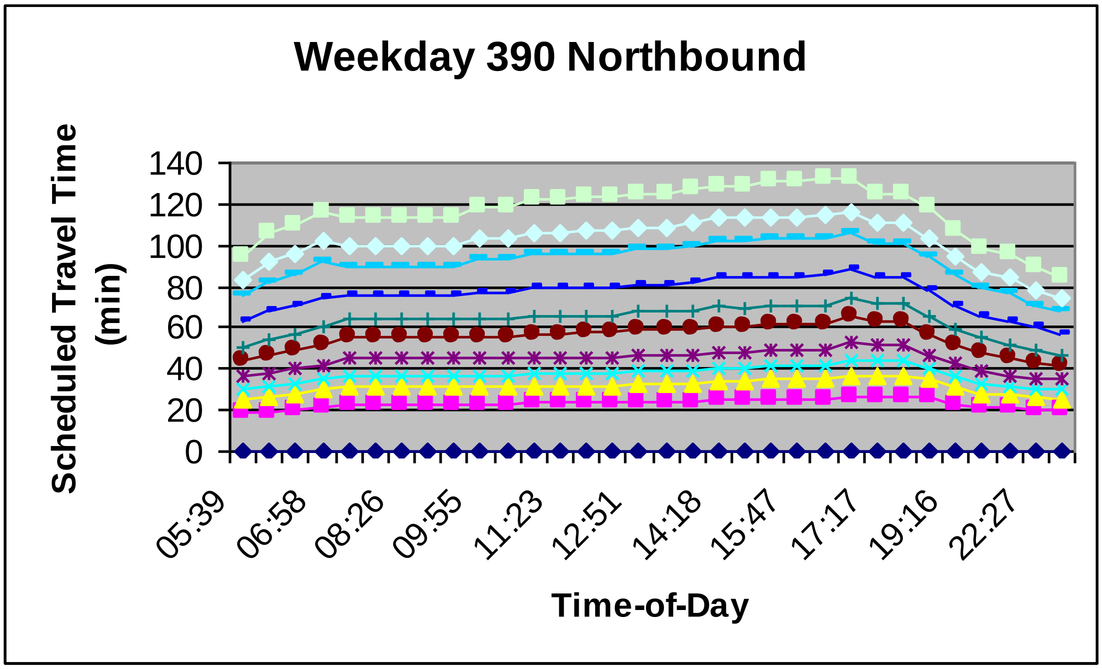


Figure 2-1 Scheduled Travel Time to Time-Points (Northbound)
(SamTrans Route 390 Weekday Trips)

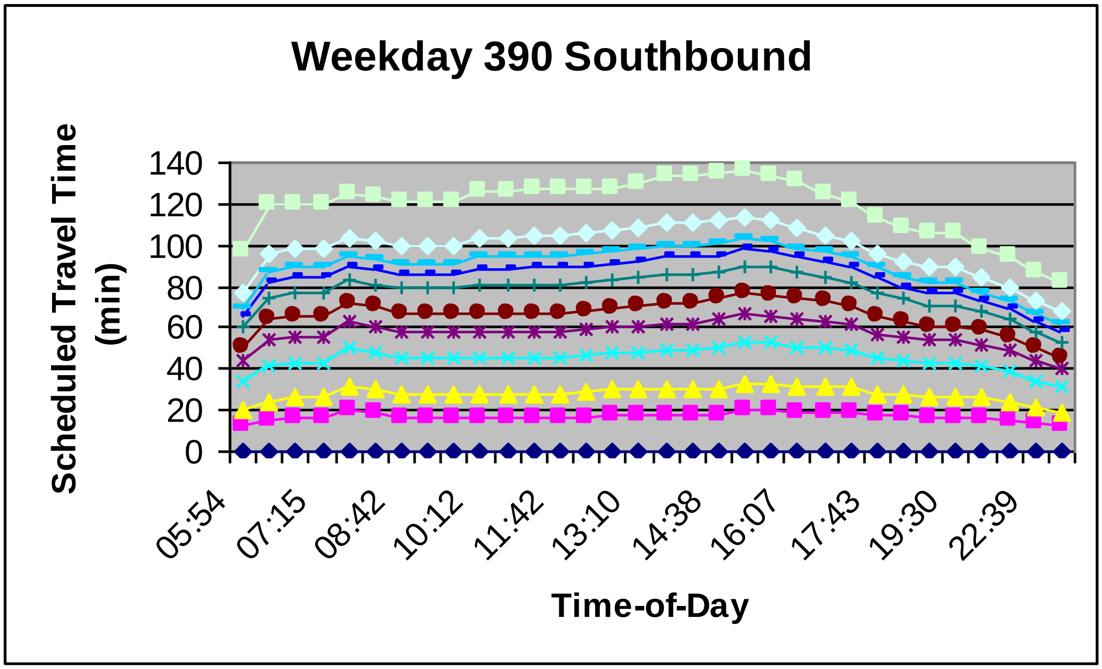


Figure 2-2 Scheduled Travel Time to Time-Points (Southbound)
(SamTrans Route 390 Weekday Trips)

When utilizing ACS for ATSP and DPI features, the impacts of the ACS polling rate need assessment. The expected travel time between two consecutive transit stops is obtained as a linear interpretation of route schedule and route length. Figure 2-3 shows a histogram of the above referenced expected travel time, for SamTrans 390 northbound trips. With a 2-minute polling rate with the current AVL/ACS, a bus often has passed through a couple of stops.

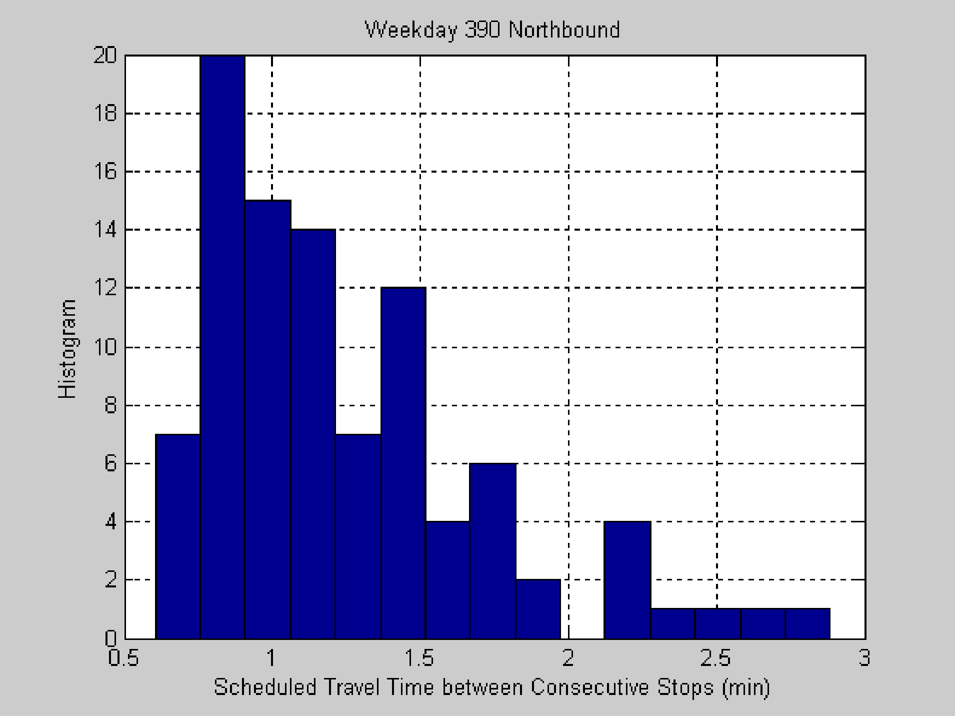


Figure 2-3 Scheduled Travel Time between Consecutive Bus-Stops (SamTrans 390 NB Weekday Trips)

With the low polling rate, there is a crucial problem in providing travelers accurate transit information. Before receiving the next bus location update, the DPI may indicate the bus is approaching or waiting at a stop, while in fact the bus has already left the stop. The inserted polls generated by dynamic polling for ATSP purposes can provide additional bus location updates and thereby allowing a prompt departure time update.

SamTrans bus movement data were collected via portable AVL devices instrumented on 15 buses. Out of the 34 runs on weekday northbound route 390 services, one run was selected for this analysis. The selected run is scheduled to leave the origin (Palo Alto Transit Center) at 2:18 PM and arrive at the destination (Daly City BART) at 4:27 PM. The expected trip time is 129 minutes. A total of 105 bus trips were extracted from archived data. Bus trajectories are plotted in Figure 2-4. The small red circles in the plot indicate the locations and schedules for the 11 time-points.

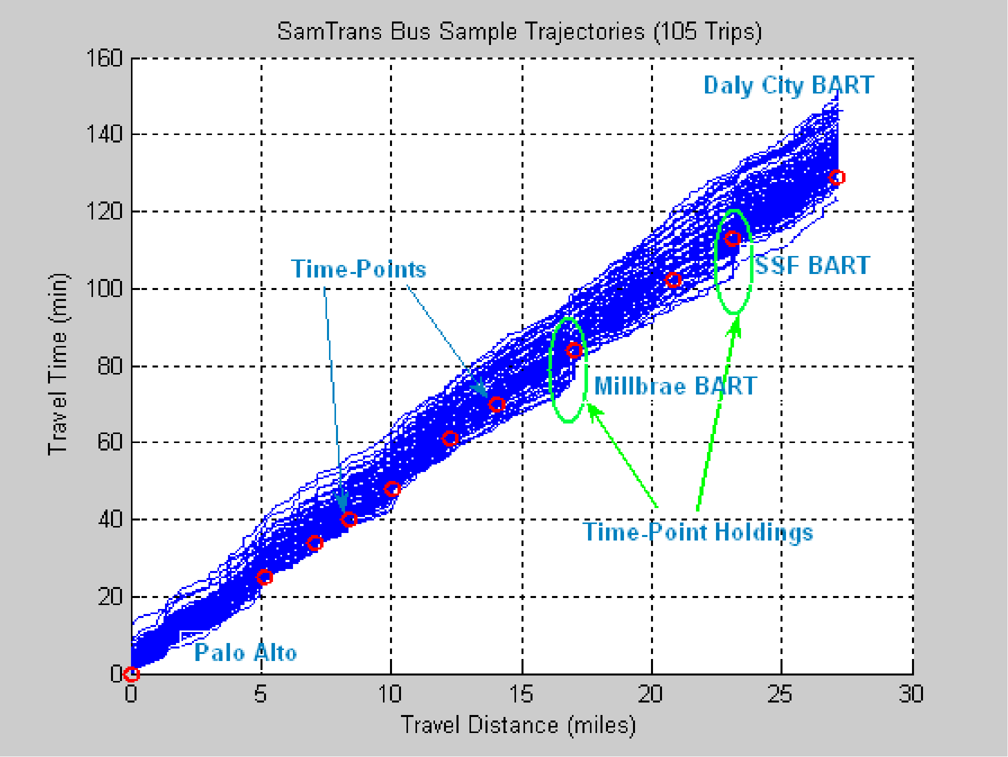


Figure 2-4 Sample Bus Trajectories

Time-holding phenomenon can be observed at multiple time-points, including stops at the Millbrae Transit Center (Millbrae BART) and the South San Francisco (SSF) BART station. The longest waiting time observed at the Millbrae BART stop is about 6 minutes and that at the SSF BART stop is approximately 10 minutes.

Buses are likely to be behind schedule, as is shown in Figure 2-5, which plots the box-plots of schedule deviation (i.e., the difference of actual departure time from schedule) at each time-point. The medians of schedule deviation do not form a straight line, indicating the existence of both the schedule recovery and delay propagation processes. The variances of schedule deviation become larger when further into the trip, indicating bus delays at far-apart time-points are not so strongly correlated.

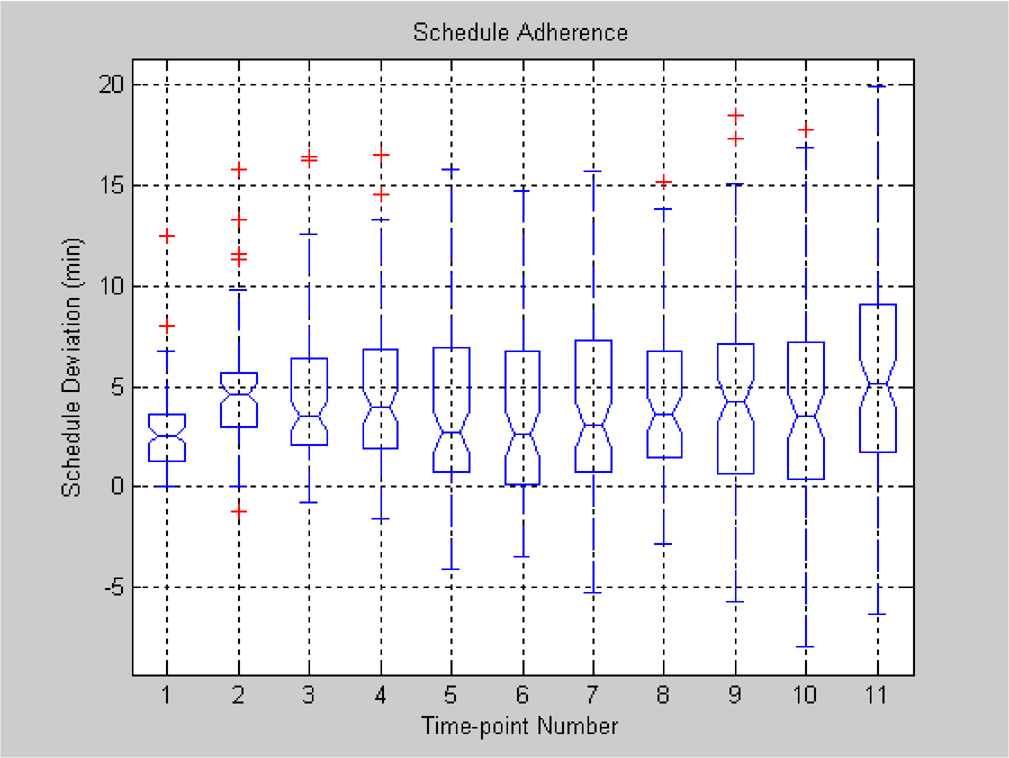


Figure 2-5 Bus Schedule Adherence

2.2. An Integrated ATSP and DPI System

2.2.1. System overview

The primary purpose of the integrated ATSP/DPI system is to utilize the existing AVL/ACS to request and obtain signal priority in real-time at prioritized intersections and to provide real-time transit information to passengers. The information about ATSP operation status is not directly provided to passengers but transit passengers will experience ATSP benefits by reduced intersection delay and bus trip travel time. The transit information includes real-time location of transit vehicles, the estimated arrival times, thereby the next vehicle arrival times, for selected transit stops. The transit information would be accessible via the Internet, smart phones, as well as electronic information display boards at key transit stops.

The integrated ATSP/DPI system is to include the following characteristics:

- Utilize the existing transit AVL/ACS system for vehicle location and tracking;
- Utilize the existing traffic signal control infrastructure for priority control;
- Dynamically poll bus location updates for buses need signal priority and automatically generate signal priority requests to the traffic signal control systems;
- Provide timely transit traveler information to customers;
- Provide transit traveler information that can be used by travelers for pre-trip planning and en-route guidance; and

- Provide transit travel information to transit agency's administrative and management staff for system administration, maintenance and statistical analysis.

The operation of the integrated ATSP/DPI system interfaces with two existing systems, the transit AVL/ACS system and the traffic signal control system. The transit AVL/ACS system provides real-time bus reporting messages that include bus GPS location, schedule and route adherence status, and the predicted arrival time at the next signalized intersection to the ATSP/DPI system, while the traffic signal control system provides real-time loop counts and signal phase and timing (SPAT) information to the ATSP/DPI system. The ATSP/DPI system use data feeds from these two systems to

1) assess movement-based traffic intersection delay and bus intersection delay;
2) determine which buses are in the need of signal priority and generate dynamic inserted polling requests for timely location updates from those buses;
3) generate priority requests based on responses from inserted polls and send to the traffic signal control system for ATSP control;
4) predict bus arrival times for all existing and future transit stops through the transit network, using bus reporting messages from both regular and inserted (when available) polls; and
5) generate and disseminate transit traveler information in real-time.

The traffic signal control system is responsible for providing real-time loop counts and SPAT data feeds to the ATSP/DPI system and executing TSP control upon receiving priority requests from the ATSP/DPI system. The transit AVL/ACS system is responsible for providing real-time bus reporting messages to the ATSP/DPI system and carrying out dynamic inserted polling requests with ACS. Current ACS polling does not implement the dynamic polling features and the existing regular polling has a low update rate (around 2 minutes from each bus in the fleet). In addition, current bus reporting messages do not include information about the predicted arrival time at the next intersection. Therefore, there are two modifications needed on the existing AVL/ACS system to support the operation of integrated ATSP and DPI, 1) adding an onboard software module to predict bus arrival time at the next intersection and expanding the message size to include the predicted information, and 2) adding dynamic polling feature with the ACS server.

Due to the constraints of getting access to the existing ACS onboard and server processing software, this project adopted an alternative approach – emulated dynamic polling instead of actual ACS dynamic polling, to test the integrated ATSP/DPI system. Under the emulated approach, the bus arrival time prediction algorithm was implemented on the central processor of ATSP/DPI instead of an onboard computer; selected buses were instrumented with portable AVL systems to record and send second-by-second bus GPS location updates; and the dynamic polling algorithm was implemented on the ATSP/DPI central processor with emulated ACS operation rather than on the ACS server with actual ACS operation.

2.2.2. **System Functions**

The functional description of the integrated ATSP and DPI system is depicted in Figure 2-6. A principal function element is to process input data, including traffic and transit network data. This function must be able to match bus movements with transit route, schedule and road network. A second functional element is the decision-generation function that: 1) generates T2A prediction at bus stops and signalized intersections, 2) determines when and to which bus for dynamic insert polls, and 3) determines signal timing for transit priority. The third element is the decision presentation, which communicates TSP requests to traffic controllers, dynamic polling requests to ACS, transit traveler information to customers via the Internet and smart-phones, and displays bus arrival time at bus stops.

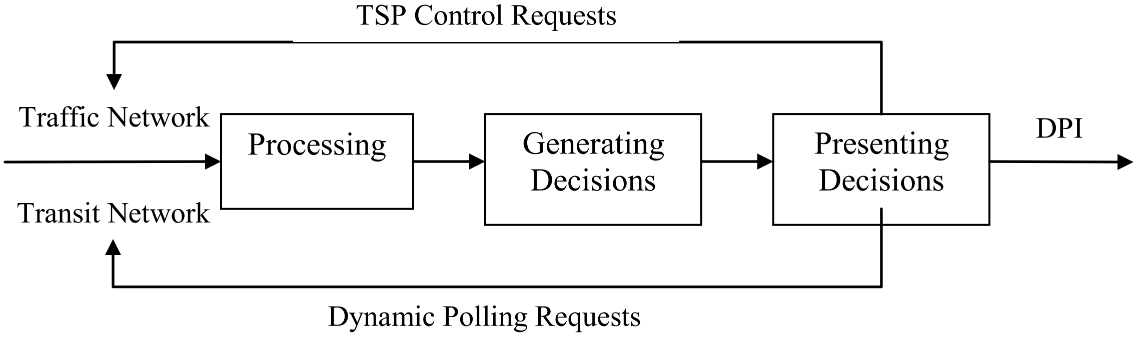


Figure 2-6 Integrated ATSP/DPI System Functions

Transit data inputs are categorized into *static data* and *real-time* data. The static data include published route schedules, transit route network and stop GIS data sets. The static data shall be currently in use and updated when any changes have been made. The real-time data for buses are the operation data from the bus AVL/ACS system. The data shall include the information about the route the bus is currently serving, schedule adherence status, location, speed and heading.

Traffic data inputs also include *static* and *real-time* data. The static data shall include locations of signalized intersections and control timing plans. The real-time data shall include loop counts and signal phasing and timing information.

2.3. System Architecture

2.3.1. **Background of System Architecture Design**

A basic component of the DPI system is the advanced vehicle location (AVL) system. To collect the data from the AVL system to generate predictive arrival times, advanced communications system (ACS) is also needed. Currently at transit agencies, there are already multiple AVL/GPS systems that exist for different operational systems, including the AVL for DPI and AVL/ACS for transit operation management system, etc. Therefore multiple computer hardware and communication links need to be (or have been) installed on a single bus.

Generally speaking, the advanced traveler information system has two categories: (1) pre-trip and (2) en route traveler information system (Adler & Blue, 1998). Both the two categories can benefit from real-time transit information to enhance the information quality by incorporating more accurate bus arrival time predictions (Grotenhuis, Wiegmans, & Rietve, 2007). Basic information that real-time DPI system provides include the real-time predictive bus / train arrival (departure) times (Shalaby & Farhan, 2003), as well as other information such as disruptions, etc. This information can be delivered via traditional methods such as information kiosks at bus stops, or via web browser and mobile devices to support both pre-trip and en route usage.

The evolvement of the information delivery methods, as well as the rich content of the DPI impose the need to explore an open DPI system architecture, which is able to

(1) Utilize the existing AVL/ACS system to minimize the cost of deploying and maintaining the system;

(2) Facilitate the presentation of the real-time traveler information via different means, including kiosk display at bus / train stations, web interface or mobile phone clients; and

(3) Provide scalable services with open architecture so that information will be delivered with multiple media.

The requirements lead to a design of an open system with layered architecture that can facilitate existing communication systems and provide real-time information on-top of that as well.

Figure 2-7 is the proposed transit ITS system architecture.

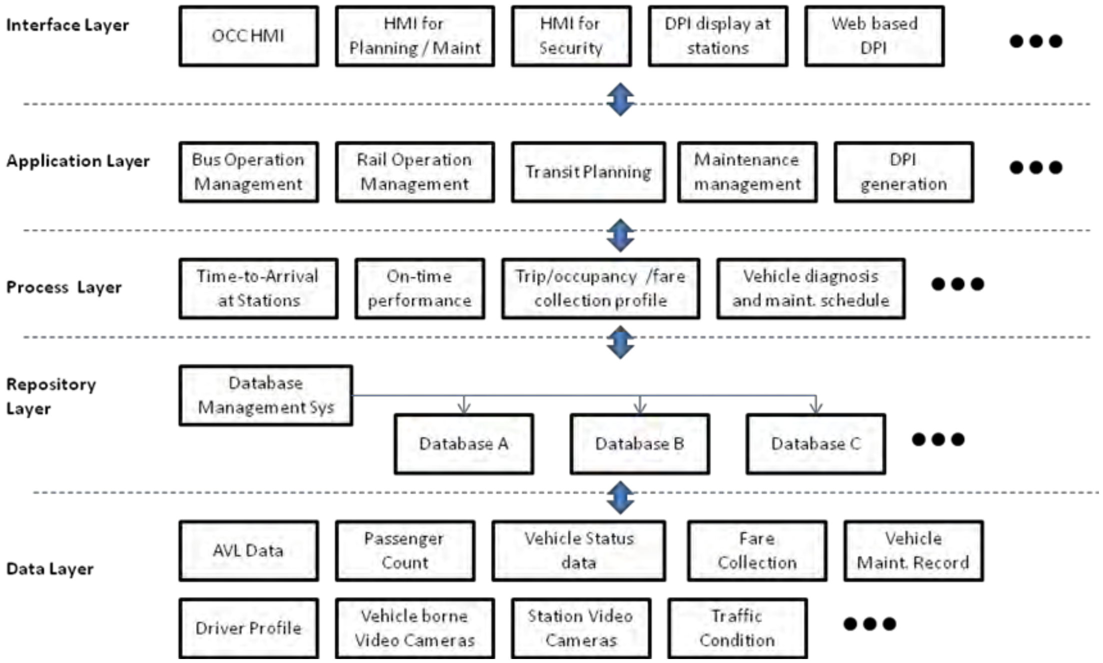


Figure 2-7 ITS System Architecture

The rationality in a layered open architecture has the following benefits that it

- allows interconnectivity and interoperability and
- provides for modularization and scalability.

In the following subsections we will describe the system architecture for the integrated ATSP/DPI system within the proposed framework of transit ITS system to meet with the requirements of the project. The reason that the ATSP/DPI system itself also needs to fit in a bigger general architecture is so that other tasks such as transit operation management, transit planning and transit maintenance management can also be integrated into an interoperable ITS system.

2.3.2. **Overview of System Architecture**

The integrated ATSP/DPI system has the following typical components (Figure 2-8):

- Onboard AVL system which obtains transit vehicle's location, using GPS devices;
- Wireless communication system which provides two way linkage between a dispatch center and transit vehicles (buses or trains);
- Traffic signal control system which obtains real-time information about signal phase and timing and traffic counts and provides TSP control;
- Database system which archives static transit schedule and route information, static traffic signal controller control-plan parameters, real-time transit AVL data and traffic condition data as well as the generated ATSP request data and DPI information data;
- Central processor which aggregates the data and generates estimated time to arrival for transit vehicles, generates ATSP requests, generates the DPI information for various information processes (for personal information, bus stop, etc) and optimizes the routes for trip planning; and
- Transit server that provides services for information in various formats and via different media.

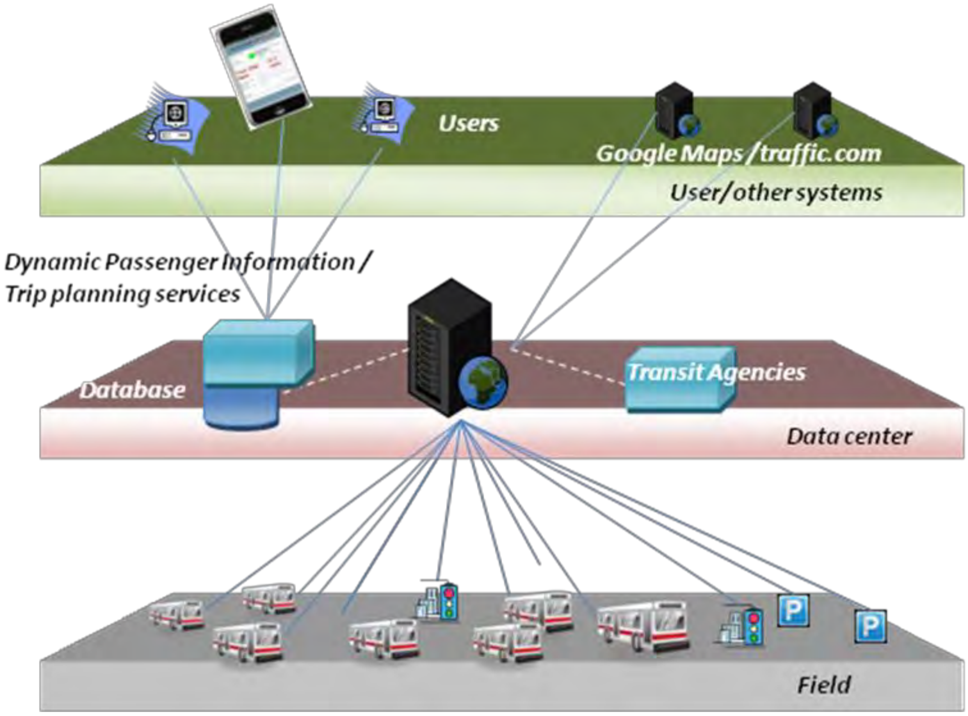


Figure 2-8 System Components

In summary, the system is designed to have a scalable architecture to meet different needs from existing systems of transit agencies. The design of the system architecture supporting several different scenarios, is scalable, and takes advantage of existing AVL/ACS system as well as the existing real-time information system to provide a highly flexible solution for the DPI system (see Figure 2-9).

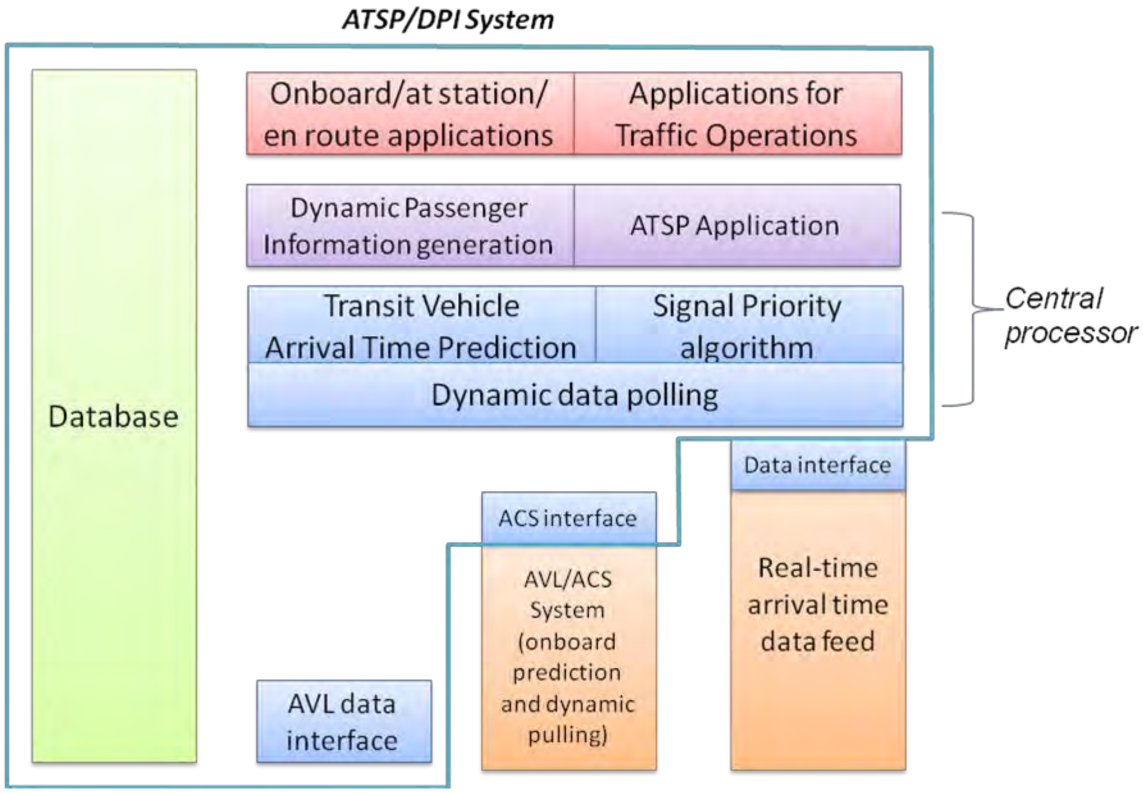

ATSP/DPI System
Database
Onboard/at station/ en route applications
Applications for Traffic Operations
Dynamic Passenger Information generation
ATSP Application
Transit Vehicle Arrival Time Prediction
Signal Priority algorithm
Dynamic data polling
Central processor
Data interface
ACS interface
AVL/ACS System (onboard prediction and dynamic pulling)
Real-time arrival time data feed
AVL data interface

Figure 2-9 ATSP/DPI System Architecture

2.3.3. Data Source

The data sources that an integrated ATSP/DPI system needs fall into the following categories:

(1) Geographical data that describes the road network and other related information to transit service routes;

(2) Static data for scheduling of transit services, including definition of routes, service types, stop sequences, time points, schedules, etc.;

(3) Static data for signal phase and timing for intersections;

(4) Real-time operation system data such as the time keeper data from transit buses, real-time signal phase and timing from intersections; and

(5) Real-time AVL data from the buses.

These are the data elements that serve as the input to the integrated ATSP/DPI system. Details of the data elements are listed in Table 2-2.

Table 2-2 Data Inputs to the Integrated ATSP/DPI System

Data elements		Application Support	Static / Real-time	Description
Geographical data	Road network (nodes and links)	ATSP + DPI	Static	
	Map-matched location of TSP intersections	ATSP	static	For TSP operation
	Bus stops and virtual points for paths	ATSP + DPI	Static	For arrival time prediction, both long term and short term
Scheduling data	Route file	DPI	Static	Route direction, head sign, description
	Block file	DPI	Static	Definition of blocks
	Time point file	DPI	Static	Time point stops
	Service schedule file	DPI	Static	Schedules at time points for each route and their corresponding service days
Signal phase and timing	Signal phase and timing table	ATSP	Static	Signal plans for the intersection
	Real-time signal phase and timing data	ATSP	Real-time	For short term prediction of arrival time at intersection
Operation data	Time keeper data	DPI	Real-time	The time on-off Block that the bus is running on
AVL data	Vehicle Location	ATSP/DPI	Real-time	GPS location of the bus
	Speed	ATSP/DPI	Real-time	Average speed during sampling period of the bus

The data elements listed in Table 2-2 have different quality restrictions in terms of their accuracy, updating frequency, sampling frequency (real-time only), data latency (real-time only) and communication channel bandwidth. The different restrictions to the data quality have great impact on the architecture of how the integrated ATSP/DPI system can be implemented. Meanwhile, ATSP and DPI applications have certain requirements for the data quality, and the requirement may vary for ATSP and DPI. For example the requirement of the sampling frequency for the DPI application can be significantly lower than that of ATSP. Table 2-3 below is a cross comparison of the requirements of the ATSP and DPI applications and the restrictions of the real-time data elements.

Table 2-3 Comparison of Requirements of Real-Time Data Elements and their Restrictions

Real-time Data elements		System restrictions	DPI requirement	ATSP requirement
Signal phase and timing	Sampling rate	1 second to 5 seconds per sample	Not applicable	Up to 3 seconds
	Accuracy	Not applicable		
	Communication channel capacity	support full speed by using dedicated wireless module		
	Latency	Within 1 second	Not applicable	Within 3 seconds
Time keeper data	Sampling rate	Change per block / trip		
	Accuracy	Not applicable (there could be errors when operator fails to sign on the route, and thereby cause missing bus in the DPI prediction)		
	Communication channel capacity			
	Latency			
AVL data	Sampling rate	1/sec		
	Accuracy			15 meters
	Communication channel capacity	Low bandwidth of ACS (1 sample every 60-120 seconds)		
	Latency			

The existing AVL/ACS system uses a dedicated wireless radio, which has very limited communication bandwidth and therefore imposes a very strict data rate limitation to the AVL data. The sampling rate of GPS data could vary from 60-120 seconds. This dedicated radio system, although very slow, provides a very reliable wireless channel for operation.

The AVL/ACS system can also be built based upon COTS GPS devices and wireless communication network such as a commercial 3G network. The utilization of the commercial wireless communication network makes the (equipment) cost of the ACS system very much reduced. Due to the advanced technology of the 3G network, it could provide much higher bandwidth for AVL data communication and the sampling rate could be second level when needed. The sampling rate is less important for fleet management. While for predictive arrival time, minute level sampling rate would degrade the performance significantly and subsequently a second level sampling rate would be desirable.

Based on the different scenarios of either using existing AVL/ACS systems, or the potential for using commercial wireless service for AVL and TSP communication, we will present various architecture designs of ATSP/DPI processing in the next subsection.

2.3.4. ATSP/DPI Processing with Various Communication Technologies

Two processing architectures have been developed. The first is the system design based on the existing AVL/ACS technology. From Table 2-3, it is clear that the limited bandwidth of ACS communication could not support a fully centralized ATSP/DPI with desired operational performance. The other architecture is to consider using commercial wireless network for AVL data collection and TSP operations. The commercial network usually has higher bandwidth therefore allows higher data rate from field to center and thereby allows either centralized or distributed processing. For small transit agencies that do not have the dedicated ACS radio system deployed, the second scenario could be a desirable choice.

Two different processing architectures were therefore developed for the scenarios, as illustrated in Figure 2-10 and Figure 2-11. A distributed ATSP has to be adopted to integrate DPI with dynamic polling for better system performance. And a centralized processing scheme would require a high bandwidth communication channel to be available for the AVL data collection.

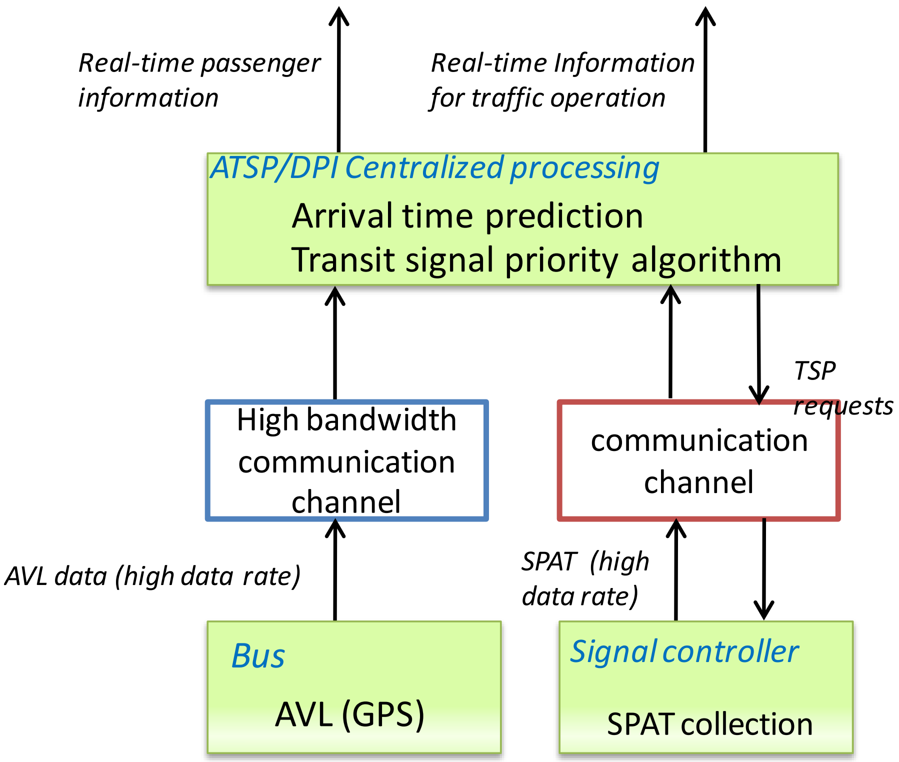

Real-time passenger information
Real-time Information for traffic operation
ATSP/DPI Centralized processing
Arrival time prediction
Transit signal priority algorithm
TSP requests
High bandwidth communication channel
communication channel
AVL data (high data rate)
SPAT (high data rate)
Bus
AVL (GPS)
Signal controller
SPAT collection

Figure 2-10 Centralized Processing for ATSP/DPI

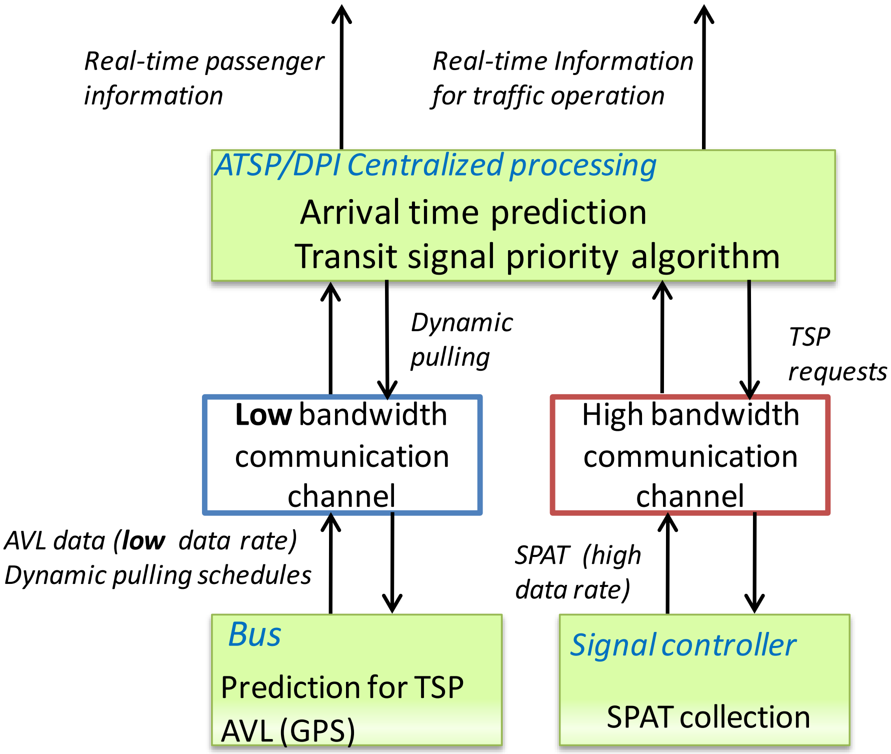

Real-time passenger information
Real-time Information for traffic operation
ATSP/DPI Centralized processing
Arrival time prediction
Transit signal priority algorithm
Dynamic pulling
TSP requests
Low bandwidth communication channel
High bandwidth communication channel
AVL data (low data rate)
Dynamic pulling schedules
SPAT (high data rate)
Bus
Prediction for TSP
AVL (GPS)
Signal controller
SPAT collection

Figure 2-11 Distributed TSP Prediction with Dynamic Polling

Table 2-4 shows the differences of the two different schemes.

Table 2-4 Comparison of Distributed versus Centralized TSP Processing for Different Communication Channel Capacities

	Low AVL communication bandwidth	High AVL communication bandwidth
Processing scheme	Distributed ATSP processing with dynamic polling	Centralized ATSP / DPI processing
Onboard processing	Onboard prediction for TSP and scheduling of dynamic polling	Minimized (send the AVL data out)
AVL data sampling	Low rate fixed sampling plus on-demand polling for intersection and bus stop check in / check out	High speed sampling
Complexity	**High**: involves dynamic polling and more onboard processing	**Low**

The eventual goal of the ATSP/DPI integration is to use the distributed ATSP prediction with dynamic polling as shown in Figure 2-11. In this project, we use centralized processing as shown in Figure 2-10 instead as an alternative approach due to limited access to the existing software of the ACS onboard computer.

2.3.5. Database Architecture

A. Overview of database architecture

The database and the software are binding together to form a service-oriented-architecture (SOA), which maximizes the sharing of module/service, interoperability and integration with existing applications and web services.

The database subsystem includes the database management system and the layered modules / design of the database for supporting the DPI system.

Open source database management system MySQL is selected (Wikepedia). This is relational database management software with a GPL license and supports multiuser access. MySQL has been widely used in many high-scale systems such as Craigslist (Craigslist).

B. Layered Database Design

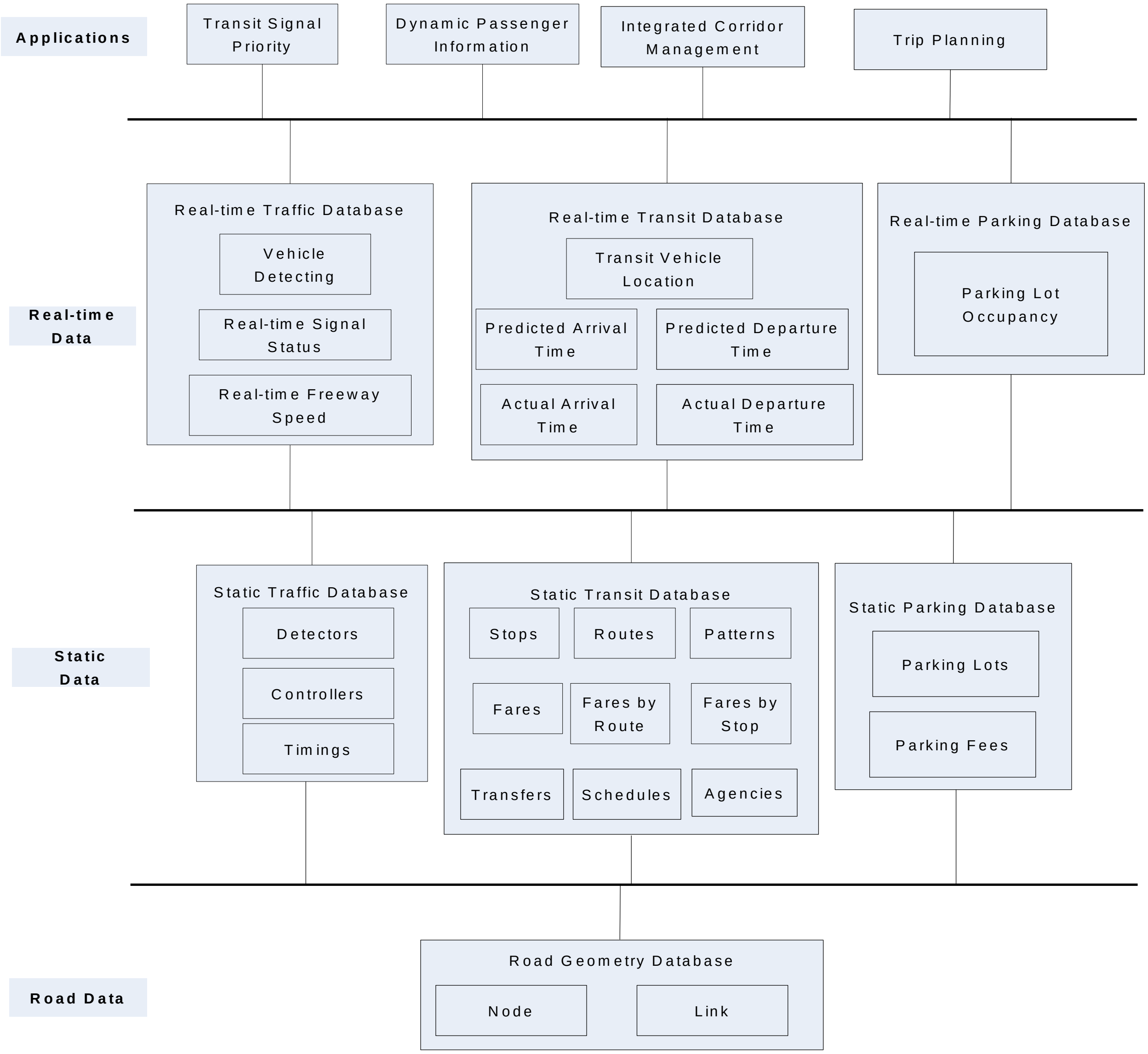


Figure 2-12 Layered Database Design

In order to improve flexibility and reduce the complexity, we decompose the whole data management system into several layers (c.f. Figure 2-12). The lowest layer is the underlying road network, including intersections and road segments. The information of each intersection, or a node in the network term, is stored in a database table. The table columns include a specified ID, intersection name (usually tow crossing road names), latitude and longitude. Each road segment, or a link in the network term, is stored in another table, containing the information of a specified link ID, starting and ending node IDs, road length, and numbers of lanes.

The second layer is for the traffic and transit data. The traffic controller information is associated with a node and is stored in a database table. The real-time and historical information of the signal status can be stored in another table, including the current date and time, phase information, force-off point, etc. The signal status table is associated with the node table by using the node ID. For the transit data, bus stop information is stored in a table of the transit database, with the specified stop ID, agency name, bus stop name, longitude and latitude and associated link ID. The bus route is then defined as a sequence of bus stop IDs in the route and stored in another table. The real-time GPS information of buses is stored in a table with the current time, position, speed, etc.

The applications, such as the transit signal priority and dynamic passenger information, can be implemented on the third layer. The applications may generate more information, such as the predicted arrival time to an intersection. Such information will be stored in a database table.

2.3.6. Central Processor

At the central processor, the aggregated data from AVL/ACS system, static transit data such as the published schedule and user information will be processed to generate real-time passenger information, planning results and be further personalized to fit into the user’s itinerary.

The modules of the DPI system processor are also listed below.

A. *Data Interfaces*

- Data interface to AVL/ACS system;
- Data interface to AVL and wireless communication system;

  The modules were developed for the two different kinds of AVL/ACS systems.

  The functionalities of the data interface modules are to interface with the external data source from AVL/ACS and AVL systems and to archive the data into database.

  Technologies used for the data interfaces can be specified as the following:

  - Communication APIs:

    The communication between the DPI and the AVL/ACS systems will be message protocols defined by AVL/ACS vendor;
  - Languages:

    C/C++;
  - Messaging

    AVL/ACS: The data is pushed from the transit agencies using sftp uploading to the central processor;

    For commercial 3G communications, data is pushed from the remote client using TCP/IP messages.

B. *Repository and processing modules– Transit Vehicle Arrival Time Prediction*

The repository and processing of transit data is directly related to the types of the transit service. There are two types of services for fixed-route transit, i.e., schedule-based and headway-based, that the transit vehicle arrival time prediction process shall be able to work with (Li, Zhou, Zhang, & Zhang, 2010). The prediction accuracy shall meet requirements. The requirements of the data accuracy vary for different applications. For pre-trip planning, the long-term prediction shall provide both prediction and variance information (variance usually comes from the historical data) to address the applicability and reliability of real time data into the optimization procedure employed in the trip planner. While for the en-route passenger information generation, the prediction data used are usually in the mid-to-short-term. The accuracy requirement is related to the resolution of the display of the information, which is whether to provide the information in tens-of-minutes, or minute or even sub-minute resolutions. The application of TSP, however, would impose the most critical requirement to the accuracy of short-term prediction so that the green window to be extended or number of seconds of early green to be requested can be minimized to have a better chance to acquire signal priority.

Inputs to arrival time prediction process include transit vehicles' positioning data, coming from either ACS or onboard AVL system, and the static road network, transit route network and schedule data stored in the database. Transit agencies manage fixed-route services by runs. Different runs could have different schedules and make stops at different transit stops/stations. The arrival time prediction process shall be able to correctly match transit vehicle's positioning data with transit runs. When ACS data are available, the information about the current run a transit vehicle is serving can be part of the ACS data. When position data are coming from onboard AVL system that does not provide the current run information, the arrival time prediction process shall be able to do an automatic match. AVL/GPS data have irregularities, measurement errors and GPS signal blockage among others. The arrival time prediction process shall be able to filter the AVL/GPS data for error correction and provide uninterrupted arrival time prediction during the blockage zones.

Outputs from the arrival time prediction process include the run (route, direction and run sequence number) a transit vehicle is serving (database Table "VehRunningRoute"), the predicted arrival time at all downstream stops/stations (database Table "PredVehArrTime"), the predicted arrival time at the next signalized intersection (database Table "PredVehSigTime"), the actual arrival time and departure time at passed stops/stations (database Table "ActualVehArrTime" and "ActualVehDepTime"), and the actual departure time at signalized intersections (database Table "ActualSigDepTime"). The first 3 tables, i.e., VehRunningRoute, PredVehArrTime and PredVehSigTime provide inputs to other processes, i.e., trip planning, en-route information generation and TSP control. The last 3 tables, i.e., "ActualVehArrTime", "ActualVehDepTime" and "ActualSigDepTime", store necessary information for system evaluation and improvement.

C. *Repository and processing modules – Passenger Information Generation*

The DPI system should be designed to meet the different variations of the information media, and to provide a flexible representation of the information for different needs, as can be seen in Figure 2-13. To meet this requirement, the generation of the DPI information should be designed in such a way that data processing and information generation are well modularized with primitive API interfaces for the generation of DPI information. Examples of the DPI information include the bus arrival time at a specific stop for given routes, location based information which is generated for a given GPS location, or more advanced information which is based on the GPS location and the user itinerary.

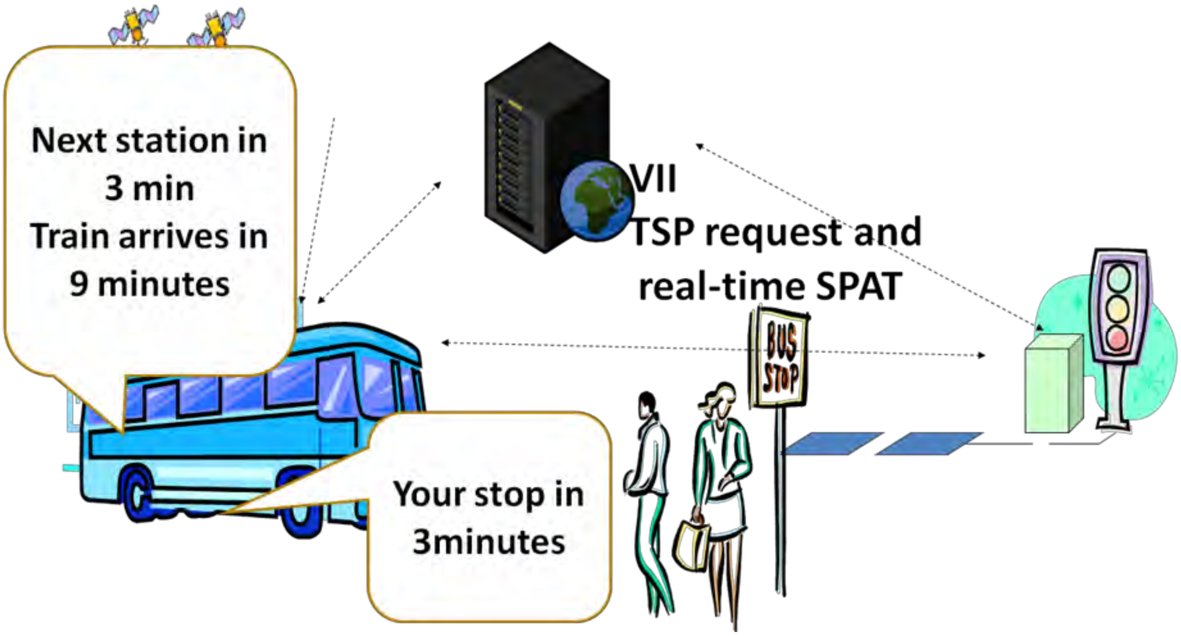


Figure 2-13 DPI Information

D. Repository and processing modules– Trip Planning

Figure 2-14 illustrates the architecture of the trip planning server. While the essence of most multi-modal trip planners is to seek good travel routes for given origin, destination and starting/arrival time, finding good routes is far more complicated than solving a simple shortest path problem. First, different users may have different preferences. For example, some users prefer trains to buses. In addition, it is difficult to model these preferences using some quantitative weights. Assume that there are two routes, where one route requires a slightly longer walking distance, while the other one requires a slightly longer time staying in the bus. Different users may have different opinions on which route is better. Therefore, multi-modal planners generally provide several good routes to users so that the users can choose the best one from these routes by themselves.

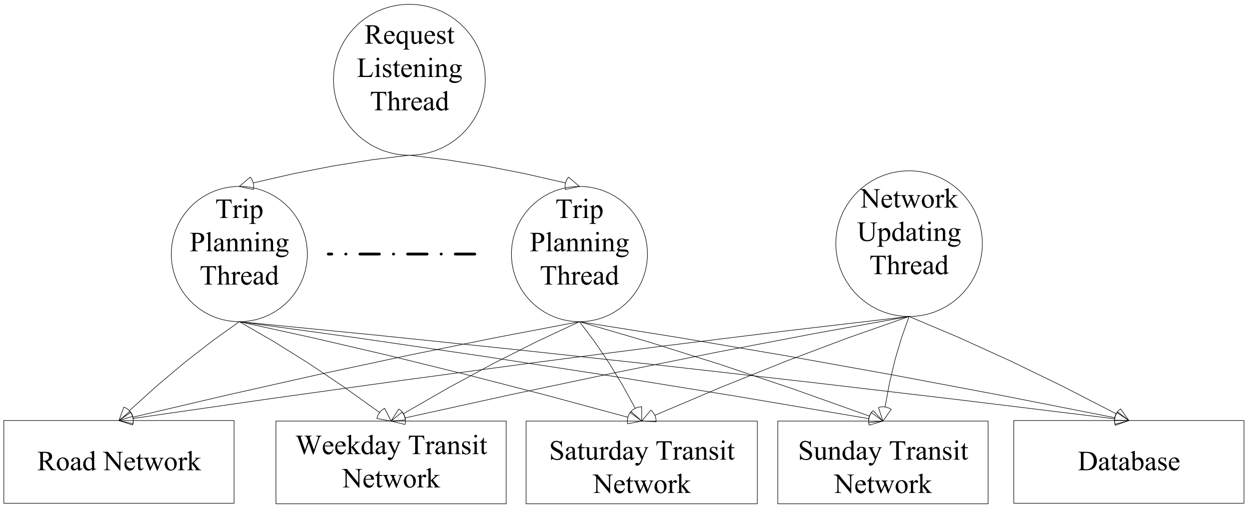


Figure 2-14 The Architecture of the Trip Planning Server

When users choose to drive from a given origin to a destination, the quickest route is returned and a Dijkstra algorithm is applied (D' Angelo, Al-Deek, & Wang, 1999). If users choose the mode of transit only or park-and-ride, we first select the transit stops or parking lots that are near the origin. Then, the transit stops nearby the destination are determined. In summary, we solve three kinds of shortest path problems: (i) from the origin to the nearby transit stops or parking lots; (ii) from the destination to the nearby transit stops; and (iii) from the transit stops or parking lots that are close to the origin to the transit stops that are close to the destination. These routes will be combined together to yield an overall route.

E. Application Layer

The Applications are built upon the layered architecture design that all different data sources are processed to a unified format and stored in the database. Applications will be able to access the data from different sources in real-time. The major benefits of the system are shown in the application layer: Applications do not need to know the details about the data processing and data format and data source. Either the data comes from AVL equipment (GPRMC standard GPS data), AVL/ACS system (vendor specific data) or a third party real-time data feed (e.g., XML data feed); in either case, they are all converted into the same intermediate data in the database.

- Web application

  With the data processing layer that generates the intermediate data in real-time, various web-based applications are enabled, to include the web based trip planner (using real-time data), web based real-time transit information update and even web based transit vehicle tracker.

  The design and development of the web based application will need some third party library support. Most commonly needed is the Google Maps service which is the top choice for map visualization. Another example is the Google Geocoding service (which converts the road addresses input by the user to GPS latitude and longitude, useful for trip planner).

The software tools used to develop the web application are:

(1) Web engine: Sun application server and Apache http server;

(2) To allow dynamic information on the web page, the popular AJAX (asynchronous javascript and XML) technology is adopted as the primary web framework, so that the application can be more user-responsive and rich in dynamic contents.

(3) Supported web browsers: Firefox, Safari and Chrome.

- Mobile Applications
  Mobile applications will be the applications that utilize the GPS capability of the portable device and to get some location-based information in real-time. To support mobile applications, specific software needs to be developed for certain mobile platforms, e.g., windows mobile platform, iPhone, etc.

3. Development of ATSP/DPI System

The development of the integrated ATSP/DPI system will be addressed in this and the next section. Below is a table that outlines the different topics that will be addressed in this report about the system design.

Table 3-1 Outline of the ATSP/DPI Design Report

Topic	Layer	Section
Database design	Database layer	Section 3.1
Central Processor- Arrival Time Prediction	Processing Layer	Section 3.2
Central Processor: Signal Priority and Dynamic polling algorithms	Processing Layer	Section 4.1
ATSP application modules	Application and processing layer	Section 4.1
DPI application modules	Application and processing layer	Section 3.3

Among all the topics, the signal priority and dynamic polling algorithms are the critical portion for the integration of the ATSP/DPI and are highlighted in a separate section - Section 4.1.

3.1. Design of database for Integrated ATSP/DPI

Data management is critical for supporting the open architecture ITS applications. Archiving and managing a large set of data efficiently is very challenging since there are many different types of data, including traffic data, transit data, and underlying road network data. Each category can include many different types of data. For example, the traffic data may include signal timing and detector data. In addition, some data is static, while other data is dynamic in nature. Data from these different sources are strongly related to each other. This database architecture reflects this correlation. For example, in order to implement the transit signal priority, both the current bus location and signal status of the approaching intersection are needed. Moreover, the amount of traffic and transit data is substantial. For instance, in the case of the GPS location of each bus, data are sent to the transit center once every second. The database may contain billions of records when the data for hundreds of buses for a year is stored. Hence, our goal is to design flexible, scalable and efficient ITS databases.

Another design consideration of the database is the adaptation of standard data formats and structures for better compatibility. As more and more transit agencies have started using the Google Transit Feed Specification (GTFS, (Google, 2010)) for their exchange of static information (Google, 2011), we have designed our database tables for the static transit data based on the GTFS data formats. This way we expect the system to be more open and compatible for future expansions.

3.1.1. **Overview of Database Structure**

In order to improve flexibility and reduce complexity, we decomposed the data management component into several layers, as shown in Figure 3-1.

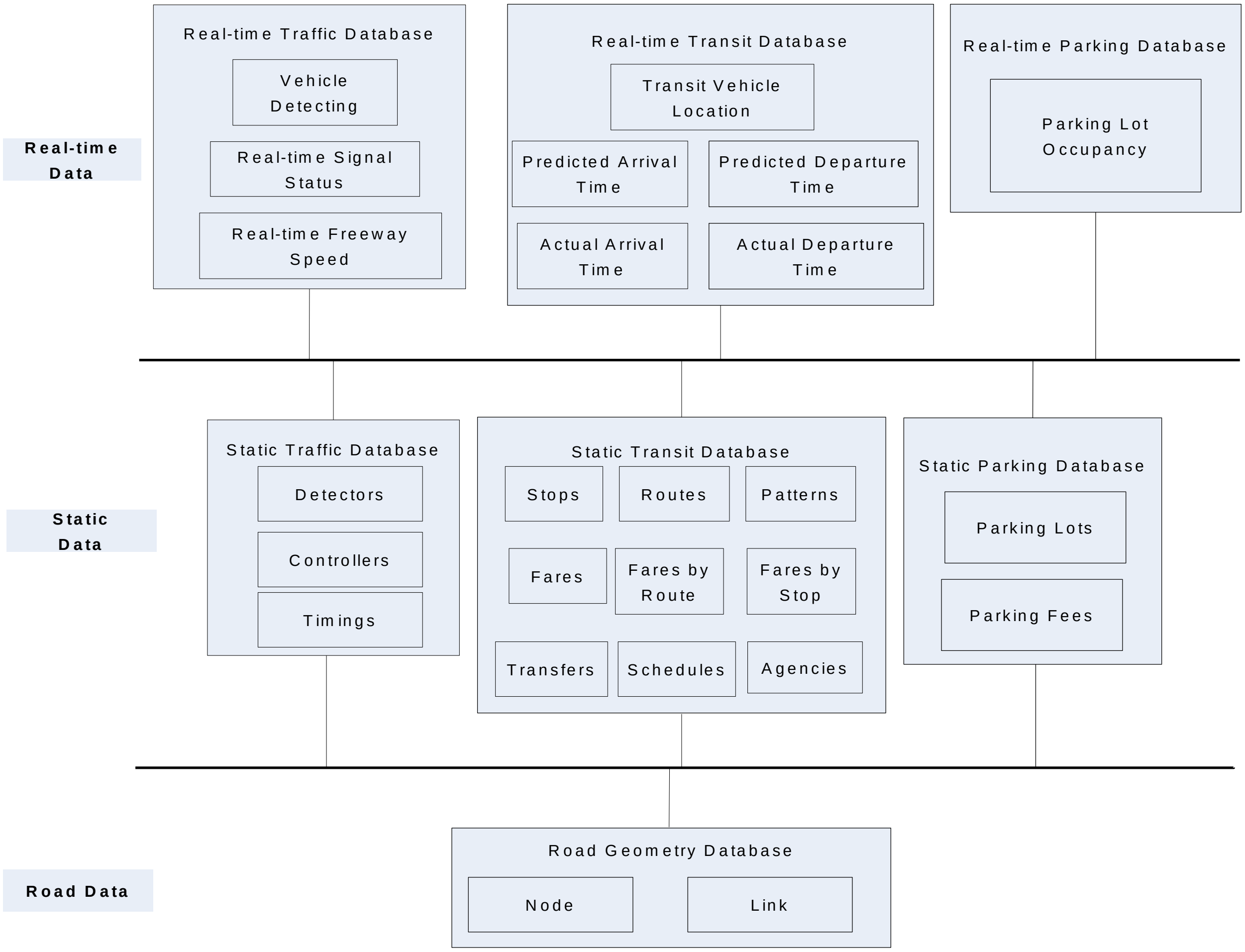


Figure 3-1 Layered Database Architecture

The lowest layer is the underlying road network, including intersections and road segments. An intersection is modeled as a node, and a road segment is modeled as a link, as seen in Figure 3-2.

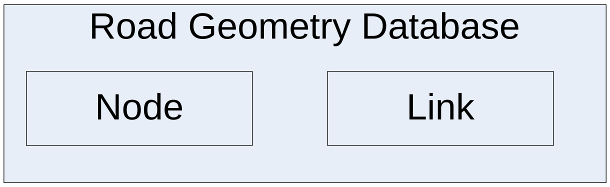


Figure 3-2 Road Geometry Database

The second layer is designed for storing the static traffic, transit, and parking data. The static traffic database includes the data of detectors, traffic controllers, timing tables, etc., as shown in Figure 3-3. The static transit database includes the data for transit stops, transit routes, route patterns, schedules, fares, transfers between transit routes, agencies, etc., as shown in Figure 3-4. The design of the static transit database is based on the GTFS specification (Google, 2010). The static parking database includes the data for parking lots, parking fees, etc., as shown in Figure 3-5.

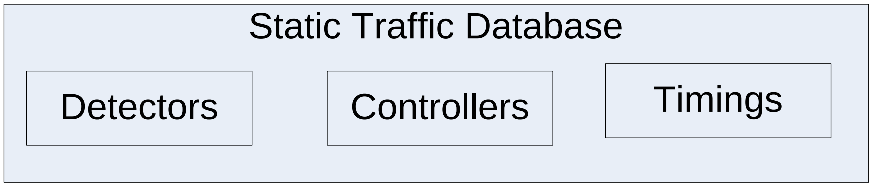


Figure 3-3 Static Traffic Database

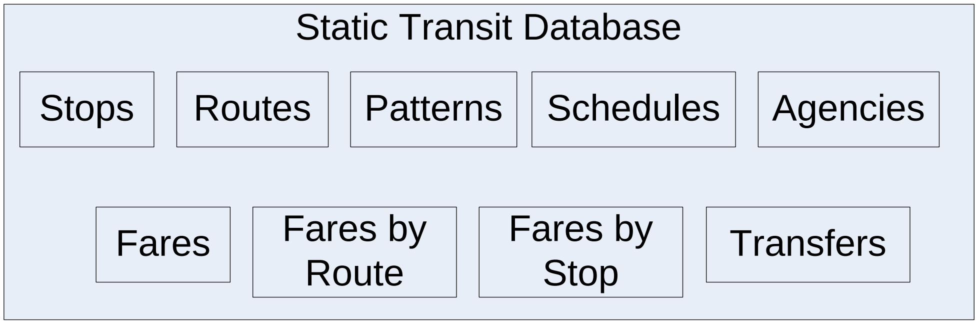


Figure 3-4 Static Transit Database

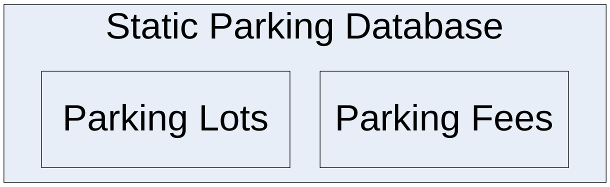


Figure 3-5 Static Parking Database

The third layer is designed for storing the real-time traffic and transit data. The real-time traffic database includes one for vehicle detecting information, real-time traffic signal status, real-time freeway speed, etc., as shown in Figure 3-6. The real-time transit database includes the data for real-time transit vehicle location, predicted arrival time, predicted departure time, actual arrival time, actual departure time, etc., as shown in Figure 3-7. The real-time parking database includes the data for real-time occupancy of parking lots, as shown in Figure 3-8.

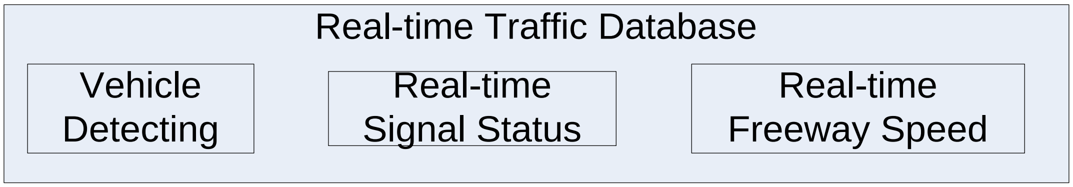


Figure 3-6 Real-Time Traffic Database

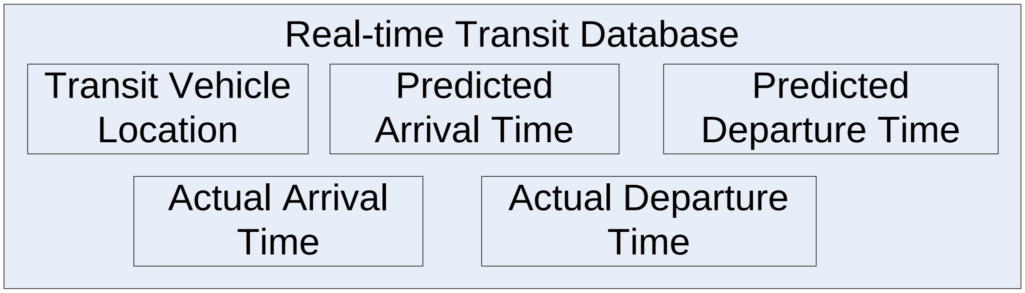


Figure 3-7 Real-Time Transit Database

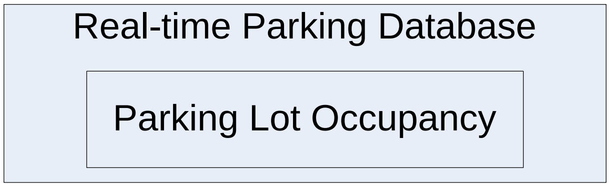


Figure 3-8 Real-Time Parking Database

3.1.2. Database Structure for Static Data

A large number of data are static in nature, including the road geometry data, static traffic data, static transit data, and static parking data. Such data is often related to road geometry, transit routes, transit stops, detector locations, signal controllers, etc. It is not necessary to present all the tables with complete records since a database table may include more than one hundred structures. In this section, we describe major tables with important structures.

The information from each node (intersection) includes a specified ID, intersection name (usually two crossing road names), latitude, longitude, and a combination of latitude and longitude (see Table 3-2).

Table 3-2 Major Structures of Table Node

Name	Type	Description
ID	integer	ID of the intersection
Longitude	double	longitude of the intersection
Latitude	double	latitude of the intersection
Name	varchar	name of the intersection
LatLong	point	combination of longitude and latitude, which is used for speeding up queries

Each segment table (road) contains the information of a specified link ID, starting and ending node IDs, road length, road category, road name, etc. (see Table 3-3).

Table 3-3 Major Structures of Table Segment

Name	Type	Description
LINK_ID	integer	ID of the road segment
StartNode	integer	ID of starting intersection
EndNode	integer	ID of ending intersection
ST_NAME	varchar	name of the segment
FUNC_CLASS	integer	an indication of road category: local, arterial, freeway, etc

Major structures of table Agency contain agency id, name, URL, phone number, and time zone of a transit agency (see Table 3-4).

Table 3-4 Major Structures of Table Agency

Name	Type	Description
agency_id	Integer	ID of a transit agency
agency_name	Varchar	name of the transit agency
agency_url	Varchar	URL of the transit agency
agency_phone	Varchar	phone number of the transit agency
agency_timezone	Integer	time zone of the transit agency

Major structures of table Routes contain the agency ID, short route name, route direction, transit type, long route name, route description, starting stops of the route, and ending stops of the route (see Table 3-5).

Table 3-5 Major Structures of Table Routes

Name	Type	Description
agency_id	Integer	ID of a transit agency
route_short_name	Varchar	Short name of the route
route_dir	Integer	Route direction
transit_type	Integer	Transit type: bus, train, metro, ferry, etc
route_long_name	Varchar	Long name of the route
route_desc	Varchar	Description of the route
route_start	Varchar	Starting stops of the route
route_end	Varchar	Ending stops of the route

Major structures of table Stops contain the agency ID, stop ID, transit type, stop name, stop description, latitude and longitude of the stop, etc. (see Table 3-6).

Table 3-6 Major Structures of Table Stops

Name	Type	Description
agency_id	integer	ID of a transit agency
stop_id	integer	ID of the transit stop
transit_type	integer	Transit type: bus, train, metro, ferry, etc
stop_name	varchar	Name of the stop
stop_desc	varchar	Description of the stop
stop_lat	double	latitude of the stop
stop_lon	double	longitude of the stop
lat_lon	point	combination of longitude and latitude, which is used for speeding up queries

Major structures of table Schedules contain the agency ID, short route name, starting date of schedules, day of week, route direction, run ID, stop ID, sequence of stop in the trip, scheduled time, availability days, time point flag, etc. (see Table 3-7).

Table 3-7 Major Structures of Table Schedules

Name	Type	Description
agency_id	integer	ID of a transit agency
route_short_name	varchar	Short name of the route
start_date	varchar	Starting date of the schedule
day	varchar	Weekday, Saturday, or Sunday
route_dir	integer	Route direction
run	integer	Run per a directional route, starting from 1 as the first trip
stop_id	integer	Stop ID
Seq	integer	Sequence of the stop in the trip
schedule	varchar	Published or estimated schedules
avail_days	varchar	For Weekday trips only, since some trips are available only for specific days, say Tuesday only
tp_flag	integer	Time point or not

Each route may have a few different patterns at different times of the day. Table RunPattern associates a trip with a route pattern. Major structures of table Schedules contain the agency ID, short route name, starting date of schedules, day of week, route direction, run ID, and pattern ID, etc. (see Table 3-8).

Table 3-8 Major Structures of Table Segment

Name	Type	Description
agency_id	integer	ID of a transit agency
route_short_name	varchar	Short name of the route
start_date	varchar	Starting date of the schedule
day	varchar	Weekday, Saturday, or Sunday
route_dir	integer	Route direction
run	integer	Run per a directional route, starting from 1 as the first trip
pattern_id	varchar	Route pattern ID

Table Route_stop_seq is to define the route pattern. Major structures of table Route_stop_seq contain the agency ID, short route name, route direction, run ID, pattern ID, stop ID, and sequence of stop, and time point flag (see Table 3-9).

Table 3-9 Major Structures of Table Route_stop_seq

Name	Type	Description
agency_id	integer	ID of a transit agency
route_short_name	varchar	Short name of the route
route_dir	integer	Route direction
pattern_id	varchar	Route pattern ID
stop_id	integer	Stop ID
seq	integer	Sequence of the stop in the trip
tp_flag	integer	Time point or not

There are different definitions on transit fares: some agencies define fares on routes (most bus routes), while others may define fares based on origin and destination (Caltrain and BART). Table Fare_on_route includes agency ID, short route name, prices, currency type, and the number of allowable transfers (see Table 3-10). Table Fare_on_stops includes agency ID, the ID of original stop, the ID of destination stop, prices, currency type, and the number of allowable transfers (see Table 3-11).

Table 3-10 Major Structures of Table Route_stop_seq

Name	Type	Description
agency_id	integer	ID of a transit agency
route_short_name	varchar	Short name of the route
price	integer	Prices of a trip in the unit of cents
currency_type	varchar	Currency type (USD by default)
transfers	integer	The number of allowable transfers

Table 3-11 Major Structures of Fare_on_stops

Name	Type	Description
agency_id	integer	ID of a transit agency
origin_id	integer	ID of the origin stop
destination_id	integer	ID of the destination stop
price	integer	Prices of a trip in the unit of cents
currency_type	varchar	Currency type (USD by default)
transfers	integer	The number of allowable transfers

Table Controllers include the ID of the intersection, ID of the traffic controller, and controller type (see Table 3-12).

Table 3-12 Major Structures of Table Controllers

Name	Type	Description
node_id	Integer	Intersection ID
controller_id	Integer	ID of the traffic controller
controller_type	Varchar	Controller type: fixed time control, actuated control, real-time control, etc

Table Parking Lot is for definition of a parking lot, which includes the ID of parking lot, ID of train station, the capacity of the parking lot, the number of spaces for handicap, parking lot type, ID of the agency, the latitude and longitude of the parking lot, and parking lot description (see Table 3-13). Table ParkingFee includes the ID of the parking lot, ID of train station, ID of the agency, and parking fee (see Table 3-14).

Table 3-13 Major Structures of Table ParkingLot

Name	Type	Description
u_id	integer	Parking lot ID
u_station_id	integer	Train station ID
u_inventory	integer	Capacity of the parking lots
u_handicapped	integer	The number of spaces for handicap
u_type	integer	Parking lot type
u_agency_id	integer	ID of agency
latitude	double	Latitude of the parking lot
longitude	double	Longitude of the parking lot
s_desc	varchar	Parking lot description

Table 3-14 Major Structures of Table ParkingFee

Name	Type	Description
u_agency_id	integer	ID of agency
u_id	integer	Parking lot ID
u_station_id	integer	Train station ID

fee	integer	Parking fee

3.1.3. Database Structure for Real-Time Data

A large number of data are constantly being updated, for example the bus location data, traffic density of the freeway, and available spaces of a parking lot. The real-time information is stored in various tables, including the real-time traffic data, real-time transit data, and real-time parking data.

Real-time travel time information is stored in Table Real_time_traffic, which is periodically queried from traffic.com. Major structures include the road link ID, direction, average speed on the link, date and time when the average speed was recorded (see Table 3-15).

Table 3-15 Major Structures of Table Real_time_traffic

Name	Type	Description
LinkID	integer	The ID of the road link
Dir	integer	Direction of the link
AvgSpeed	double	Average speed on the link
RecordedDate	Date	Date when the average speed was recorded
RecordedTime	Time	Time when the average speed was recorded

Real-time signal status is recorded in table signal_status, which includes the ID of the agency, the ID of controller, date, time, and milliseconds when the real-time was recorded and received by the server, the current phase and interval of rings A and B, pedestrian calls, the local and master clock, and the force off points of approaches from 1 to 8 (see Table 3-16).

Table 3-16 Major Structures of Table signal_status

Name	Type	Description
agency_id	integer	ID of the agency
controller_id	integer	ID of the traffic controller
RecordedDate	Date	Date when the real-time data was recorded
RecordedTime	Time	Time when the real-time data was recorded
RecordedMs	integer	Milliseconds when the real-time data was recorded
ReceivedDate	Date	Date when the real-time data was received
ReceivedTime	Time	Time when the real-time data was received
ReceivedMs	Integer	Milliseconds when the real-time data was received
ringA_phase	Integer	Current phase of ring A
ringB_phase	Integer	Current phase of ring B
ringA_interval	Integer	Current interval of ring A
ringB_interval	Integer	Current interval of ring B
pedcalls	Integer	The number of pedestrian calls
local_clock	Integer	Local clock
master_clock	Integer	Master clock

force_off_1, …, 8	Integer	Force off for approaches 1 to 8

Real-time raw transit data is stored in Table gps_fixes, which includes the IDs of the transit vehicle and agency, real-time speed, latitude, longitude, and date, time, and milliseconds when the real-time information was recorded (see Table 3-17).

Table 3-17 Major Structures of Table gps_fixes

Name	Type	Description
VehID	Integer	ID of the transit vehicle
RecordedDate	Date	Date when the real-time location was recorded
RecordedTime	Time	Time when the real-time location was recorded
RecordedMs	Integer	Milliseconds when the real-time location was recorded
Speed	Double	Real-time vehicle speed
latitude	Double	Real-time latitude of the transit vehicle
longitude	Double	Real-time longitude of the transit vehicle
agency_id	Integer	ID of the transit agency

The raw real-time data is then processed, and a prediction algorithm is applied. The predicted arrival time to a transit stop is then generated. The information is stored in Table PredVehArrTime, which includes agency ID, route ID, vehicle ID, direction, stop ID, and predicted date, time and milliseconds (see Table 3-18).

Table 3-18 Major Structures of Table PredVehArrTime

Name	Type	Description
RecordedDate	Date	Date when the real-time location was recorded
RecordedTime	Time	Time when the real-time location was recorded
RecordedMs	Integer	Milliseconds when the real-time location was recorded
AgencyID	Integer	ID of the agency
Route	Varchar	Transit route
VehID	Integer	Vehicle ID
Direction	Integer	Direction of the route
StopID	Integer	Stop ID
PredictedDate	Date	Predicted date when the vehicle will arrive
PredictedTime	Time	Predicted time when the vehicle will arrive
PredictedMs	Integer	Predicted milliseconds when the vehicle will arrive

The actual transit arrival time is stored in table ActualVehArrTime with the similar definitions of the structures (see Table 3-19).

Table 3-19 Major Structures of Table ActualVehArrTime

Name	Type	Description
RecordedDate	Date	Date when the real-time location was recorded
RecordedTime	Time	Time when the real-time location was recorded
RecordedMs	Integer	Milliseconds when the real-time location was recorded
AgencyID	Integer	ID of the agency
Route	Varchar	Transit route
VehID	Integer	Vehicle ID
Direction	Integer	Direction of the route
StopID	Integer	Stop ID
ActualArrDate	Date	Actual date when the vehicle arrived
ActualArrTime	Time	Actual time when the vehicle arrived
ActualArrMs	Integer	Actual milliseconds when the vehicle arrived

The raw parking data includes the ID of parking lot, ID of train station, and currently available parking spaces (see Table 3-20).

Table 3-20 Major Structures of Table Real_time_parking

Name	Type	Description
u_id	integer	Parking lot ID
u_station_id	integer	Train station ID
u_available_spaces	integer	Available spaces of the parking lots

3.2. Bus Arrival Time Prediction

3.2.1. Projection of GPS Points onto the Bus Route

The bus route network is originally presented by a sequence of GPS points (latitude and longitude). A node-link representation is used to describe the bus route in 2-dimentional Euclidean surface space. Each GPS point is denoted as a node and the road segment between two adjacent nodes is denoted as a link. The first node is defined as the origin. Each node has 5 attributes $(x_i, y_i, s_i, l_i, h_i)$, where x_i and y_i are the coordinates of the i^{th} node (East and North), s_i is the distance into the route, l_i is the length of the link, and h_i is the heading direction from node i to node $i+1$, with respect to North.

The GPS point with latitude and longitude could either be a bus stop, a signalized intersection, or a real-time GPS trace of bus location. Heading measurement is also available for real-time GPS traces. Projected point on bus route is represented by 4 parameters (j, u, v, θ), where j is the link the point is projected onto, $u \in [0,1]$ is the projected distance into the link normalized by the link length, v is the lateral distance from the projected link, and θ is the angle of the lateral deviation as illustrated in Figure 3-9 with u given by

$$u = \frac{(\vec{r} - \vec{r}_i^{\,u})\cdot(\vec{r}_i^{\,d} - \vec{r}_i^{\,u})}{\left\|\vec{r}_i^{\,d} - \vec{r}_i^{\,u}\right\|^2}$$

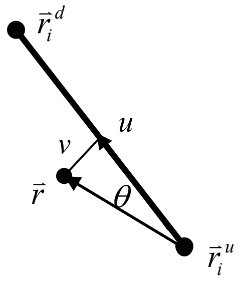


Figure 3-9 Projection of GPS Point onto Bus Route

The projection is formulated as finding the link j that satisfies the constraints:

$$u \in [0, 1]$$
$$\|v\| < v_{max}$$
$$|\theta - h_j| < \theta_{max}$$

where v_{max} and θ_{max} are predetermined thresholds.

Once the GPS point is projected, its distance into the route is determined by

$$s = s_j + u \cdot l_j$$

The distances to downstream bus-stops and intersections can then be easily calculated from the node-link representation.

3.2.2. Prediction of Bus Arrival Times at Intersections

A prediction algorithm for ATSP operations has been developed under a previous ATSP project. The algorithm combines the real-time bus movement data and historical data to predict bus arrival time at the intersection. Detailed deception of the prediction model can be found in a PATH research report (PATH Report, 2010). Below is a brief summary of the model, which is used as the foundation for the prediction model in predicting bus arrival time at a bus-stop.

Bus movement between two stops in traffic is modeled in terms of linear translational kinematics.

$$d = v \cdot t + b + n$$

where d and t are the distance and time, respectively, traveled since the last stop in traffic, v and b are model parameters, and n is model noise.

A second-order Kalman filter is applied to track bus movement and to remove GPS data noise. A linear regression method is then applied to adaptively estimate the model parameters in real-time.

Real-time estimate of *v* only depicts bus prevailing average travel speed since the last stop in traffic. In order to predict the arrival time at the downstream intersection, knowledge about possible travel speed in downstream should be considered in the prediction. This knowledge at a certain level is captured by historical data. Observation pairs of (*D*, *T*), where *D* and *T* are distance and travel time between two stops in traffic, retrieved from historical bus trips can be used to construct the likelihood function of $p(D|v_H)$. The prediction of travel speed to the downstream intersections is then formulated as the maximum a posteriori probability estimate

$$\hat{v} = \operatorname{argmax} p(v_H|v)$$

3.2.3. Prediction of Bus Arrival Times at Bus-Stops

The prediction of bus arrival time at bus-stop shares the same model framework with that for intersection arrival time prediction while the historical model is extended to include the consideration of bus operation characteristics, i.e., the schedule recovery phenomena, as described below.

Let S_u and S_d denote the scheduled departure time at an upstream and a downstream time-point, respectively, d_u and d_d denote the distance into the route of the two time-points, and τ_u denotes the schedule deviation at the downstream time-point, the expected schedule deviation, τ_d, at the downstream time-point is modeled as

$$\tau_d = \alpha \cdot \tau_u + \beta \cdot (S_d - S_u) + n$$

In other words, schedule deviation at the downstream time-point is contributed by two parts, the experienced schedule deviation at the upstream time-point and the scheduled travel time between the two. The equation above is equivalent to

$$\frac{1}{v} = (\alpha - 1) \cdot \frac{\tau_u}{d_d - d_u} + (\beta + 1) \cdot \frac{S_d - S_u}{d_d - d_u} + n$$

Model parameters α and β can be estimated from historical bus trips.

3.2.4. Performance Requirements for Predicted Arrivals

The ATSP/DPI system predicted real-time bus arrival information shall be accurate and the accuracy shall remain consistent during changing conditions such as morning and afternoon peak operation, transition to off-peak operation, etc.

The accuracy requirement is considerably different for ATSP purposes than for DPI purposes. The accuracy shall be within +/- 5 seconds at the time a Green Extension request was made and shall be within +/- 10 seconds at the time an Early Green request was made. The accuracy for bus-stop arrival time prediction shall be within +/- 4 minutes for most of bus trip time.

3.3. DPI Application Modules

In this section, the design and implementation of the DPI system regarding the generation and delivering of the DPI information is discussed. The DPI application software on the integrated ATSP/DPI center is the major focus, with one critical part of this central processor, the arrival time prediction, already addressed in section 3.2. Figure 3-10 shows the components of the software at the system center.

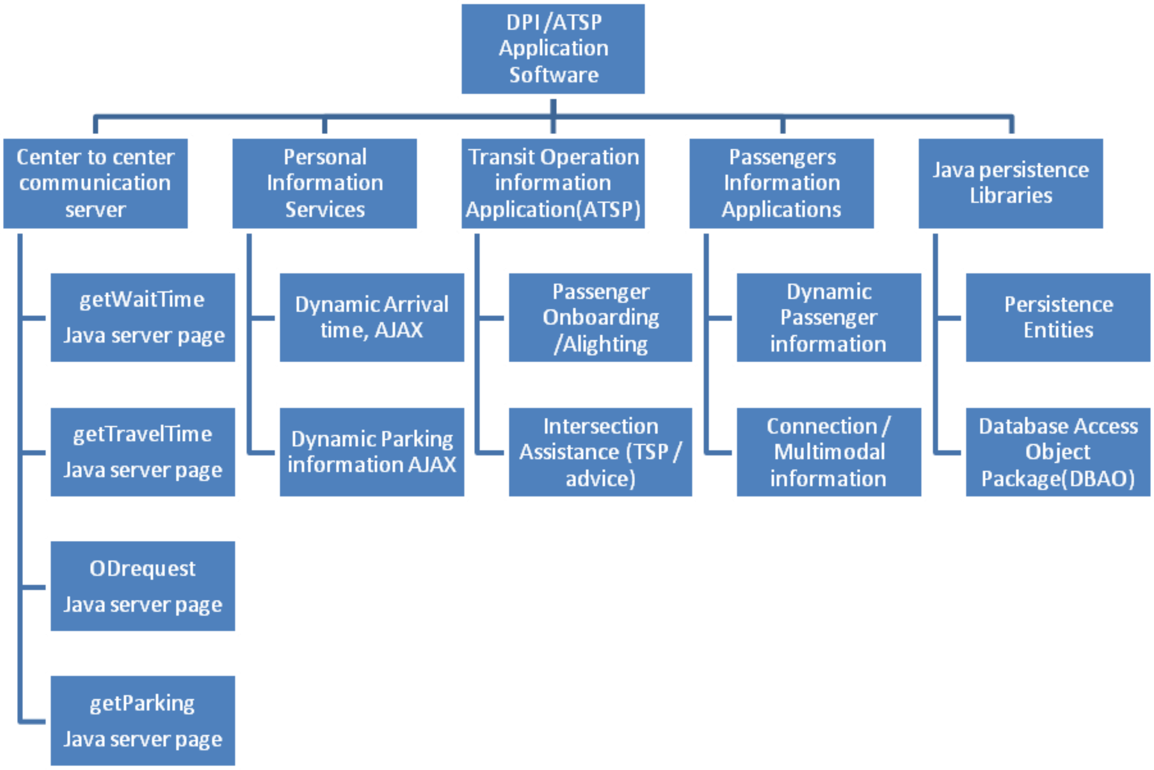

DPI /ATSP Application Software
Center to center communication server
Personal Information Services
Transit Operation information Application(ATSP)
Passengers Information Applications
Java persistence Libraries
getWaitTime Java server page
getTravelTime Java server page
ODrequest Java server page
getParking Java server page
Dynamic Arrival time, AJAX
Dynamic Parking information AJAX
Passenger Onboarding /Alighting
Intersection Assistance (TSP / advice)
Dynamic Passenger information
Connection / Multimodal information
Persistence Entities
Database Access Object Package(DBAO)

Figure 3-10 ATSP/DPI application software at central processor

This section covers the dynamic passenger information generation module in the system block diagram Figure 2-9. Figure 3-11 shows the processing and application modules for the integrated ATSP/DPI system. Remainders of the modules are addressed in different portions of this report, with the arrival time prediction module being addressed in section 3.2, signal priority and dynamic polling algorithms and ATSP application modules in section 4.1.

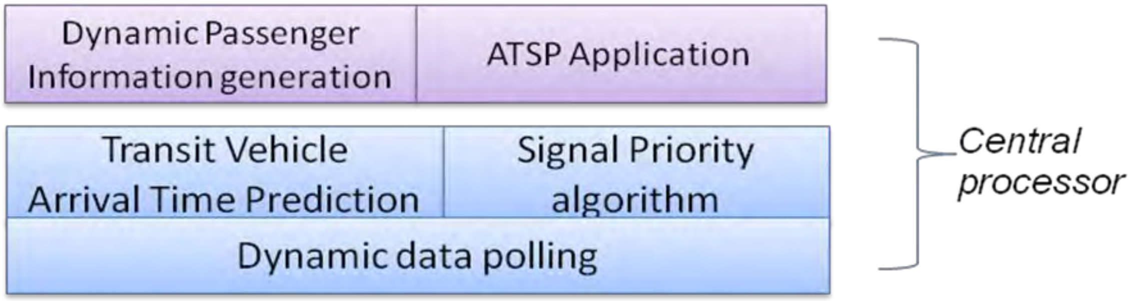

Dynamic Passenger Information generation
ATSP Application
Transit Vehicle Arrival Time Prediction
Signal Priority algorithm
Dynamic data polling
Central processor

Figure 3-11 Processing Modules in the Central Processor of the Integrated ATSP/DPI System

3.3.1. Overview of Generation of DPI Information

The challenges associated with the generation of DPI information are several-fold. The data from different sources may have different formats, accuracy and dictionaries for encoding of major data elements, such as stop, route, etc. For example, the AVL data from the AC Transit / SamTrans OrCAD system have proprietary plain text format, while from the Nextbus service the center to center AVL data samples are in XML data format (Nextbus). Presenting the DPI information to the travelers pre-trip, at bus stop or en route onboard a bus with uniform application interface is another challenge.

The ATSP/DPI should aggregate the data of different formats and data quality, process them and then disseminate the information using an open platform. This is partially discussed in Section 2.3.3. Third party services that greatly help to disseminate and present the information should also be integrated. One example is the map visualization services that can help to associate the location data to an actual map. Service providers include Google Maps, Microsoft Bing Maps, etc.

The DPI information generation procedure at the ATSP/DPI system is described in the following parts:

- Data aggregation, for both the static data and the real-time data;
- Data processing for DPI information (excluding the arrival time prediction which has been covered in previous section); and
- DPI information dissemination using an open platform.

Then we will summarize the information processes for different scenarios based on the system application procedures.

3.3.2. Data Aggregation for ATSP/DPI Data Sources

There are several different data sources that need to be aggregated. We will use three example agencies here, AC Transit, SamTrans and VTA, to show the data sources and their formats.

Table 3-21 Formats of Different Data Sources

Agency	Data	Format	Description
SamTrans	Static schedule	511 XML format	Transit route description, time point file, service days and schedule at time-points
	Real-time AVL data	Plain text	Vehicle ID, GPS location, speed, block #, running route and direction, time point offset.
VTA BRT	Static schedule	GTFS (Google transit feed specification)[3]	Including every aspect of the transit static data.
	Real-time AVL data	NMEA GPRMC[4] (second by second GPS only)	Vehicle id, GPS location, speed and serving route.
AC Transit	Static Schedule	511 XML	Same as Samtrans
	Real-time AVL data	Plain text	Vehicle ID, GPS location, speed and block #.

To aggregate the data sources with different formats and contents as seen in Table 3-21, the DPI system needs to have data pre-processing model that either works offline (for static data which updates every a few months to more than a year, depending on the agency) or online (for real-time data).

A. *Static transit data aggregation*

For the static data, the major task for the data aggregation is to convert the data from other formats to a suitable format that can be imported into the database structure. Based on the generality of the GTFS data sources, we have designed our database structure to work with GTFS data directly and therefore other sources of data which do not comply with GTFS will need to be converted first before it can be imported. And due to the complicated structure of the transit data, various programs have been developed to convert different formats and import the data into database. An importer is a program that has a database connector (DBC)

[3] Google transit feed specification has been widely adopted as format for exchange of static transit schedule information http://code.google.com/transit/spec/transit_feed_specification.html

[4] NEMA National Marine Electronics Association (NMEA) has developed a specification that defines the interface between various pieces of marine electronic equipment. GPRMC is a standard sentence format for GPS data exchange. http://www.gpsinformation.org/dale/nmea.htm

embedded that can insert into the database structured data that fully conforms to the schema in the database tables as described in section 3.1.2. This process is illustrated in Figure 3-12.

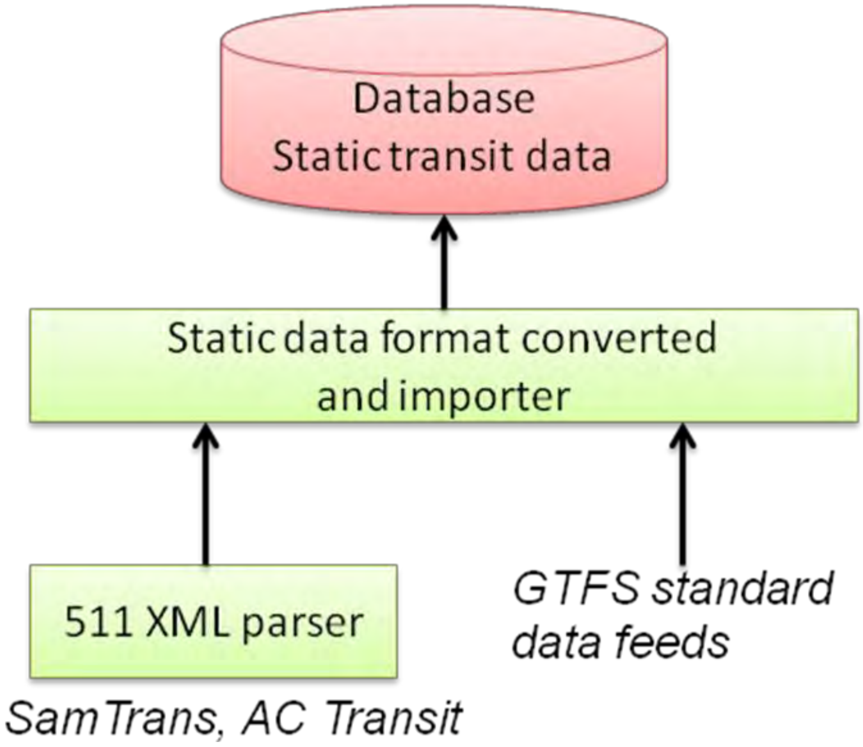


Figure 3-12 Static transit data aggregation

B. *Aggregate real-time transit data*

The pre-processing needed for aggregating real-time data is more complicated than the static data, when important information from the AVL data is missing for arrival time prediction purpose. This happens to all three different types of AVL data sources that we use for the DPI system as mentioned in Table 3-21 when part of the real-time operation information is missing from the AVL data which associates a bus to a running route, direction and service trip.

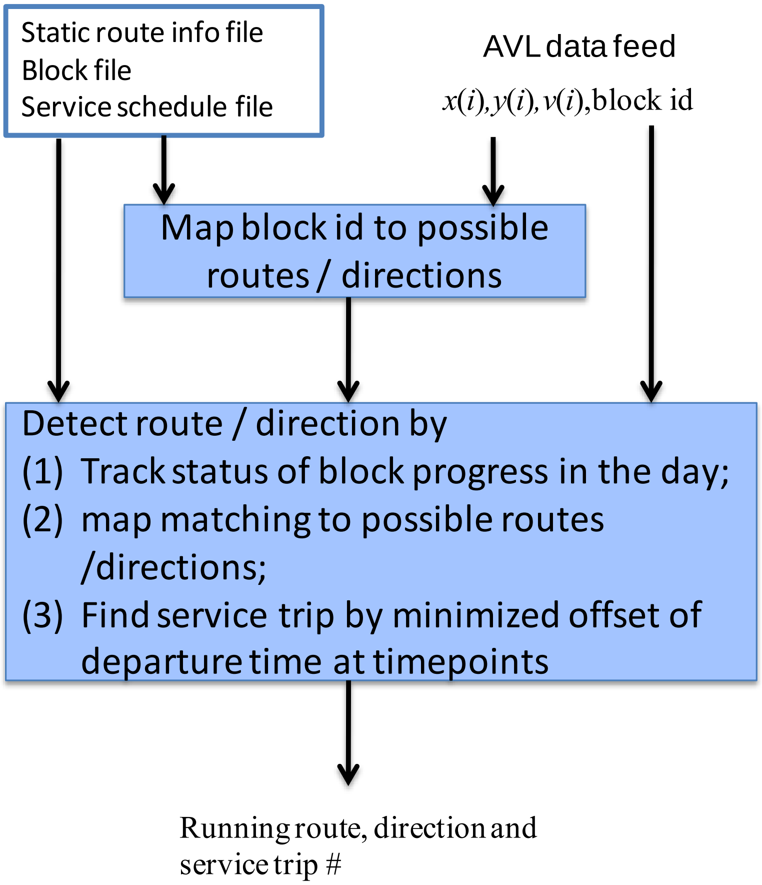

Static route info file
Block file
Service schedule file
AVL data feed
x(i), y(i), v(i), block id
Map block id to possible routes / directions
Detect route / direction by
(1) Track status of block progress in the day;
(2) map matching to possible routes /directions;
(3) Find service trip by minimized offset of departure time at timepoints
Running route, direction and service trip #

Figure 3-13 Pre-Processing for AVL Data with Location, Speed and Block Id

Figure 3-13 shows the processing procedure for the case when AVL data contain only location, speed and block id (registered by the operator), which is the case for AC Transit or the VTA BRT line as can be seen in Table 3-21. Two step processing is used, where the first step is the one-to-many mapping from a block to routes and directions and the second is to find the exact route and direction, and the service trip by tracking the progress of the bus moments within a block and map matching of the actual route to geographical database.

For the situation with better AVL data when running route, direction and time point offset is available as is the case with SamTrans, the preprocessing needed is to find the service trip in a day, which can be easily done by matching the schedules at given time point with the real-time data.

3.3.3. Real-time DPI Information Generation

The output of the data aggregation of real-time data feed are the unified formatted data feed into the system. These data include information about the vehicle, the agency, route, direction, service type, trip, location and historical statistics.

For the DPI information generation purpose, the following information is needed:

- For each bus, approaching stop(s) and estimated time of arrivals;
- For each stop and each route, the next arrivals (number of arrivals depend on the availability of the real-time data);
- Actual arrival/departure time at the bus stop when the bus has already arrived at / departed from that bus stop; and
- The travel time between given two stops for a certain bus (or equivalently a trip of a route);

All the information is based on a core module: the arrival time prediction. This prediction module has been addressed in section 3.2. The outputs of the prediction module are then converted into different formats and saved into the database for concurrent queries by upper layer applications for the information listed above.

3.3.4. **DPI Information Process**

The DPI information is provided to travelers under different situations, and for each situation we will present the information process to illustrate how the information is generated and presented by the system. DPI information process discussed below includes dynamic passenger information process for bus station, pre-trip planning or mobile DPI information retrieval, and information display onboard a bus. The designed ATSP/DPI system supports all the three processes but only the first process – dynamic passenger information process for bus station, was field tested.

- Dynamic passenger information process for bus station

  The dynamic passenger information provided at a bus station may be queried results for a given location (bus stop) and given direction. These results may include dynamic arrival time of upcoming buses of the routes there, (dynamic) location map with overlaid vehicles, route information, etc.

  This service will be provided as an internet service. A dedicated computer at or near the bus stop will subscribe/query to the service provider and display the information directly or publish the information via low-cost field display(s). The information process is shown in Figure 3-14.

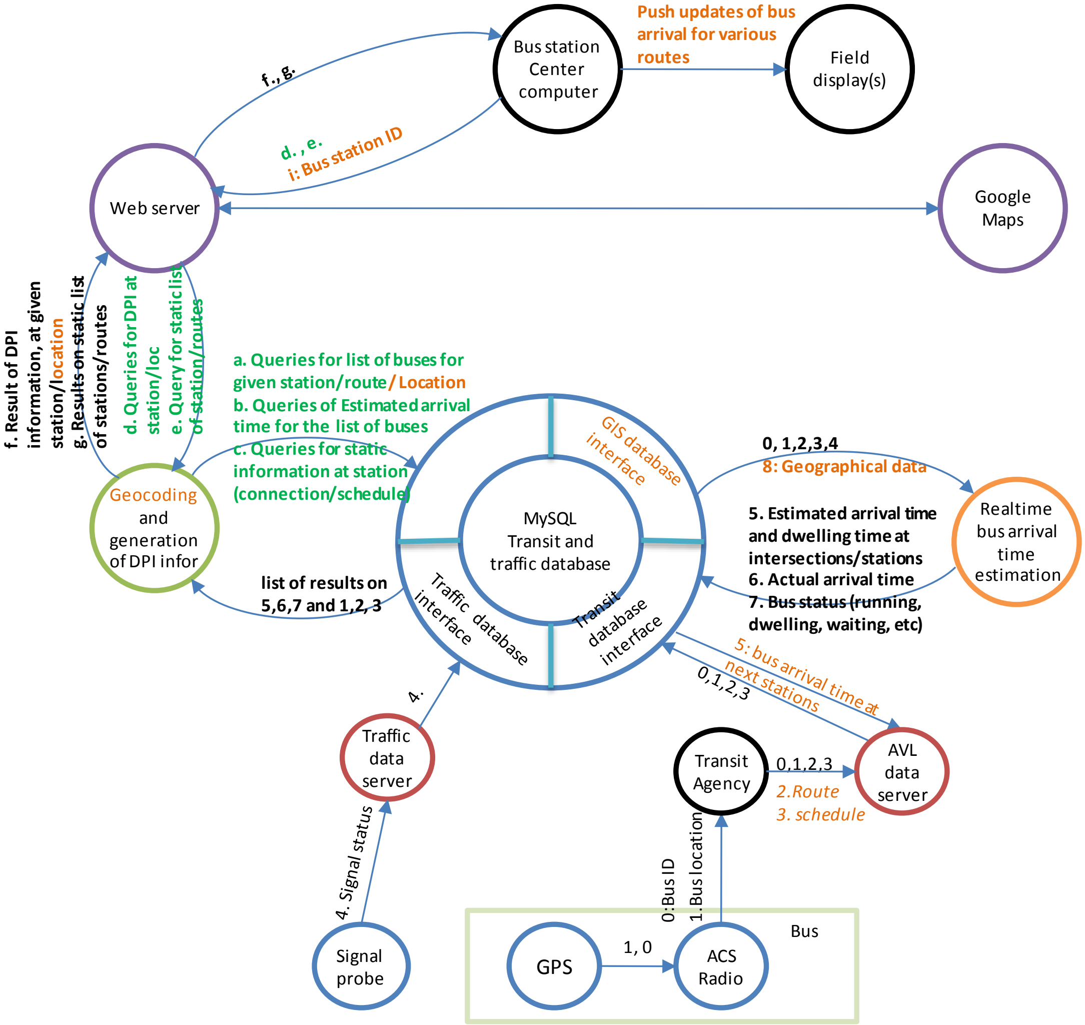

Bus station Center computer
Push updates of bus arrival for various routes
Field display(s)
f., g.
d. , e.
i: Bus station ID
Web server
Google Maps
f. Result of DPI information, at given station/location
g. Results on static list of stations/routes
d. Queries for DPI at station/loc
e. Query for static list of station/routes
a. Queries for list of buses for given station/route / Location
b. Queries of Estimated arrival time for the list of buses
c. Queries for static information at station (connection/schedule)
Geocoding and generation of DPI infor
list of results on 5,6,7 and 1,2, 3
GIS database interface
MySQL Transit and traffic database
Traffic database interface
Transit database interface
0, 1,2,3,4
8: Geographical data
Realtime bus arrival time estimation
5. Estimated arrival time and dwelling time at intersections/stations
6. Actual arrival time
7. Bus status (running, dwelling, waiting, etc)
5: bus arrival time at next stations
0,1,2,3
4.
Traffic data server
4. Signal status
Signal probe
Transit Agency
0,1,2,3
2.Route
3. schedule
AVL data server
0:Bus ID
1.Bus location
Bus
GPS
1, 0
ACS Radio

Figure 3-14 Information Process for Dynamic Passenger Information at Bus-Stop

- Pre-trip planning or mobile DPI information retrieval

  The required information: Real-time service availability, fare, transfer opportunities, real-time connection and travel time.

  Other data may include: nearest transit to the origin/destination; parking availability at transit stop(s); route details; location map; other related information. The information process is shown in Figure 3-15.

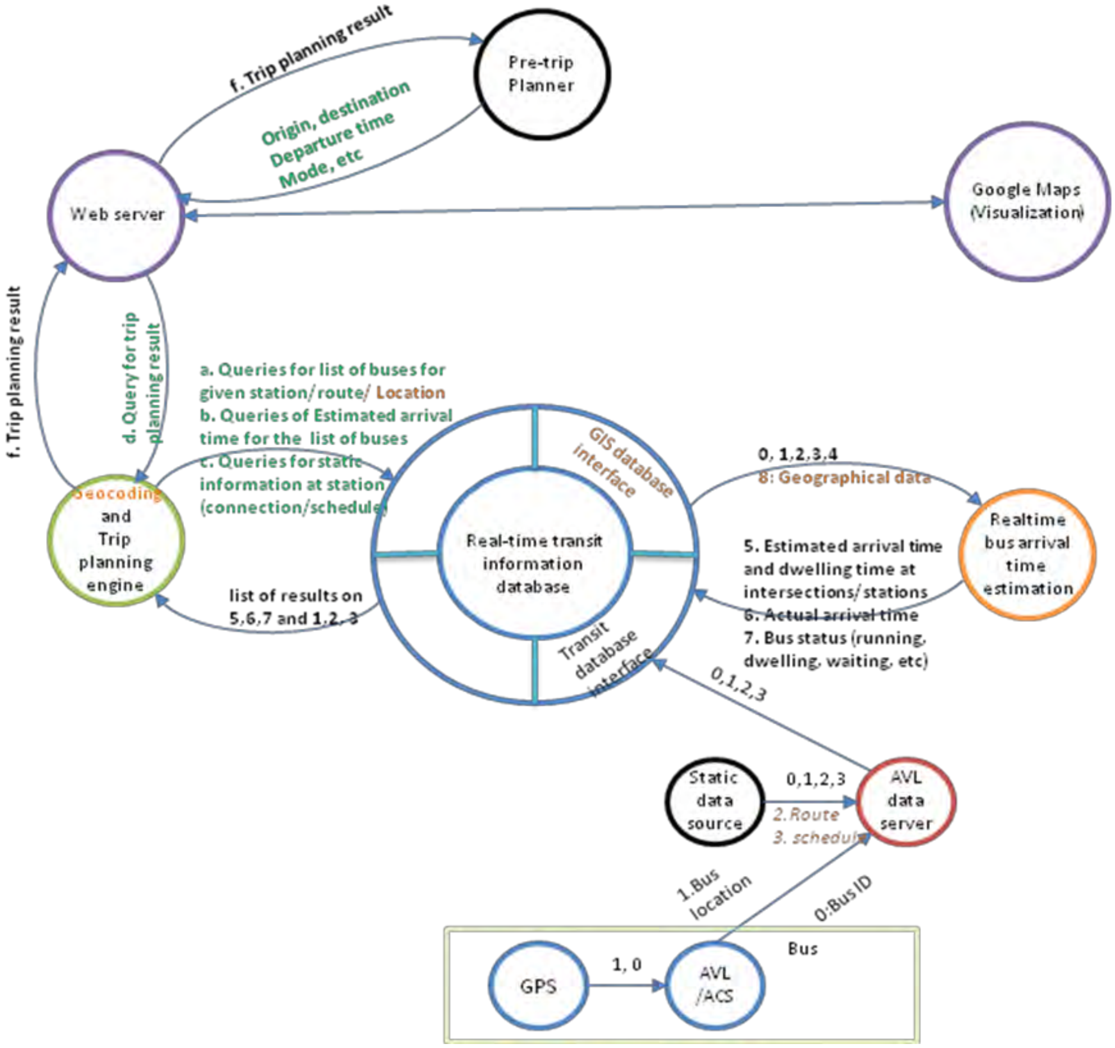

f. Trip planning result
Pre-trip Planner
Origin, destination
Departure time
Mode, etc
Web server
Google Maps (Visualization)
f. Trip planning result
d. Query for trip planning result
a. Queries for list of buses for given station/route/ Location
b. Queries of Estimated arrival time for the list of buses
c. Queries for static information at station (connection/schedule)
Geocoding and Trip planning engine
list of results on 5,6,7 and 1,2,3
GIS database interface
Real-time transit information database
Transit database interface
0, 1,2,3,4
8: Geographical data
Realtime bus arrival time estimation
5. Estimated arrival time and dwelling time at intersections/stations
6. Actual arrival time
7. Bus status (running, dwelling, waiting, etc)
0,1,2,3
Static data source
0,1,2,3
2. Route
3. schedule
AVL data server
1.Bus location
0:Bus ID
Bus
GPS
1, 0
AVL /ACS

Figure 3-15 Information Process of Pre-Trip Planning and En-Route Information Retrieval

- Information display onboard a bus
  In this case, the information is still generated at the ATSP/DPI system and then sent back to the bus via the two-way communication channel of the ACS system. The information is then displayed on an onboard computer. This information process would not require extra communication capability beyond the existing ACS equipment on the bus therefore would be more cost effective. The information process is shown in Figure 3-16.

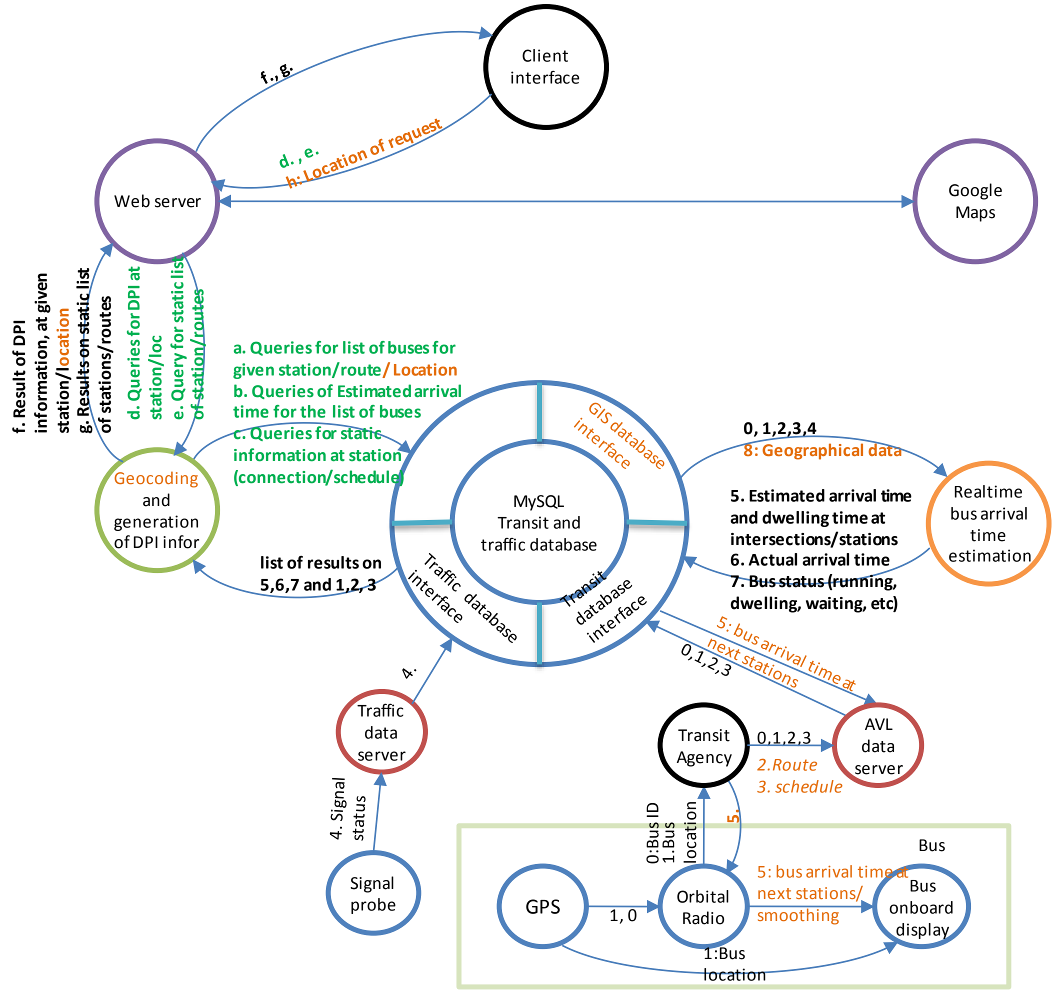


Figure 3-16 Information Process for Onboard Display of DPI Information

However, when a commercial high bandwidth wireless communication channel is available on the bus, then the information process is different such that the process can be very similar to that of the display at the bus stop. The information can be directly retrieved via an internet connection to the DPI server instead of via the ACS system.

3.3.5. DPI Application Design for Display at Bus-Stops

As has been discussed in the previous subsection 3.3.4, the information process for various DPI applications, either it is pre-trip planning or en route application, can share similar information process and open technology, which has the advantage of better compatibility of various kinds of communication channels, computer hardware and better scalability for different number of clients and users.

Adopting web-platform for information display is becoming more and more popular in real-world practice, and this is equally true for the DPI applications. Using the web platform as the basis for open platform DPI information delivery has many advantages over other traditional non-web based platforms. Table 3-22 shows the advantages of web-based approach.

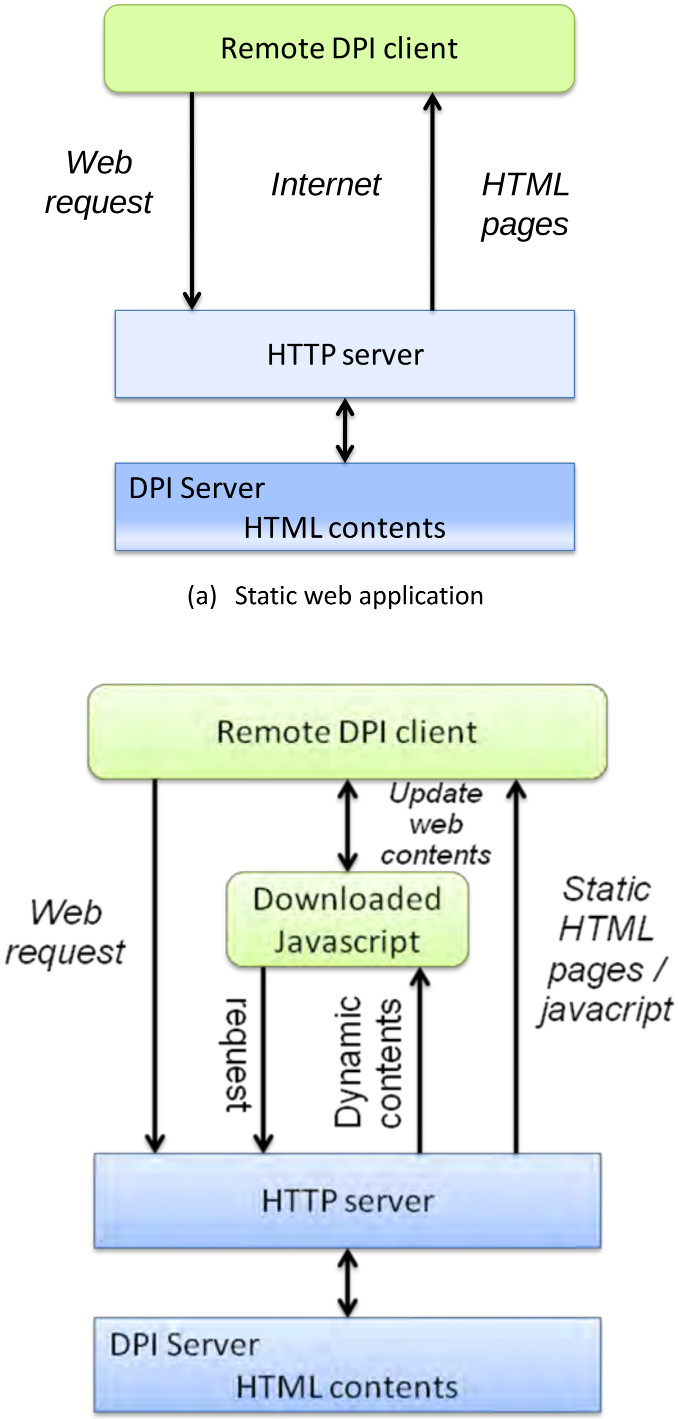
Table 3-22 Advantages of using Web-Based Platform for DPI Information Delivery

	Web-based platform	**Non-web-based platform (Such as the old system previously deployed at SamTrans)**
Open platform	✓ Yes	Usually not
Standardization	✓ Highly standardized	Less standardized, usually includes proprietary protocols
Development efforts	✓ Low, Good support of available tools and software modules	Higher
Compatibility with hardware / software systems	✓ Better: Web support by computers and operating systems is prevalent.	Depends on the design
Cross-platform migration efforts	✓ Very low: usually takes no efforts to display web based application on different platform due to the standardization	Depends on the design (can be very high)

The major concern of using a web-based platform for information delivery is usually performance related. However, this is less an issue for DPI applications when the performance requirement of showing the bus arrival is very minimal. The fact that hardware performance keeps enhancing and even a very low-end computer nowadays can handle the web display of DPI applications without a problem, therefore the hardware performance is no longer a problem for DPI display.

A. Web technology framework for DPI applications

The information to be displayed for DPI applications is essentially dynamic. This would require the underlying framework to support dynamic contents seamlessly for better performance and reliability.



(a) Static web application

(b) Dynamic web application using Ajax

Figure 3-17 Static Web versus Dynamic Web Application using Ajax

Ajax (Asynchronous Javascript and XML) is a popular framework for displaying dynamic contents using the web technology (Wikipedia). Figure 3-17 compares the information process between static Web application and dynamic Web application using Ajax. Ajax provides a flexible and asynchronous way to update the web content in the background. This technology framework fits into the need of our DPI display perfectly:

- it allows using standardized technology,
- it is web-only, therefore is cross-platform portable;
- it does not require development on the client side since only a web browser would be needed in the kiosk.

Reliability of the DPI display is another major concern when it is deployed as a public service. That would require high service availability, which will require the DPI system to overcome the frequently encountered outages of the communication channel, and / or the outage of the server where the real-time information is provided.

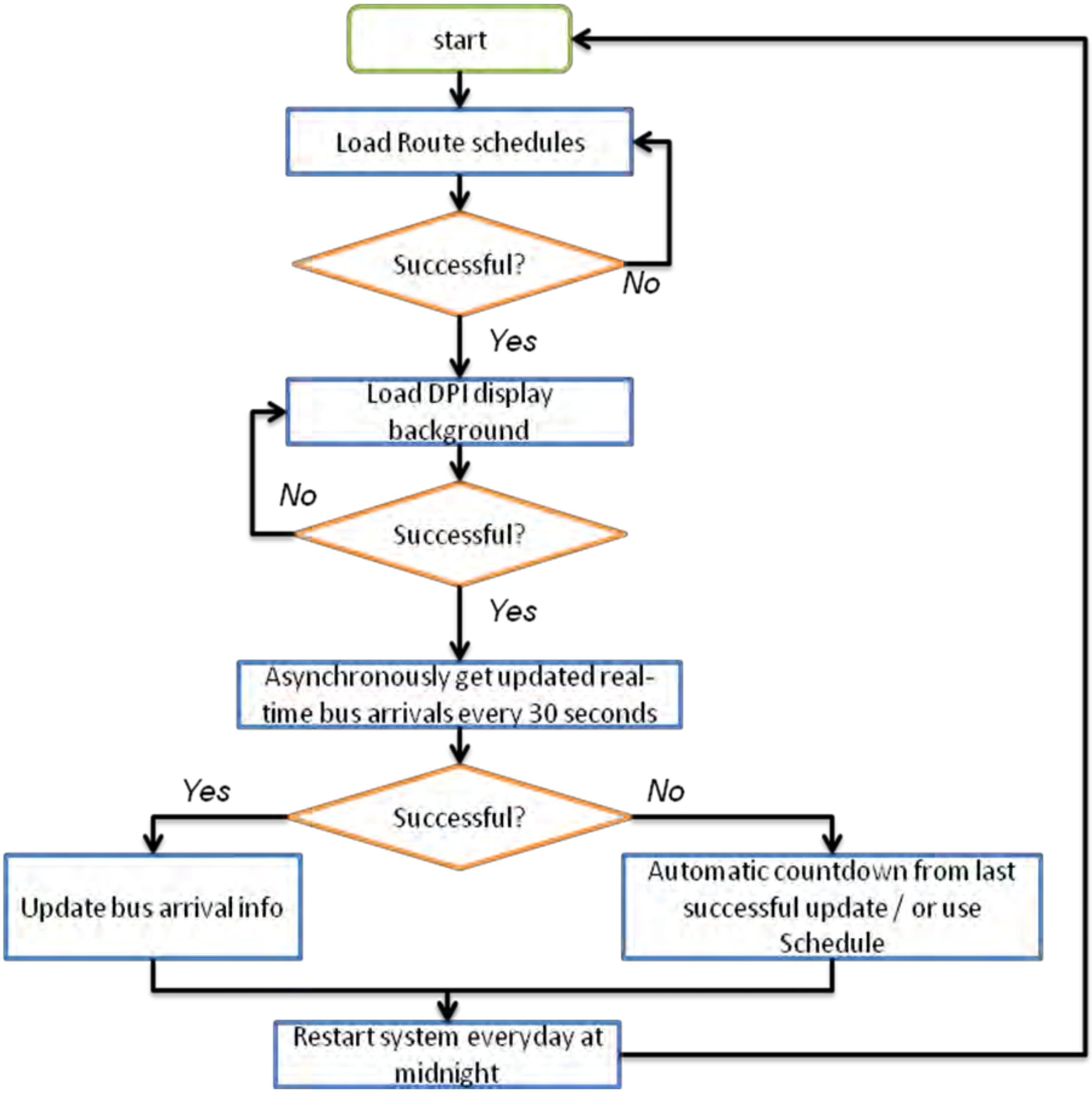

start
Load Route schedules
Successful?
No
Yes
Load DPI display background
No
Successful?
Yes
Asynchronously get updated real-time bus arrivals every 30 seconds
Yes
Successful?
No
Update bus arrival info
Automatic countdown from last successful update / or use Schedule
Restart system everyday at midnight

Figure 3-18 Flow Chart for the DPI Display based on Ajax Technology

Figure 3-18 shows the flowchart of the Ajax-based client-side logic, which is fulfilled by the Javascript programming language developed and saved on the DPI server and automatically downloaded and executed by the web browser when loading the DPI website. The Javascript technology, as an essential part of this Ajax framework, enables the client computer to run cross-platform software which is developed as part of the web-site locally at run-time. This execution only requires internet connectivity at the first load of the static web site, and thereafter would be running locally with the capability of handling failures of communication outage and server failures.

We also note that the implementation of the automatic restart as shown in Figure 3-18 should be fulfilled by the system scheduling software instead, rather than a part of the Javascript code, which is supposed to be running in a sandbox and would not have the privilege to control the system at runtime[5].

B. DPI display at Millbrae transit center

We have deployed the display of the integrated predictive bus arrival time information at the Millbrae transit center, starting from July 2010.

Millbrae transit center is the largest intermodal terminal in the United States west of Mississippi[6]. It provides cross-platform connections for BART, Caltrain, and 4 SamTrans bus lines, i.e., 359, 390, 391 and 397. Predicted bus arrival information for the 4 SamTrans lines are displayed at the Millbrae station. Figure 3-19 shows a bird eye's view of the Millbrae transit center. The kiosk signage with 4 screens is on the inside bottom-right corner of the picture.

[5] Javascript security, http://en.wikipedia.org/wiki/JavaScript#Security

[6] see http://en.wikipedia.org/wiki/Millbrae_Intermodal_Terminal

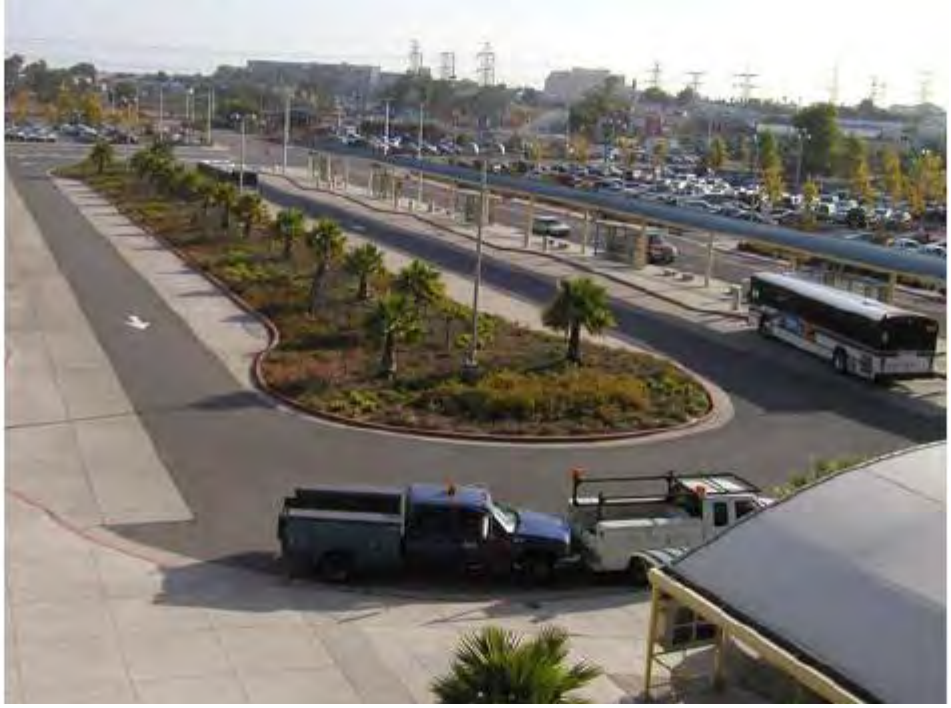


Figure 3-19 Bird Eye's View of the SamTrans Bus Bays and the Signage (Down Right Corner, Picture Courtesy of MTC)

Figure 3-20 shows a picture taken at the Millbrae transit center, in front of the SamTrans information kiosk (c.f. Figure 3-19 for the overview of the station, from reference (Wilbur Smith Associates / Harley & Associates, 2006)). The displayed information is from a website that is developed based on the integrated ATSP/DPI system following the technology framework as has been discussed in a previous subsection.

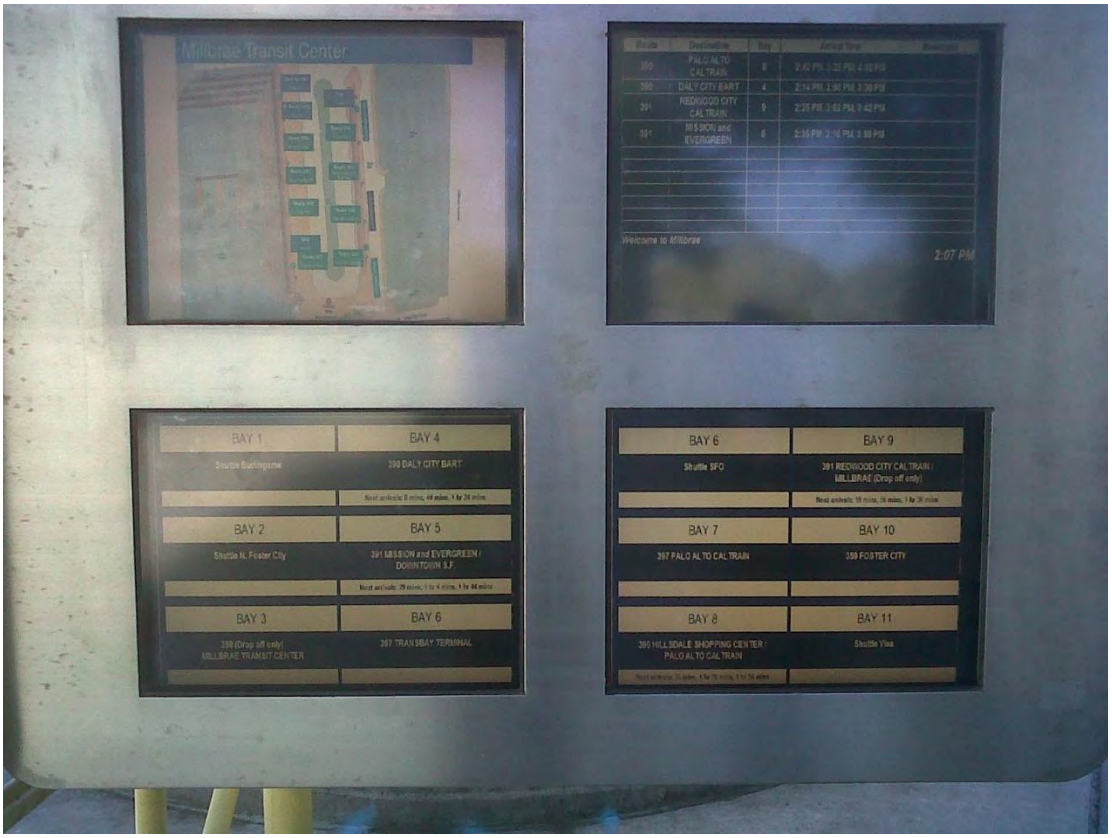
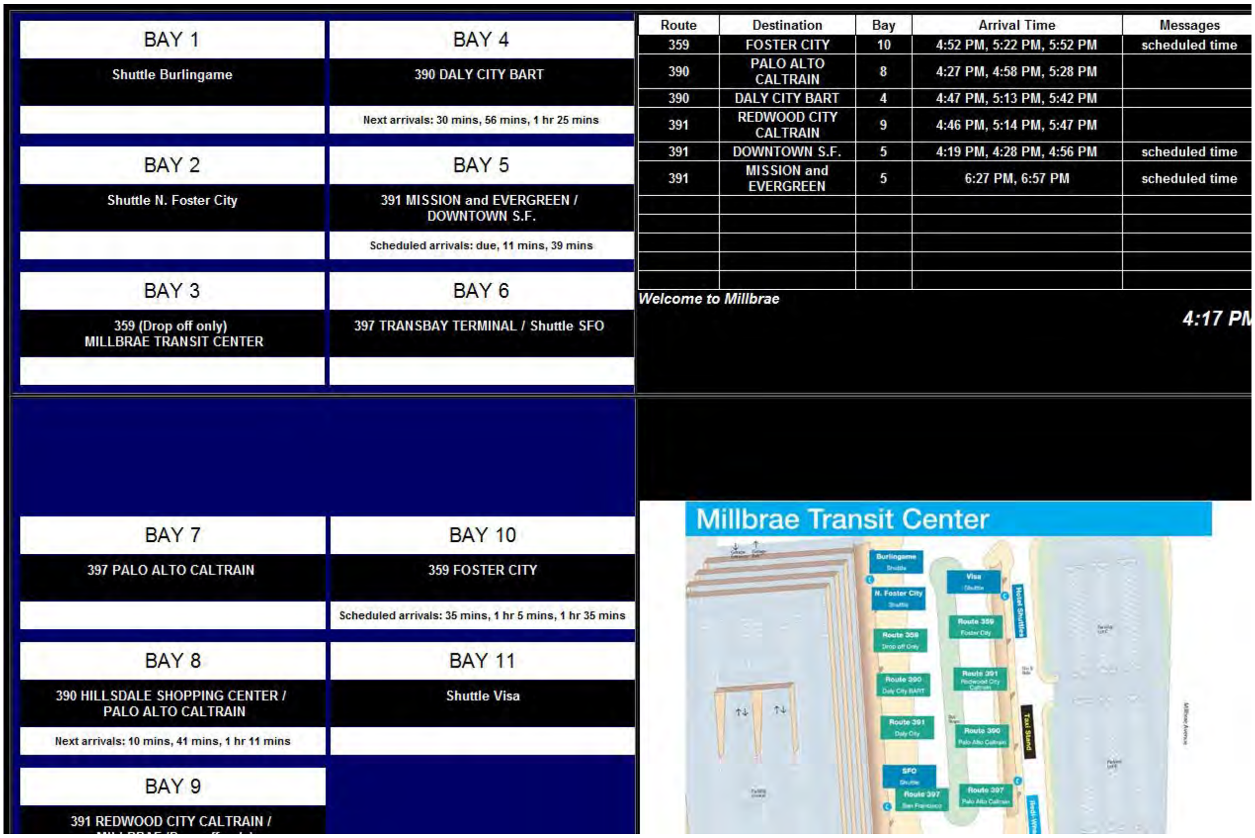


Figure 3-20 The Picture Taken at the Millbrae Transit Center

The web design of the specific display for a Millbrae kiosk can be seen in Figure 3-21, where the layout can be clearly seen to have been specifically designed to match the kiosk layout with four monitors (California PATH, 2010).



BAY 1

Shuttle Burlingame

BAY 4

390 DALY CITY BART

Next arrivals: 30 mins, 56 mins, 1 hr 25 mins

BAY 2

Shuttle N. Foster City

BAY 5

391 MISSION and EVERGREEN / DOWNTOWN S.F.

Scheduled arrivals: due, 11 mins, 39 mins

BAY 3

359 (Drop off only)
MILLBRAE TRANSIT CENTER

BAY 6

397 TRANSBAY TERMINAL / Shuttle SFO

BAY 7

397 PALO ALTO CALTRAIN

BAY 10

359 FOSTER CITY

Scheduled arrivals: 35 mins, 1 hr 5 mins, 1 hr 35 mins

BAY 8

390 HILLSDALE SHOPPING CENTER / PALO ALTO CALTRAIN

Next arrivals: 10 mins, 41 mins, 1 hr 11 mins

BAY 11

Shuttle Visa

BAY 9

391 REDWOOD CITY CALTRAIN /

Route	Destination	Bay	Arrival Time	Messages
359	FOSTER CITY	10	4:52 PM, 5:22 PM, 5:52 PM	scheduled time
390	PALO ALTO CALTRAIN	8	4:27 PM, 4:58 PM, 5:28 PM	
390	DALY CITY BART	4	4:47 PM, 5:13 PM, 5:42 PM	
391	REDWOOD CITY CALTRAIN	9	4:46 PM, 5:14 PM, 5:47 PM	
391	DOWNTOWN S.F.	5	4:19 PM, 4:28 PM, 4:56 PM	scheduled time
391	MISSION and EVERGREEN	5	6:27 PM, 6:57 PM	scheduled time

Welcome to Millbrae

4:17 PM

Figure 3-21 Web Design of the Millbrae Display

The predictive bus arrival times are organized in two different formats, as can be seen in the web design (Figure 3-21). Two monitors on the left show the information sorted by the bay number at the station, while the table list display on the upper right monitor shows the arrival times sorted by their route name and directions.

C. *Web-based DPI display at bus stops*

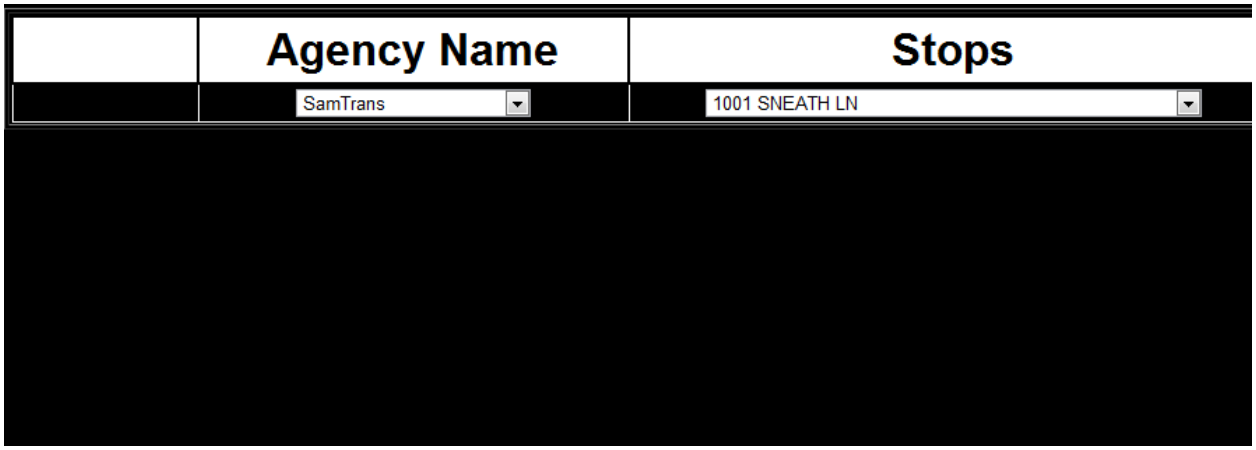
In addition to the deployed web-based DPI display, we have also developed a generalized tool that can be used for DPI information display at any given bus stop, using the aforementioned Ajax-based web framework.

Real-time predictive arrival time for any given bus stop or train station and for any bus routes that are served at that stop is shown automatically.

The site is available online at http://tlab.path.berkeley.edu:8080/DPIStops/lists.html. It currently supports real-time predictive arrival times from almost all SamTrans routes using the integrated ATSP/DPI system (California PATH, 2011).

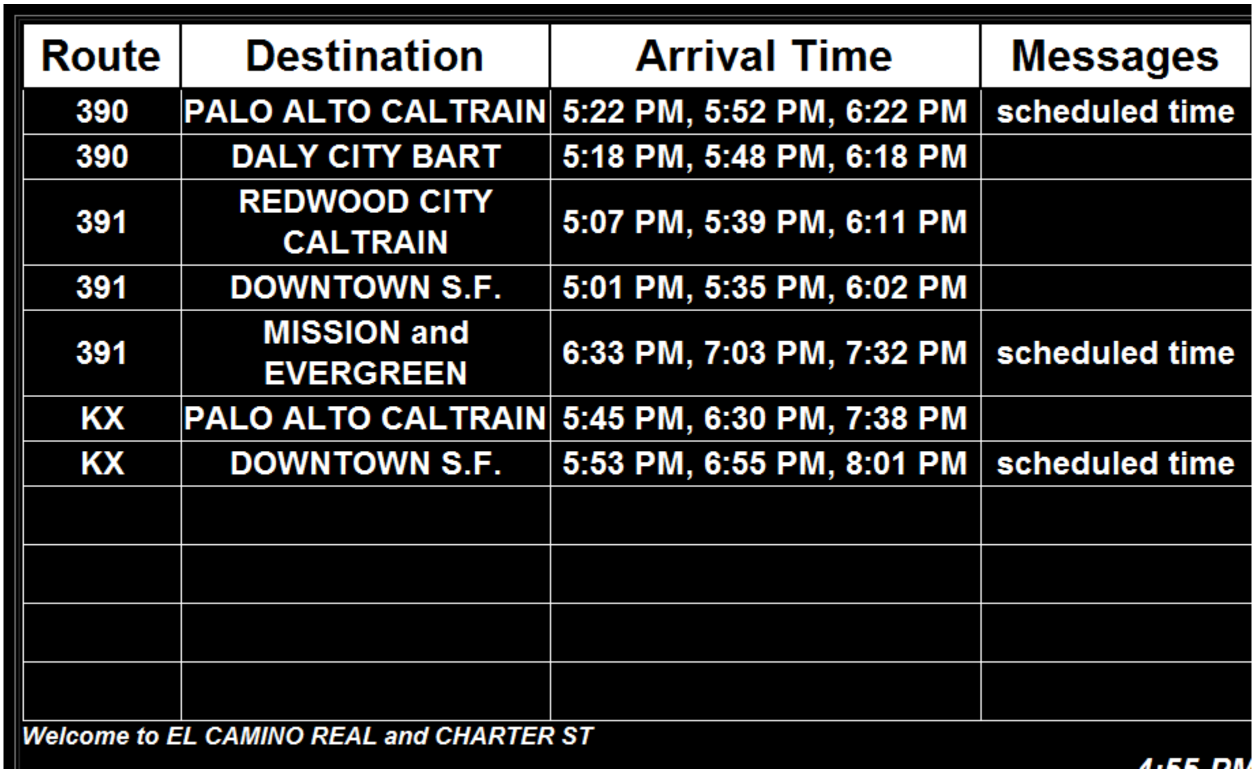
To display the DPI information from a certain bus stop, there are two ways to specify the parameter for the information web tool: (1) using the URL parameter with station id, that is defined by the XML data specification files; or (2) selecting the stop manually by using the web interface.

Figure 3-22 shows via the web interface that a bus stop can be selected manually. When a bus stop is selected, the DPI display for all routes serving that stop will be popped up and loaded with data using the Javascript software flow chart as specified in Figure 3-18. Figure 3-23 is such an example when El Camino Real and Charter Street is selected as the bus stop. Another example is shown in Figure 3-24 where the SamTrans bus stop “Palo Alto Caltrain” is selected.



	Agency Name	Stops
	SamTrans	1001 SNEATH LN

Figure 3-22 Selecting a Bus-Stop from the DPI Information Display Website

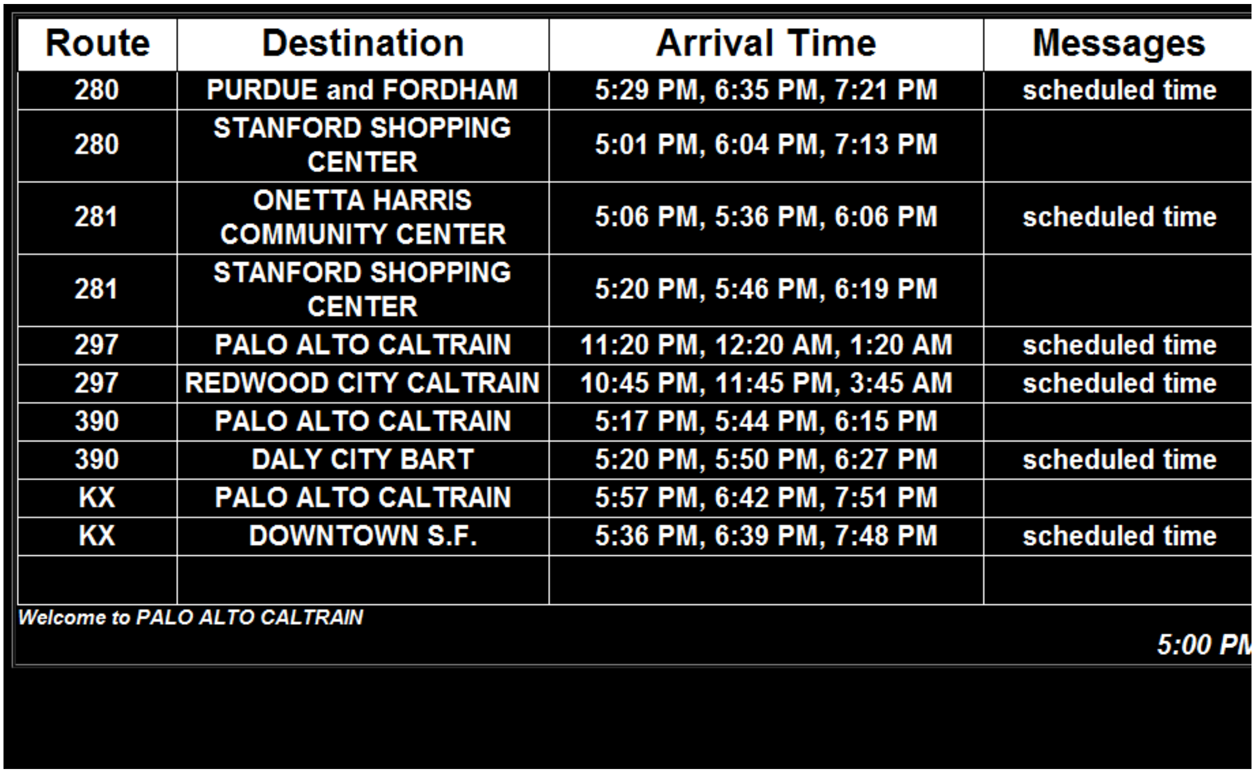


Route	Destination	Arrival Time	Messages
390	PALO ALTO CALTRAIN	5:22 PM, 5:52 PM, 6:22 PM	scheduled time
390	DALY CITY BART	5:18 PM, 5:48 PM, 6:18 PM	
391	REDWOOD CITY CALTRAIN	5:07 PM, 5:39 PM, 6:11 PM	
391	DOWNTOWN S.F.	5:01 PM, 5:35 PM, 6:02 PM	
391	MISSION and EVERGREEN	6:33 PM, 7:03 PM, 7:32 PM	scheduled time
KX	PALO ALTO CALTRAIN	5:45 PM, 6:30 PM, 7:38 PM	
KX	DOWNTOWN S.F.	5:53 PM, 6:55 PM, 8:01 PM	scheduled time

Welcome to EL CAMINO REAL and CHARTER ST

4:55 PM

Figure 3-23 Sample DPI Display for a Bus-Stop: El Camino Real and Charter St.



Route	Destination	Arrival Time	Messages
280	PURDUE and FORDHAM	5:29 PM, 6:35 PM, 7:21 PM	scheduled time
280	STANFORD SHOPPING CENTER	5:01 PM, 6:04 PM, 7:13 PM	
281	ONETTA HARRIS COMMUNITY CENTER	5:06 PM, 5:36 PM, 6:06 PM	scheduled time
281	STANFORD SHOPPING CENTER	5:20 PM, 5:46 PM, 6:19 PM	
297	PALO ALTO CALTRAIN	11:20 PM, 12:20 AM, 1:20 AM	scheduled time
297	REDWOOD CITY CALTRAIN	10:45 PM, 11:45 PM, 3:45 AM	scheduled time
390	PALO ALTO CALTRAIN	5:17 PM, 5:44 PM, 6:15 PM	
390	DALY CITY BART	5:20 PM, 5:50 PM, 6:27 PM	scheduled time
KX	PALO ALTO CALTRAIN	5:57 PM, 6:42 PM, 7:51 PM	
KX	DOWNTOWN S.F.	5:36 PM, 6:39 PM, 7:48 PM	scheduled time

Welcome to PALO ALTO CALTRAIN

5:00 PM

Figure 3-24 Sample DPI Display for a Bus-Stop: Palo Alto Caltrain station

4. Integration of ATSP and DPI

The integration of ATSP/DPI with the existing transit AVL/ACS systems has the following three aspects:

1) Integrating DPI with AVL/ACS;
2) Integrating ATSP with AVL/ACS using emulated dynamic polling; and
3) Integrating DPI with ATSP

Today's bus AVL/ACS systems include both the core location tracking capabilities and information exchange capabilities between fleet vehicles and dispatch for real-time fleet operations management. The primary onboard components of AVL/ACS systems include (TRB, 2008)

- Global position system (GPS) receiver and antenna;
- Vehicle logic unit (VLU) computer;
- Mobile data terminal (MTD) operator interface terminal;
- One or more radios and antennas to provide wide-area voice and data communications;
- Automatic passenger count (APC) subsystem;
- Automated onboard announcements subsystem; and
- Data network to support communications between onboard devices.

The primary central components include

- Central server;
- Communications gateways;
- Central database; and
- Management software.

Upon bus operator logging in to a run, the AVL/ACS onboard system continuously determines bus location using second-by-second GPS date and tracks schedule and route adherence in real-time. The central system initiates polling requests to get reports from the entire bus fleet. Within each polling cycle, every bus in the fleet is polled by turn, and the polled bus responses with a message that contains its location, schedule and route adherence status data. In other words, each bus in the fleet is reporting to dispatch on a frequent periodic basis. The polling rate (i.e., the polling cycle interval) is determined by the capacity of the mobile data communications system, the fleet size and the polling strategies. Today's bus AVL/ACS systems utilize data channel to carry out bus polling. The utilization rate of AVL/ACS channel capacity for this regular bus polling is about 60 to 70 percent. The remaining capacity is reserved and used as a contention channel for meeting other fleet management application needs, such as voice communications and emergency text messages. With the strategy of polling every bus in the fleet by turn and utilizing data channel only, current AVL/ACS systems usually have a polling interval in the range of 1 to 2 minutes. The polling rate for SamTrans' AVL/ACS system is around 2 minutes.

Major attributes of bus real-time reporting messages include bus ID, reporting time, route ID, direction, schedule deviation, next time-point ID and scheduled time, latitude and longitude. Combining with the static transit route file and schedule file, bus reporting message provides a one-to-one match of current bus trip with the trip (run) sequence ID in the Schedules table (Table 3-7). The detailed real-time transit data pre-processing and trip (run) matching was described in Section 3.3.2 and illustrated in Figure 3-13.

Bus reporting messages already contain primary data elements for the generation of dynamic passenger information. Therefore, the integration of DPI with AVL/ACS does not necessarily require any modifications on the existing AVL/ACS systems. The central processor of the integrated system, as described in Section 2.3.6 and illustrated in Figure 2-9, gathers bus reporting messages through the data interface to AVL/ACS, predicts bus arrival times at downstream bus-stops whenever a new bus reporting message was received, generates dynamic passenger information, and disseminates the information to the users.

The bus polling rate at minute-level is not practical to support transit signal priority application. A bus could have passed a couple of intersections within one polling cycle. Therefore, the integration of ATSP with AVL/ACS does require modifications on the existing AVL/ACS systems to overcome the limitation of a low bus polling rate. A contention channel provide additional channel capacity for getting bus location and status reports therefore can be used for ATSP purposes. As described in Section 1.3, a dynamic polling algorithm was designed and tested in simulation under the previous project of developing the ATSP system. The algorithm utilizes a contention channel for inserting additional polls for buses that need signal priority. Implementing the dynamic polling algorithm to work with the existing AVL/ACS system is one of the major tasks for this project.

Integrating DPI with ATSP can provide users better DPI information. It is commonly accepted that TSP reduces bus travel time as well as its reliability. With the integration, information about TSP savings is shared with the prediction of bus arrival time at bus-stops thereby producing better predicted results. In addition, inserted polls for ATSP purpose provide additional bus location and status reports which can be used for DPI information generation to provide more frequent and accurate predicted bus arrival information at bus-stops.

4.1. Integrated ATSP/DPI and AVL/ACS

One of the critical elements for integrated ATSP/DPI and AVL/AS is to enable the existing AVL/ACS systems to support need-based bus polling for ATSP operation, through dynamic polling. Dynamic polling needs to address two questions: 1) which bus to poll and 2) when to poll. The polling strategy used for the existing bus polling though data channel is for the central server of the AVL/ACS system to initiate and poll each vehicle in the fleet by turn. Applying this same polling strategy for inserted polls to be carried out through a contention channel is not sufficient to serve an ATSP purpose. At most it doubles the polling frequency even when all channel capacity is being fully utilized and the resulting polling rate is still at the minute-level. A need-based polling strategy for inserted polls is desired for the best use of additional capacity provided by contention channel. The generation of dynamic polling requests shall be based on bus status (the expected bus arrival time at the prioritized intersection) and intersection status (the expected bus phase at the time bus will arrive). As the onboard AVL/ACS system already keeps tracking bus location using second-by-second GPS data, a natural modification of onboard processing software for integrated ATSP/DPI with AVL/ACS is to include the capability of predicting bus arrival times at downstream intersections and to carry the predicted arrival information within bus reporting messages to the AVL/ACS central server. The central server, based on received predicted arrival information from buses and projected traffic signal status received from the integrated ATSP/DPI system, dynamically determines which bus and at what time for requesting inserted polls. Bus reporting messages from both regular and inserted polls are then forwarded to the ATSP/DPI system to serve for the generation of TSP requests and DPI information.

Figure 4-1 illustrates the necessary onboard and central system components of AVL/ACS for integrated ATSP/DPI and AVL/ACS. The blocks of “Intersection arrival time prediction” in the onboard system and “Dynamic polling via contention channel” in the central system are the necessary modifications required.

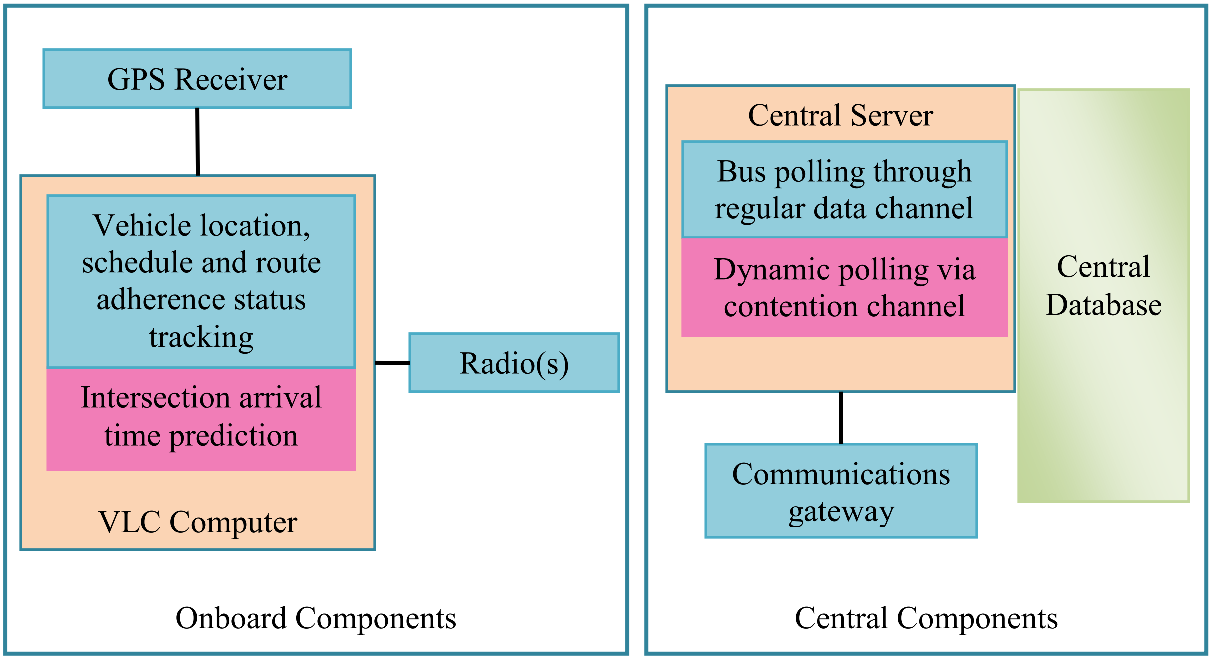

GPS Receiver
Vehicle location, schedule and route adherence status tracking
Intersection arrival time prediction
VLC Computer
Radio(s)
Onboard Components
Central Server
Bus polling through regular data channel
Dynamic polling via contention channel
Central Database
Communications gateway
Central Components

Figure 4-1 AVL/ACS System Components for Integrated ATSP/DPI

Due to the lack of access to both AVL/ACS onboard and central processing software, implementing the onboard “Intersection arrival time prediction” module and the central “Dynamic polling” module on the existing AVL/ACS system is not practical. An alternative approach was adopted to overcome this constraint and to test how AVL/ACS can support integrated ATSP/DPI. An ACS emulator was developed to mimic ACS channel management on both data and contention channels. The validation of integrated ATSP/DPI with ACS emulation through a field operation test is an important step towards the deployment of integrated

ATSP/DPI and AVL/ACS. It makes necessary preparations for commercial vendors to implement the dynamic polling algorithm on their AVL/ACS systems to fulfill the great benefits of integrated ATSP/DPI and AVL/ACS.

4.1.1. System Configuration of Integrated ATSP/DPI with ACS Emulation

Under the ACS emulation approach, the central server of AVL/ACS performs its regular bus polling though data channel and forwards bus reporting messages to the central processor of the integrated ATSP/DPI system. The ACS emulation processing module and the intersection arrival time prediction module are hosted by the ATSP/DPI central processor, with the ACS emulation module mimicking ACS operations in the way it would be when dynamic polling is implemented, and the intersection arrival time prediction module serving as the "pseudo" onboard processing software (see Figure 4-1).

When running onboard, the intersection arrival time prediction module uses second-by-second GPS data and schedule and route adherence status data as inputs. In order to have the "pseudo" module running on the ATSP/DPI central processor to receive the same types of data inputs, cell-phone-based AVL systems were installed on 15 SamTrans buses that serve bus routes 390 and 391 to record and send second-by-second GPS data to the integrated ATSP/DPI system, where the second-by-second GPS data are merged with the regular bus polling data that contain schedule and route adherence information, by matching bus ID and UTC time from two data sources.

The system configuration of integrated ATSP/DPI with ACS emulation is shown in Figure 4-2 and was used for the field operational tests. It is worth pointing out that the cell-phone-based AVL system installed on 15 buses is used to support the field operational tests. It is not required when dynamic polling is implemented on the AVL/ACS systems.

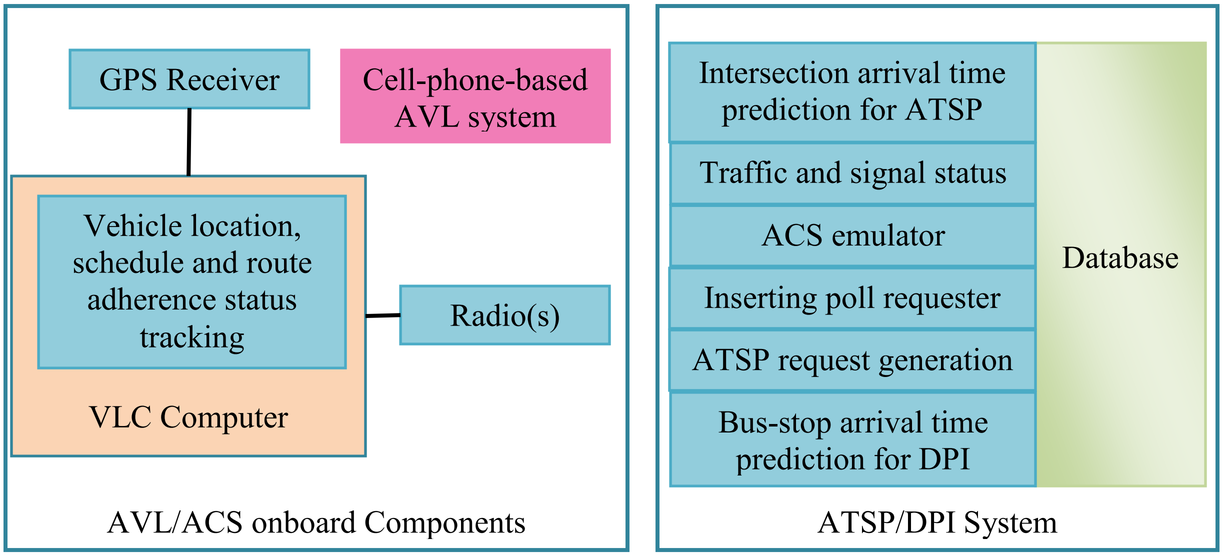
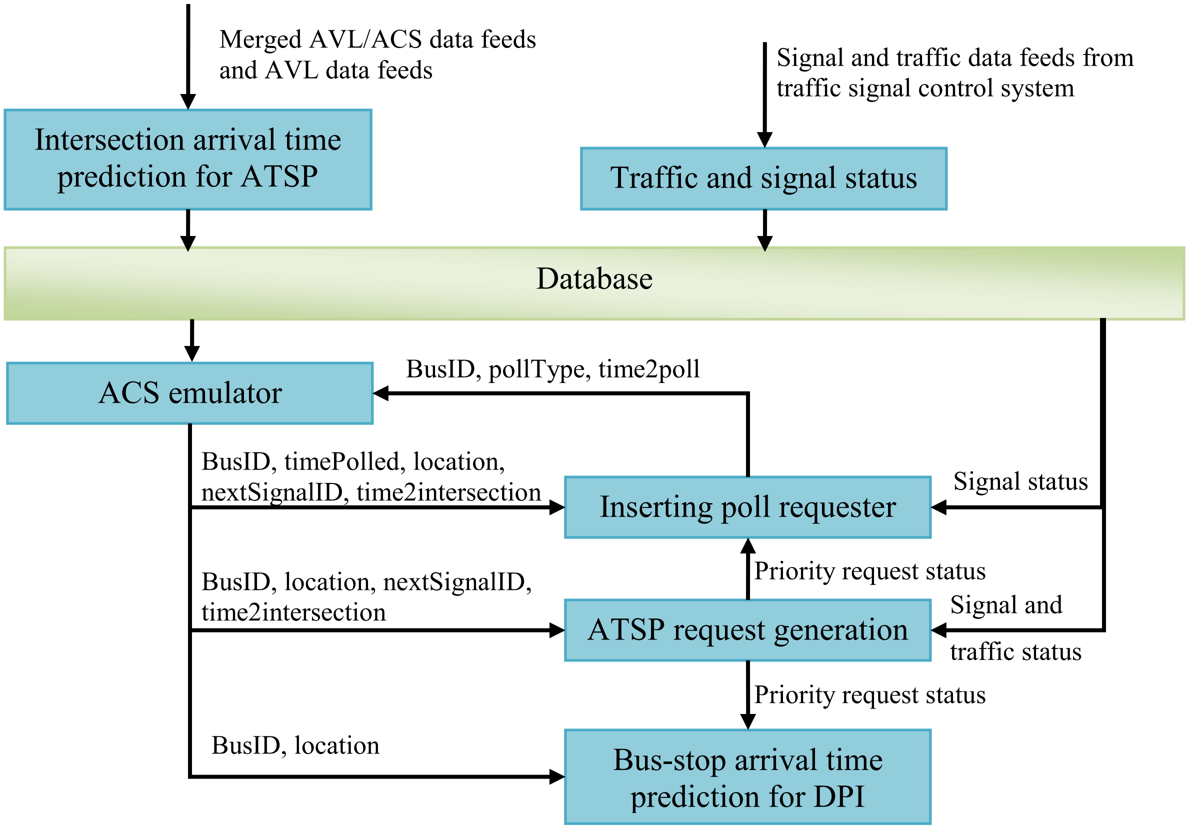

GPS Receiver
Cell-phone-based AVL system
Vehicle location, schedule and route adherence status tracking
VLC Computer
Radio(s)
AVL/ACS onboard Components
Intersection arrival time prediction for ATSP
Traffic and signal status
ACS emulator
Inserting poll requester
ATSP request generation
Bus-stop arrival time prediction for DPI
Database
ATSP/DPI System

Figure 4-2 System Configuration for Integrated ATSP/ACS with ACS Emulation

Figure 4-3 shows the block diagram for integrated ATSP/DPI with ACS emulation. Algorithms for prediction of intersection and bus-stop arrival time have been presented in Section 3.2. The design and the implementation of the ACS emulator and inserting poll requester will be addressed in Section 4.1.2 and Section 4.1.3, respectively.



Merged AVL/ACS data feeds and AVL data feeds
Signal and traffic data feeds from traffic signal control system
Intersection arrival time prediction for ATSP
Traffic and signal status
Database
ACS emulator
BusID, pollType, time2poll
BusID, timePolled, location, nextSignalID, time2intersection
Inserting poll requester
Signal status
Priority request status
BusID, location, nextSignalID, time2intersection
ATSP request generation
Signal and traffic status
Priority request status
BusID, location
Bus-stop arrival time prediction for DPI

Figure 4-3 Block Diagram of Integrated ATSP/DPI with ACS Emulation

The following describes the input, output and data exchange flow between processing modules.

A. Processing modules - Intersection arrival time prediction

Input to the intersection arrival time prediction is the merged data set of real-time regular bus polling data feed from AVL/ACS and the second-by-second GPS data feed from the cell-phone-based AVL system. Output from this module includes the data attributes contained within the regular bus polling data feed (bus ID, route ID, direction, latitude, longitude, etc.) as well as the predicted arrival information, which adds the attributes of the type of the approaching (next) node (either a bus-stop or a prioritized intersection), node ID, next signal ID, distances to the next node and next signal, and estimated arrival times at the next node and next signal.

When an instrumented bus is traveling along the implemented ATSP corridor, output from this module is on a second-by-second basis. If an instrumented bus is not traveling along the

ATSP corridor or the bus has not been instrumented with the cell-phone-based AVL system, this module simply outputs the attributes within the regular bus polling data feed and fills 0 values for attributes related to the predicted information. The update rate for the latter case is the same as the bus regular polling rate.

Output from this module is logged into the database and is used as input to the ACS emulation module.

B. Processing modules - Traffic and signal status

Input to the traffic and signal status module is the real-time data feed received from the traffic signal control system. The major data attributes include signal ID, control plan number, active phase and interval, master cycle clock, local cycle clock, and system loop counts and occupancies. For each intersection along the ATSP corridor, this module keeps tracking green time usages for all permitted phases and estimating traffic delay on every movement at the intersection. The classical Webster's delay model was implemented for the estimation of traffic delay.

Output from this module includes the estimated traffic delay and the current signal state (light color) and is used by the ATSP request generation module and the insert poll requester module.

C. Processing modules – ACS emulator

The ACS emulator module serves as the bridge between the existing AVL/ACS system and the integrated ATSP/DPI system. The ACS emulator manages both data and contention channel polling. It maintains a dynamic polling list regarding which bus (bus ID), what time (time2poll) and what type of channel (data or contention) to poll bus update. At each time slot, this module searches the list to select a bus and retrieves from the database the latest entry of the intersection arrival time prediction output associated with that bus, based on designated protocols. The retrieved data entry is then transmitted to the modules of the inserting poll requester, ATSP request generation and Bus-stop arrival time prediction. In other words, when an instrumented bus is traveling along the ATSP corridor, although the intersection arrival prediction module provides outputs on a second-by-second basis, inputs used for ATSP request generation and DPI generation are on a sampled basis, similar to the existing AVL/ACS bus polling.

Upon successfully retrieving the data entry (emulated polls), the ACS emulator updates the polling list regarding time2poll for the next polling of the associated bus: 2 minutes from the current polling time for regular polls and not-to-poll for inserting polls.

D. Processing modules – Inserting poll requester

The inserting poll requester module realizes the dynamic polling algorithm. It uses predicted arrival time at the approaching intersection and projected traffic light state (green/yellow/red) on the bus approach as input to dynamically determine time2poll for inserted polls, also based on designed rules. The requested inserting poll information (bus ID and time2poll) is sent to the ACS emulator module for carrying out polling.

E. *Processing modules – ATSP request generation*

Input to the ATSP request generation module includes output from ACS emulator regarding predicted arrival information at the approaching intersection and output from the traffic and signal status module (i.e., traffic intersection delay by movements). It optimizes green splits by minimizing the weighted bus delay and traffic delay, converts the resulting green splits to force-off points and sends the force-off points to the traffic signal control system to be executed by the local traffic signal controllers.

F. *Processing modules – Bus-stop arrival time prediction*

Input to the bus-stop arrival time prediction module consists of real-time bus reporting messages in response to both regular and inserted polling. Output is the predicted arrival time at all down-stream bus-stops and is used for the generation of DPI information.

4.1.2. **Implementing ACS Emulator**

The radio channel of SamTrans AVL/ACS system is divided into frames with 6 time slots per frame. Each time slot occupies one-sixth of a second. Out of the 6 time slots, four are used as a data channel and the other two as a contention channel. The central server of SamTrans AVL/ACS could poll and get reports from up to 4 buses in every second.

The design of ACS emulator follows this protocol. An interrupt mechanism is used to emulate the channel switching. At every one-sixth of a second (166 milliseconds), the central processor of the integrated ATSP/DPI system sends an interrupt signal to the ACS emulator process. In response, the ACS emulator process executes a signal handler to poll (or retrieve) the latest entry from the database for a selected bus, based on the following designated rules.

1) The ACS emulator process shall track the channel (or time slot) sequence number (1 through 6) and update it when an interrupt signal was caught;
2) Channel sequence of 1 through 4 shall be used only for regular polls and the sequence of 4 and 5 shall be used only for inserted polls;
3) Time gap between consecutive regular polls for any individual buses shall be not less than 2 minutes, the polling rate of SamTrans current AVL/ACS system;
4) The ACS emulator process shall maintain and update a polling list for both regular and inserted polls. The polling list contains the information about bus ID, expected polling time (time2poll), and polling channel (data or contention). Each individual bus has at most two entries on the polling list, one for regular polling and one for inserted polling;
5) New inserted polling request shall overwrite the previous request from the same bus;
6) Each emulated poll shall be carried out after its associated time2poll expires; and
7) Bus with longer elapsed time from its expected time2poll has higher priority to be selected as the polling bus.

Figure 4-4 shows the flow chart for the implemented ACS emulator.

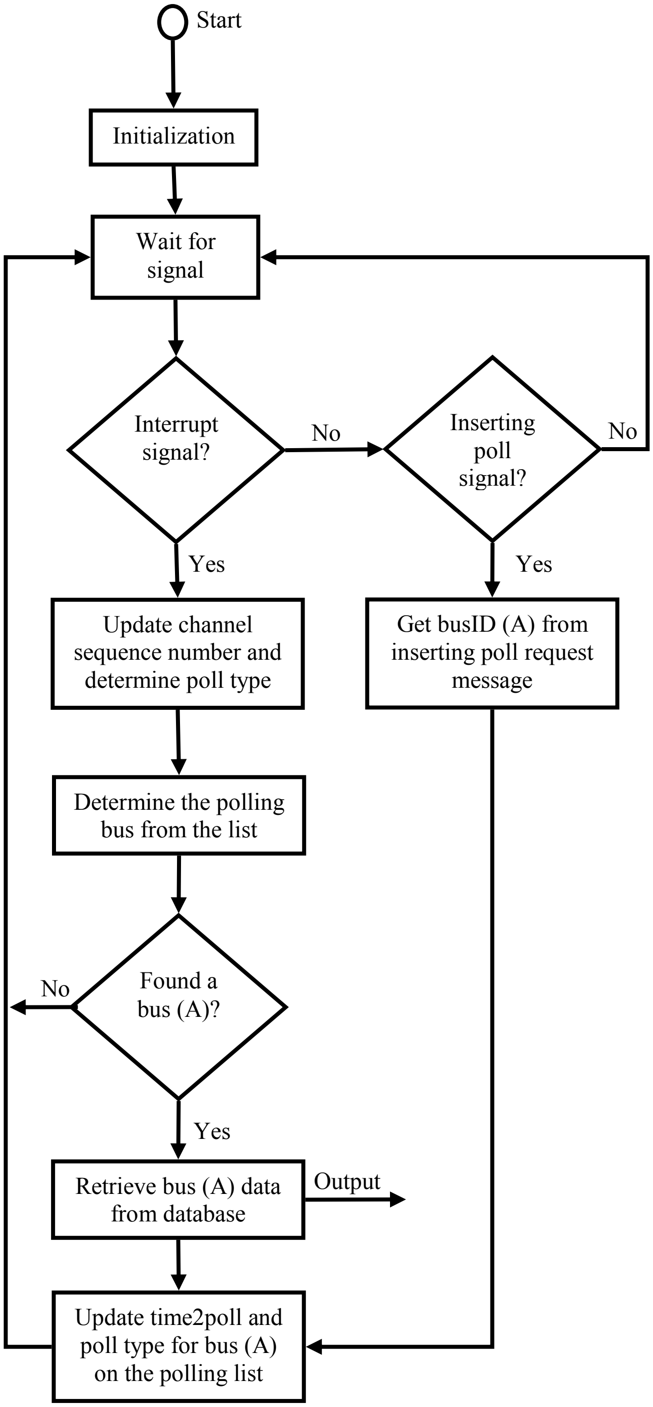

Start
Initialization
Wait for signal
Interrupt signal?
No
Inserting poll signal?
No
Yes
Yes
Update channel sequence number and determine poll type
Get busID (A) from inserting poll request message
Determine the polling bus from the list
No
Found a bus (A)?
Yes
Retrieve bus (A) data from database
Output
Update time2poll and poll type for bus (A) on the polling list

Figure 4-4 Flow Chart for ACS Emulator

When started, the program initiates the time slot sequence number, time2poll and polling type (as regular polling at initialization) for each active bus and waits for system signal calls. When receiving a trigger signal indicating a message from the inserting poll requester process has been arrived, the requested information (bus ID, time2poll and polling type as inserted polling) is updated to the polling list. When an interrupt signal is caught, the program searches the polling list for a bus candidate that meets the requirements that the bus polling type meets the current channel type, time2poll has expired, and having the highest score, measured as the time elapsed since the expected time2poll, among multiple candidates. If no candidate were found, no poll will be carried out during the current time slot. Otherwise time2poll is reset for a regular poll (2 minutes from the current time) and poll type is reset as no poll for an inserted poll, after the data entry of the latest output from Intersection arrival time prediction module has being successfully retrieved from the database.

4.1.3. Implementing Inserting Poll Requester

The inserting poll requester process generates additional bus polling requests for the ATSP purpose. It uses the output from the ACS emulator and output from traffic and signal status as the primary input to determine the time for inserted polls. Output from the ACS emulator, as aforementioned, is sampled output of intersection arrival time prediction and contains the information about bus ID, route ID, direction, run sequence, next signal ID, next node ID, and times to arrival at next signal and next node. A node is either a bus-stop or an intersection. In case that there is a downstream bus-stop before the next signal, next signal ID differs from next node ID. Output from traffic and signal status includes the current traffic light state (color) on bus approach and the expected switching time to the next state, based on static signal timing table parameters. Combining the outputs from the two modules, one can assess the traffic light state at predicted bus arrival time and thereby the need of the bus for signal priority.

The ATSP operation deploys a check-in check-out mechanism. A check-in call is placed when the bus is approaching a prioritized intersection but not yet arrives at the back of the traffic queue and a check-out call is placed when the bus is cleared from the intersection. The check-in check-out mechanism is not location-based but time-based, with respect to the next signal ID and predicted arrival time. In order to reduce potential negative impact on traffic caused by the uncertainties on bus dwelling time at the nearside bus-stop, check-in calls are initiated after the bus has passed the nearside stop.

Four polling types are designed to support ATSP operation: 1) check-in poll, 2) check-out poll, 3) bus-stop check-out poll and 4) green extension confirming poll. The green extension confirming poll is designed for the green extension TSP treatment to ensure bus need of a requested signal priority.

The inserting poll requester uses the latest received information regarding predicted arrival times at the next intersection and the next node, projected signal phase status on bus approach and priority status at the intersection to dynamically determine which bus and at what time for requesting inserted polls. Only in-service buses that are traveling along the ATSP corridor are eligible for dynamic polling.

Assume a response for the previous inserted polling request arrived at time *t* with predicted arrival time at the next intersection and at the next node as t_A and t_N, respectively. In that case the next node is the intersection, $t_A = t_N$. Otherwise a downstream bus-stop is before the intersection and $t_A > t_N$. The time for requesting an inserted poll with polling type *i* is designed as

$$t_i = \begin{cases} t_A + r_i \times W_i, & i = 1,2,4 \\ t_N + r_i \times W_i, & i = 3 \end{cases}$$

where r_i is the look-ahead look-back parameter ($|r_i| = 1$) and W_i is the predetermined look-ahead/look-back window size associated with polling type *i*. For check-in and green extension conforming polls, $r_i = -1, \quad i = 1,4$, i.e., these two types of polls shall be requested before the predicted arrival time. For check-out and bus-stop check-out polls, $r_i = 1, \quad i = 2,3$.

Figure 4-5 show an example of the timeline for the determination of polling time (time2poll). A response for the previous check-out poll is received at the current time, *t*, with $t_A = t_N$ (no bus-stop check-out polling in this case). The bus may still be far away from the intersection and the predicted arrival time is not likely accurate enough for making an ATSP request. A check-in polling request is scheduled at $t_1 = t_A - W_1$ when the bus is closer to the intersection and the prediction error would become smaller. A returned predicted intersection arrival time for the inserted check-in polling, t'_A, is used for the generation of a TSP request and the schedule for the next check-out polling request at $t_2 = t'_A + W_2$. When the return from the 2nd check-out polling request indicates the bus has passed the intersection, it is then used for the schedule of check-in polling request for the next intersection.

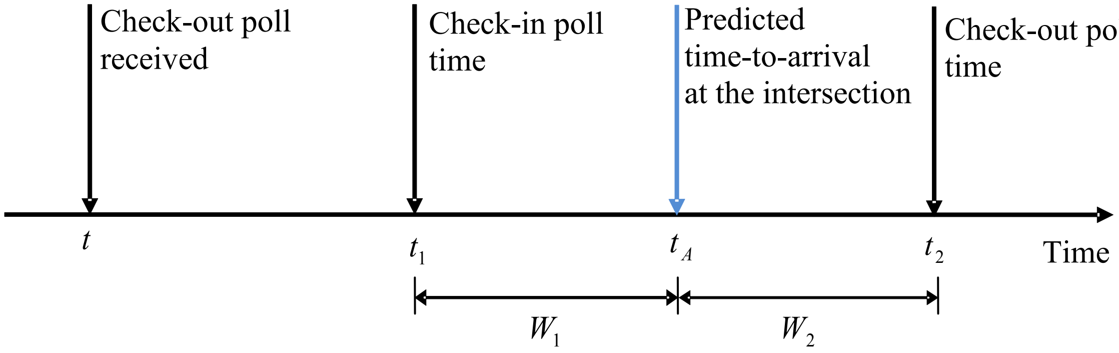


Figure 4-5 Timeline for Dynamic Polling Time

The values for look-ahead/look-back windows were determined from the archived field testing data conducted under the previous ATSP project. The values are presented in Table 4-1.

Table 4-1 Look-Ahead/Look-Back Window Size for Dynamic Polling

Polling Type	ID	*r*	*W* (sec)
Check-in poll	1	-1	30
Check-out poll	2	1	10
Bus-stop check-out poll	3	1	5
Green extension confirming poll	4	-1	5

When determining the polling type and polling time, the projected traffic light status on bus approach is also considered to eliminate redundant polling requests and for the best use of ACS channel capacity. Polling rules related with traffic light status are

- Requesting check-in poll only when the bus is projected to arrive in the red phase;
- Requesting check-out poll when the light turned green; and
- Requesting green extension confirming poll before the signal controller starts executing the treatment (i.e., prior to the yield point).

Depending on the static intersection/bus-stop location configuration and the real-time inputs, the inserting poll request process manages the polling state transitions among the 4 polling types. Figure 4-6 illustrates the state machine diagram for the implemented dynamic polling.

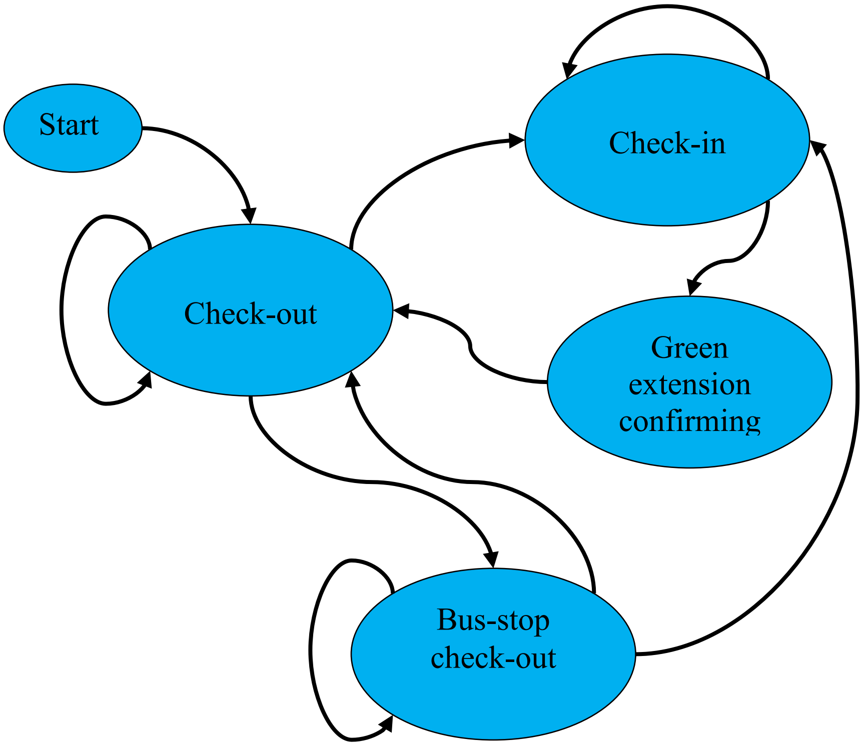


Figure 4-6 State Machine Diagram for Dynamic Polling

At an intersection for a particular bus, multiple polling requests could be made under the same polling type. A minimum polling interval rule is applied such that the time gap between consecutive polls with the same type will be larger than 10 seconds.

4.2. Hardware-in-the-Loop Testing

Before the implementation for a field operational test, two types of tests, hardware-in-the-loop testing and testing with field data, were conducted to evaluate the aforementioned dynamic polling strategies. The objectives were to investigate the contention channel loading and the performance of TSP request generation under dynamic polling. The measure of channel loading is polling latency, defined as the time duration between placement of a polling request to receipt of a polling response. Large polling latency indicates a high probability of request collision.

This subsection presents the hardware-in-the-loop testing and testing with field data is addressed in the next subsection.

A three-intersection arterial was selected for the hardware-in-the-loop testing (see Figure 4-7). Results from the previous ATSP field testing showed that these 3 intersections had relatively high probability of requesting TSP treatments due to high side street traffic and long red for bus phases.

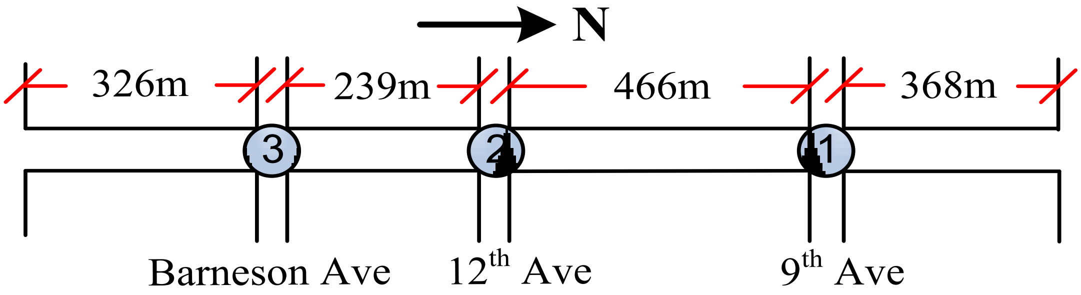


Figure 4-7 Hardware-in-the-Loop Testing Network

Three 170E controllers with the same signal control software deployed in the field were used to control the traffic lights at the three intersections. A simulation program was developed to generate bus trajectories which follow simple kinematic rules and traffic light status. The simulated buses travel back-and-forth along the testing network. Each bus is associated with specified cruising speed and turnaround time at the two ends of the test site. The settings of cruising speed and turnaround time were used to control the number of buses simultaneously running along the testing network.

Two testing scenarios were conducted, scenario #1 with 4 buses simultaneously running and scenario #2 with 8 buses. The design of the testing scenarios considered the characteristics of SamTrans bus routes 390 and 391 that travel along the ATSP corridor. These two routes have a 30 minutes time-headway. The average travel time along the ATSP corridor is around 55 minutes. So on average there are eight buses simultaneously travel within ATSP corridor. As the simulation site is considerably shorter than the ATSP corridor (980 meters vs. 10 miles), simulated buses are more likely to closely follow each other thereby having a higher chance of polling request collisions. Therefore, the channel loading tests under simulation can provide reasonable assessments for channel loading in the field.

Polling latency was first investigated. Figure 4-8 presents the histogram of polling latency for the four-bus scenario. The distribution of the polling latency has a positive skew with a long tail on the positive side. The highest peak near 0.5 seconds indicates the majority of polling requests were responded within 1 second, meaning no request collisions. The 2nd peak near 1.5 seconds indicates there are chances that a polling request was held for the next available contention channel time slot, due to a request collision. However, the probability of holding for more than 1 second is very low.

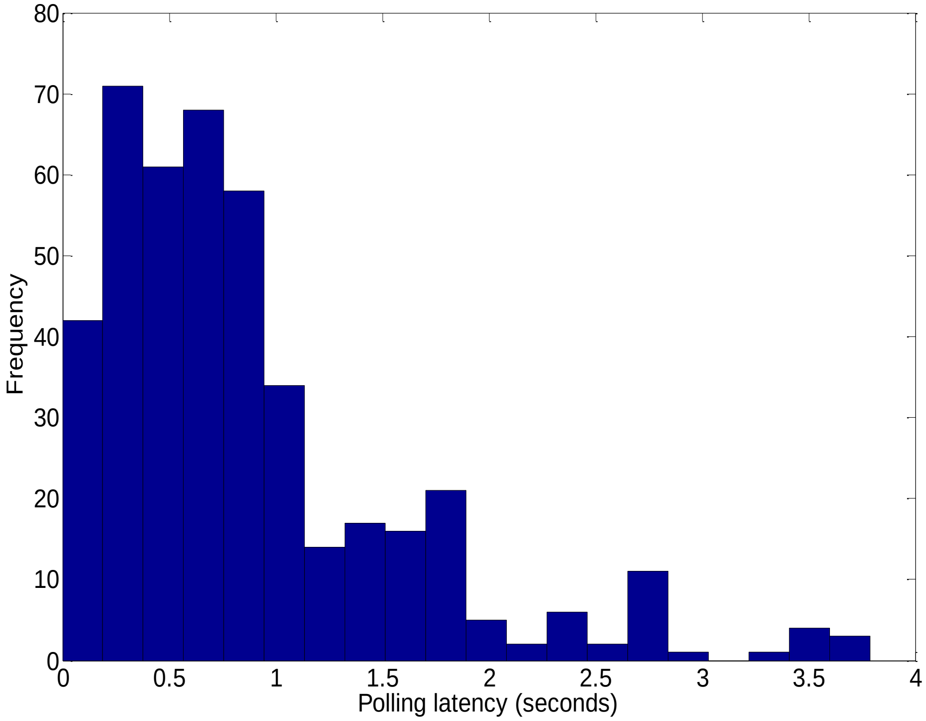


Figure 4-8 Histogram of Polling Latency for Four-Bus Scenario

Figure 4-9 shows the distribution of polling latency for the eight-bus scenario. It also has a positive skew distribution. As expected, the chance of request collisions for the eight-bus scenario is higher than that for the four-bus scenario. Still, the probability of holding polling requests for more than 2 seconds is low.

It is worth pointing out that the collision rate of polling requests under the simulation setup is larger than that in the field, as simulated buses are closely following each other thereby having concentrated polling requests from different buses.

Priority request, simulated bus trajectories and signal control status data were recorded and analyzed. ATSP requesting under dynamic polling worked as expected. No noticeable evidence was found to indicate there would be a significant downgrade of ATSP operation under dynamic polling.

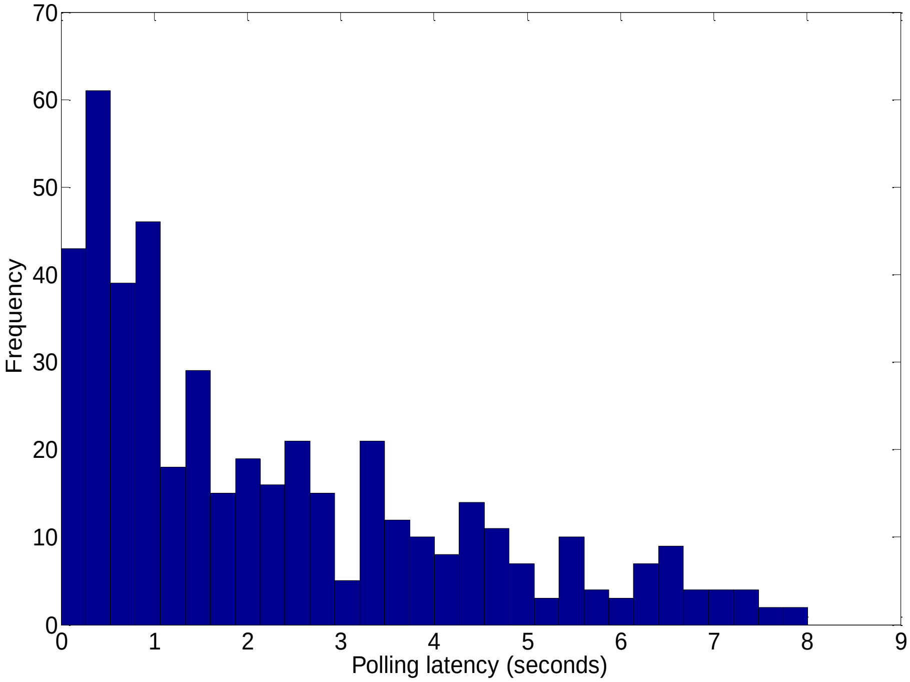


Figure 4-9 Histogram of Polling Latency for Eight-Bus Scenario

4.3. Testing with Field Data

Although the hardware-in-the-loop testing gave a green light for the integration of ATSP/DPI and AVL/ACS with dynamic polling, the simulation environment is relatively simple. In order to investigate the potential impacts of real-world factors, tests were conducted with field data. ACS emulation and dynamic polling were implemented in the field. When conducting tests with real-world data, data flow and data exchange between subsystems and processing modules were in a real-time manner, except the generated ATSP requests were not sent to the local signal controllers.

This subsection presents an example of testing with field data. More detailed performance of dynamic polling is presented in the next Section.

Figure 4-10 shows a sample route 390 southbound bus trip. The trip length is 26.2 miles and the trip duration was 126 minutes from end to end. Bus travel time within the 10-mile-long ATSP corridor was 54 minutes and the bus stopped at 13 out of the 50 prioritized intersections. There were a total of 51 regular polls for the entire trip. Of those, 20 were within the ATSP corridor. The blue line in Figure 4-10 is bus trajectory. The two red horizontal lines are the locations of two ends of the ATSP corridor. The circles along bus trajectory are the locations of regular bus polls. The 20 bus location updates by regular polling did not catch the 13 stopped at traffic light events. It shows the limitation of generating priority requests based on regular ACS bus polling.

Figure 4-11 plots the same bus trip (enlarged within ATSP corridor) with the information regarding both regular and inserted ACS polls. The horizontal red lines are the locations of signalized intersections. The green circles are locations for regular bus polls, the same as shown in Figure 4-10. The red crosses are locations for dynamic inserted polls. There were a total of 121 inserted polls were requested and responded. As shown, inserted polls were not evenly distributed but varied with the changes of bus movement and distance to the signalized intersections. ATSP requests were generated at all 13 stopped prioritized intersections.

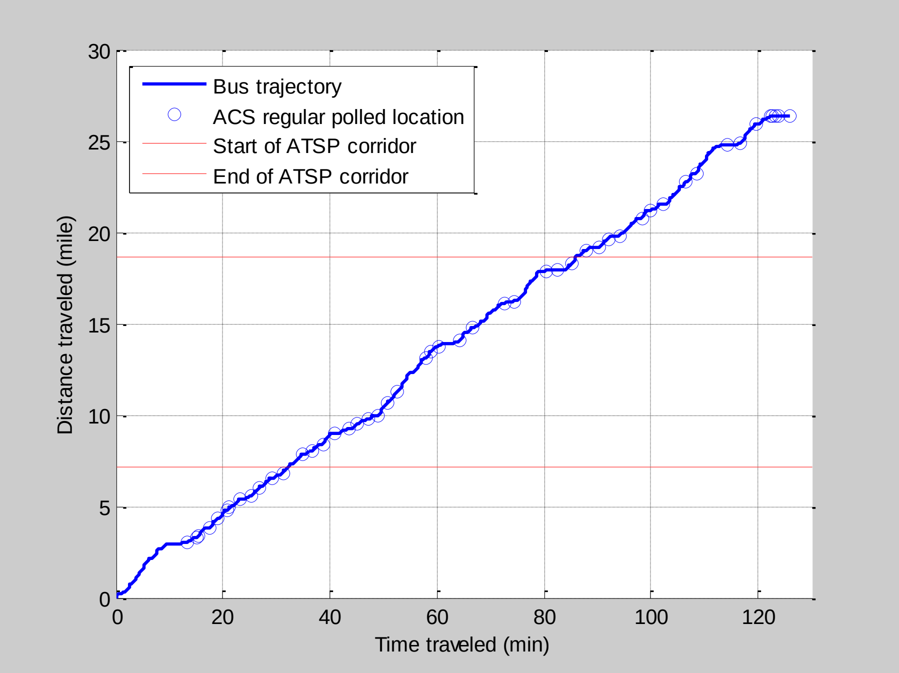


Figure 4-10 A Sample Southbound 390 Bus Trip with Regular ACS Polls

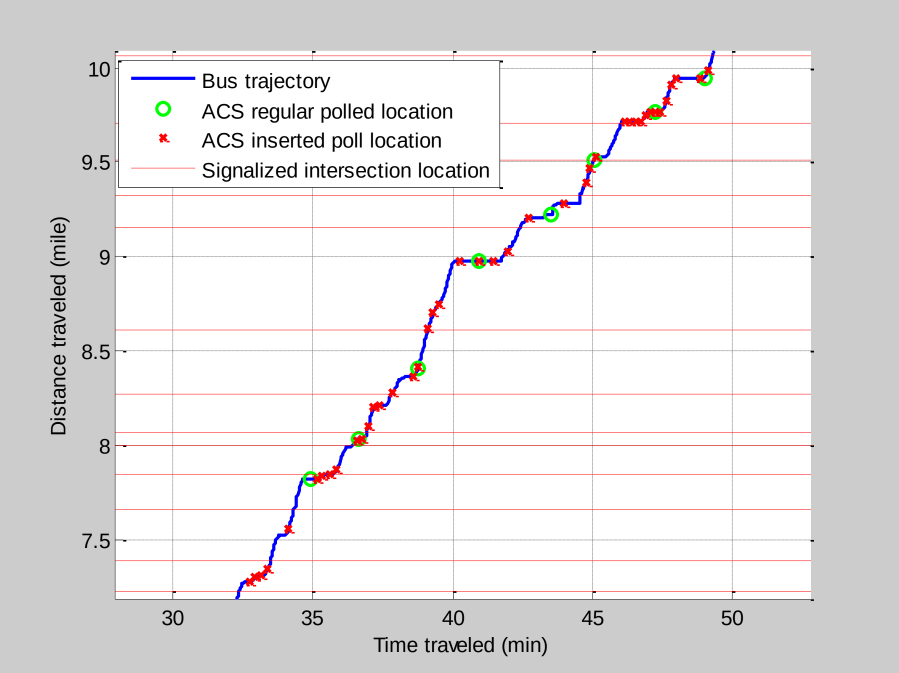


Figure 4-11 A Sample Southbound 390 Bus Trip with both Regular and Inserted ACS Polls

Some more bus trips were analyzed case by case. In conclusion, the testing with field data agrees with the hardware-in-the-loop simulation test in that the proposed dynamic polling strategies can support the integration approach.

4.4. Sensitivity Analysis

With dynamic polling, the time for generating ATSP requests is influenced by the polling strategies and the ACS channel capacity. A sensitivity analysis was conducted to investigate the impacts on ATSP performance due to the change from using second-by-second GPS data to using AVL/ACS dynamic polling. For the green extension treatment, the impact should be small as the polling strategies and ATSP requesting strategies are designed in the way that a bus will receive multiple inserted polls before it checks out from the intersection to reduce the miss of potential green extension treatment. Moreover, the predicted arrival time is likely accurate as there is no interaction with traffic queues. In contrast, the early green treatment would receive larger impacts as the magnitude of borrowed time from non-priority phases depends on the predicted arrival time and the time the ATSP request is generated. With second-by-second bus location updates, a priority request can be adjusted in response to the changes in a predicted arrival time. However, with dynamic polling, priority request is likely a one-shot case.

In order to assess the impacts of dynamic polling on ATSP performance, requesting time for early green treatments were studied using the previous field testing data. The majority of early green requests were made with the predicted time-to-travel (T2G) to the intersection falling between 20 seconds and 25 seconds. The changes in prediction errors made at T2G = 25 seconds and T2G = 20 seconds were then investigated to see how much improvement can be gained by priority request adjustment.

Figure 4-12 and Figure 4-13 shows the distribution of prediction errors at these two points (T2G-25 and T2G-20) for northbound and southbound, respectively. As shown, the shapes of the distributions are very thin with standard deviations as short as 1.37 and 1.67 seconds, respectively. The one sigma window from the mean of the distribution (μ±σ) covers 78.3% of the whole distribution for northbound and 83.9% for southbound. In other words, there is an 80% chance that the influence of prediction error due to dynamic polling is within ±1.5 seconds. The incurred increment or decrement of bus intersection delay is within 1.5 seconds and the incurred changes on each traffic movement will be less than 1 second. Therefore the designed dynamic polling strategies are able to support ATSP operation without significant compromise the proposed discrete communication protocol is able to support the ATSP system without significant compromise on ATSP performance.

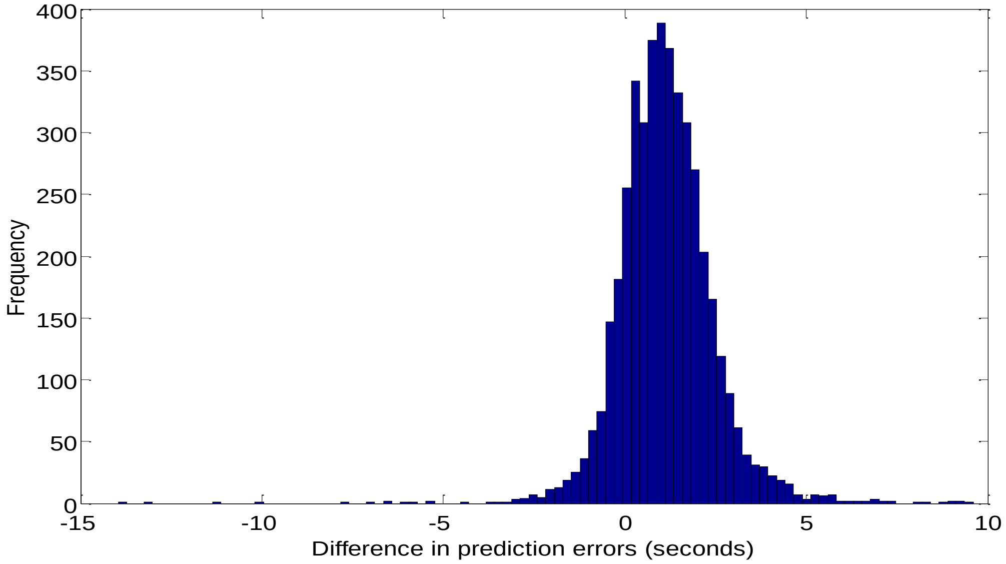


Figure 4-12 Histogram of Difference in Prediction Errors (Northbound) (between T2G-25 and T2G-20)

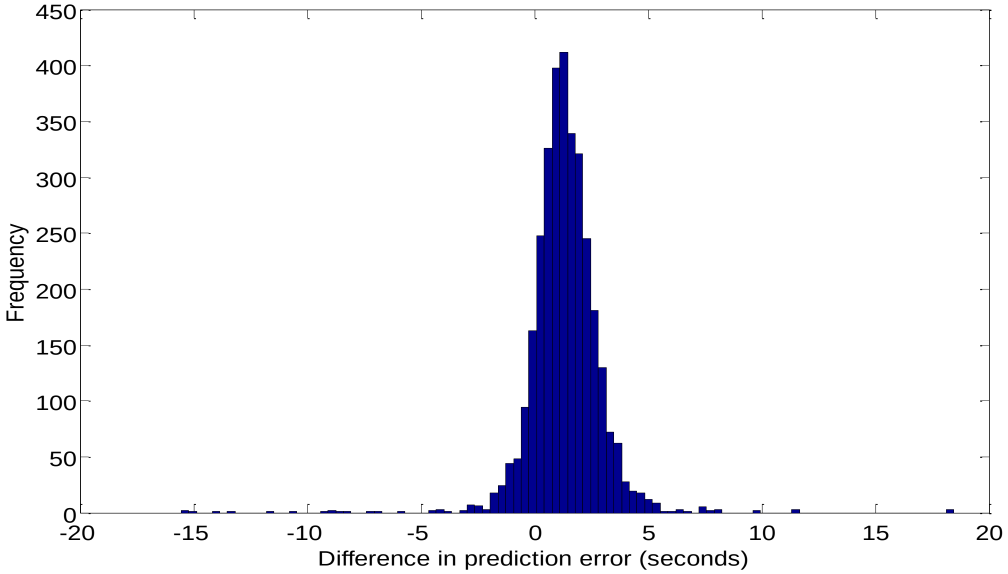


Figure 4-13 Histogram of Difference in Prediction Errors (Southbound) (between T2G-25 and T2G-20)

5. Field Operational Testing Results

As described in Section 4.1.1, the field operational test (FOT) was conducted using the ACS emulation approach, due to the lack of access to the existing AVL/ACS' onboard and central processing software. The objectives of the field operational test were to evaluate the capability of AVL/ACS in supporting the integrated ATSP/DPI system with the developed dynamic polling technology and the resulted benefits from the integration approach.

Section 5.1 provides a description of the field operational test. Testing results, including the statistics of dynamic polling, ATSP savings, and benefits of integrated ATSP/DPI on the generation of dynamic information, are summarized in Section 5.2.

5.1. Description of the Field Operational Test

The field operational test was carried out for two SamTrans bus routes 390 and 391 along El Camino Real (ECR). The ATSP corridor is the central section of routes 390 and 391. It includes 50 signalized intersections which are under 7 field master controllers. Figure 5-1 shows a map overlay of the field testing corridor. The "pin" icons on the map are the origin-destination locations of bus service patterns as described in Section 5.1.1 (see Table 5-1).

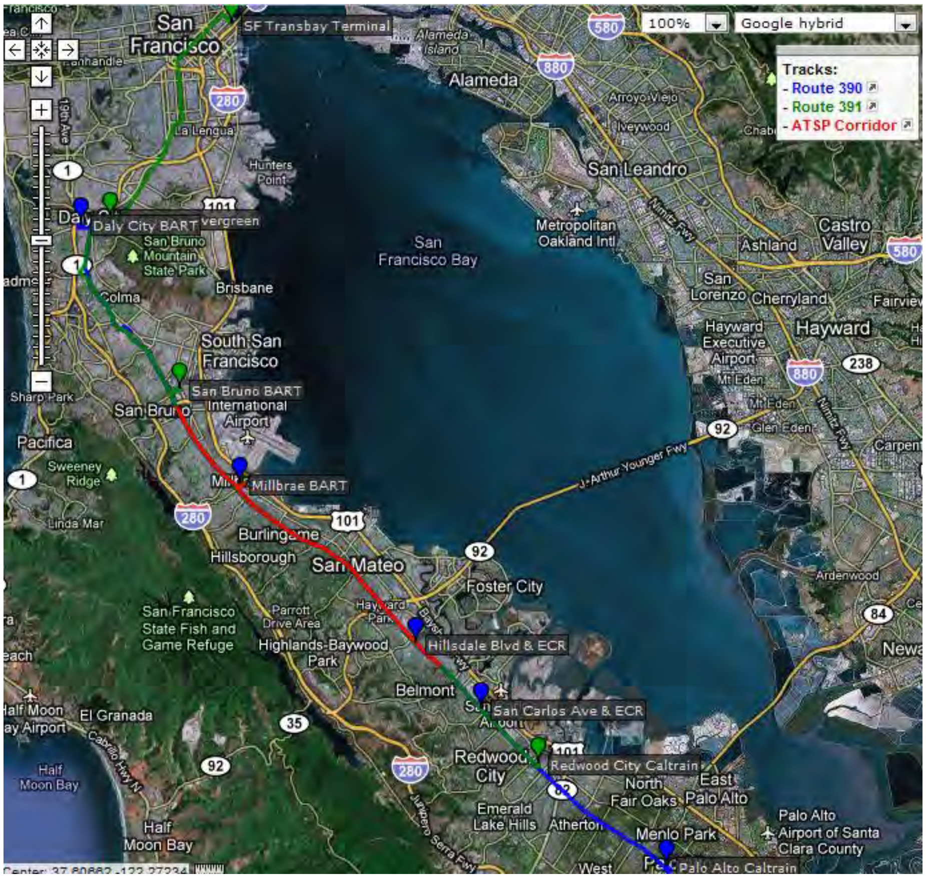
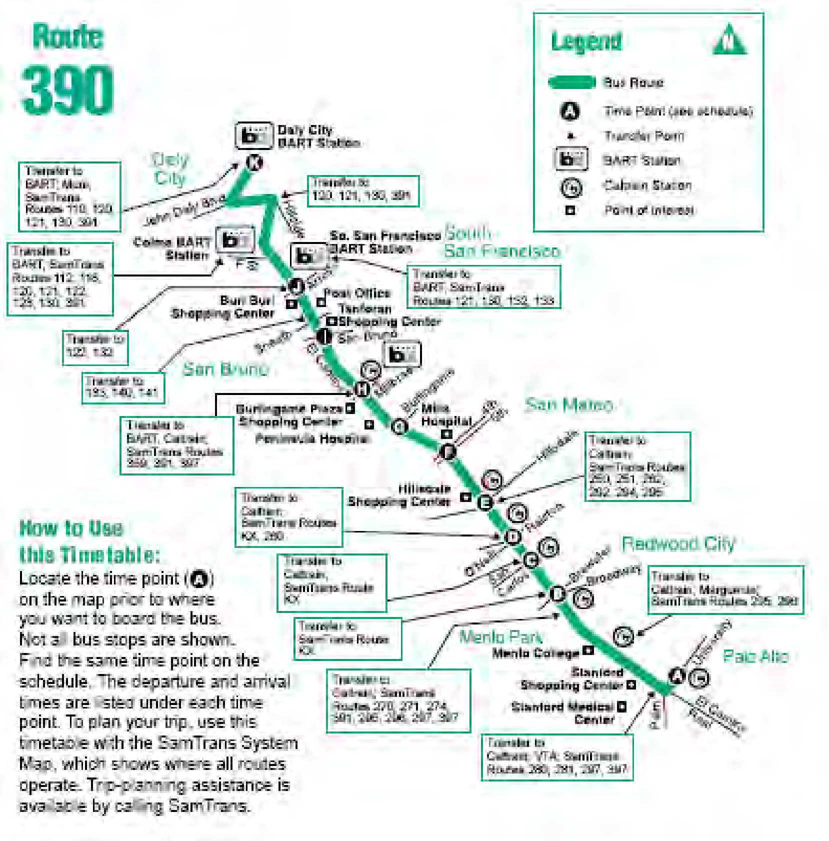


Figure 5-1 Field Operational Testing Site

5.1.1. **Testing Bus Routes**

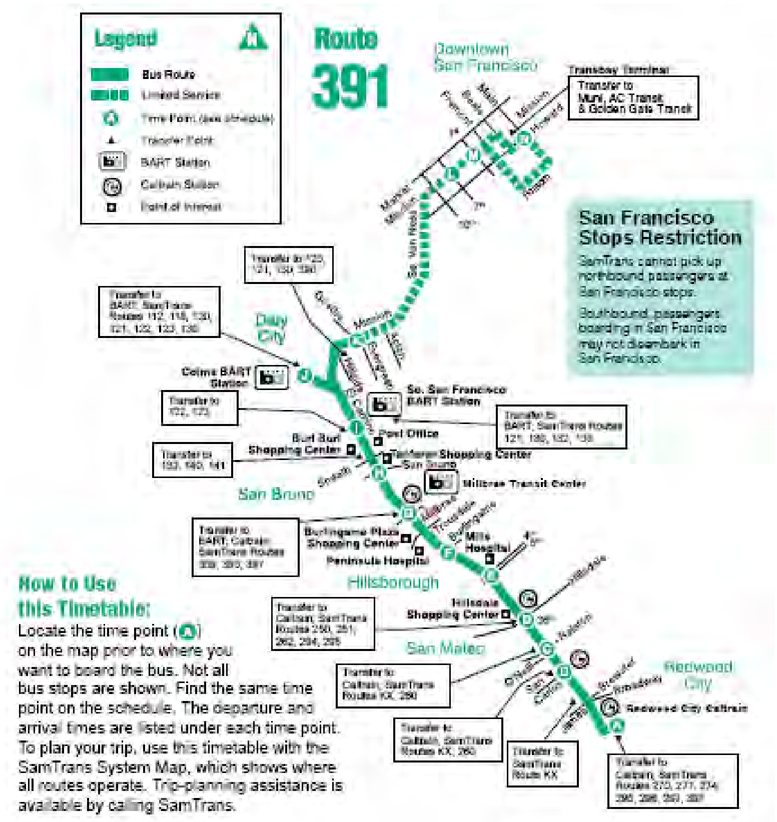
Routes 390 and 391 are the backbone of the SamTrans forty-eight fixed routes bus network and provide intercity transit bus service along El Camino Real. Together they account for one quarter of the total ridership.

Bus route 390 serves Daly City BART, Colma, South San Francisco, San Bruno, Millbrae, Burlingame, San Mateo, Belmont, San Carlos, Redwood City, Atherton, Menlo Park, and Palo Alto. Bus route 391 serves San Francisco, Daly City, Colma BART, South San Francisco, San Bruno, Millbrae, Burlingame, San Mateo, San Carlos and Redwood City. Figure 5-2 and Figure 5-3 show the route map for bus route 390 and 391, respectively. These two bus routes run with half-hour headways during the peak hours. They connect five BART stations (Daly City, Colma, South San Francisco, San Bruno and Millbrae) and nine Caltrain stations (Millbrae, Broadway, Hillsdale, Belmont, San Carlos, Redwood City, Atherton, Menlo Park and Palo Alto).


Route
390
Legend
Bus Route
Time Point (see schedule)
Transfer Point
BART Station
Caltrain Station
Point of Interest
Daly City BART Station
Daly City
Transfer to BART; Muni, SamTrans Routes 110, 120, 121, 130, 391
John Daly Blvd
Transfer to 120, 121, 130, 391
Hillside
Colma BART Station
F St
Transfer to BART, SamTrans Routes 112, 116, 120, 121, 122, 129, 130, 391
So. San Francisco BART Station
South San Francisco
Transfer to BART, SamTrans Routes 121, 130, 132, 133
Post Office
Buri Buri Shopping Center
Tanforan Shopping Center
San Bruno
Transfer to 122, 132
Transfer to 133, 140, 141
San Bruno
El Camino
Millbrae
Burlingame
Burlingame Plaza Shopping Center
Peninsula Hospital
Mills Hospital
Transfer to BART, Caltrain, SamTrans Routes 359, 391, 397
San Mateo
Hillsdale
Transfer to Caltrain; SamTrans Routes 250, 251, 260, 292, 294, 295
Hillsdale Shopping Center
Transfer to Caltrain; SamTrans Routes KX, 260
Ralston
Redwood City
Transfer to Caltrain, SamTrans Route KX
O'Neill
San Carlos
Brewster
Broadway
Transfer to Caltrain, Marguerite; SamTrans Routes 295, 296
Transfer to SamTrans Route KX
Menlo Park
Menlo College
Transfer to Caltrain; SamTrans Routes 270, 271, 274, 291, 295, 296, 297, 397
Stanford Shopping Center
Stanford Medical Center
University
Palo Alto
El Camino Real
Transfer to Caltrain; VTA; SamTrans Routes 280, 281, 297, 397
A
B
C
D
E
F
G
H
I
J
K

How to Use this Timetable:
Locate the time point (A) on the map prior to where you want to board the bus. Not all bus stops are shown. Find the same time point on the schedule. The departure and arrival times are listed under each time point. To plan your trip, use this timetable with the SamTrans System Map, which shows where all routes operate. Trip-planning assistance is available by calling SamTrans.

Figure 5-2 SamTrans Route 390 Map (from www.SamTrans.com)



San Francisco Stops Restriction

SamTrans cannot pick up northbound passengers at San Francisco stops.

Southbound passengers boarding in San Francisco may not disembark in San Francisco.

How to Use this Timetable:

Locate the time point (Ⓐ) on the map prior to where you want to board the bus. Not all bus stops are shown. Find the same time point on the schedule. The departure and arrival times are listed under each time point. To plan your trip, use this timetable with the SamTrans System Map, which shows where all routes operate. Trip-planning assistance is available by calling SamTrans.

Figure 5-3 SamTrans Route 391 Map
(from www.SamTrans.com)

Bus routes 390 and 391 provide multiple schedule-based services (patterns) during a typical day. The service pattern is defined by the trip origin, destination, and bus-stops and time-points associated with it. The time-points are bus stops that have their schedule published for the public. Table 5-1 summarizes service patterns for bus routes 390 and 391.

Table 5-1 Routes 390 and 391 Service Patterns

Route	Pattern No.	Length (Miles)	Direction	Origin	Destination	No. of Stops / Time-Points
390	1	7.1	Southbound	San Carlos Ave & ECR	Palo Alto Transit Center	26 / 3
	2	16.0	Southbound	Daly City BART	Hillsdale Blvd & ECR	63 / 7
	3	26.2	Southbound	Daly City BART	Palo Alto Transit Center	99 / 11
	4	17.2	Southbound	Millbrae Transit Center	Palo Alto Transit Center	64 / 8
	5	26.2	Northbound	Palo Alto Transit Center	Daly City BART	98 / 11
	6	16.0	Northbound	Hillsdale Blvd & ECR	Daly City BART	63 / 7
391	1	18.5	Southbound	SF Transbay Terminal	Millbrae BART	58 / 8
	2	23.2	Southbound	Mission & Evergreen	Redwood City Caltrain	85 / 11
	3	16.5	Southbound	San Bruno BART	Redwood City Caltrain	62 / 8
	4	30.8	Southbound	SF Transbay Terminal	Redwood City Caltrain	104 /14
	5	30.8	Northbound	Redwood City Caltrain	SF Transbay Terminal	107 / 14
	6	23.2	Northbound	Redwood City Caltrain	Mission & Evergreen	84 / 11
	7	18.5	Northbound	Millbrae Transit Center	SF Transbay Terminal	61 / 8
	8	25.3	Northbound	Hillsdale Blvd & ECR	SF Transbay Terminal	88 / 11

5.1.2. **Testing ATSP Corridor**

The ATSP corridor is the central section of bus routes 390 and 391, along El Camino Real. The corridor is 10 miles long and covers 50 signalized intersections in the cities of San Mateo, Burlingame, Millbrae and San Bruno. The northern end of the ATSP corridor is at the intersection of 42nd Ave at El Camino Real in the city of San Mateo and the southern end is at the intersection of Bayhill Dr at El Camino Real in the city of San Bruno. The fifty signalized intersections (see Table 5-2) are grouped and controlled by 7 traffic control systems.

Except bus route 390 service pattern #1 that runs from San Carlos Ave & ECR to the Palo Alto Transit Center, all other service patterns as described in Table 5-1 cover either the whole or partial ATSP corridor.

Table 5-2 Lists of Signalized Intersections for Field Operational Test

Signalized Intersection	City	Field Master Number
42nd Ave	San Mateo	FM #1
41st Ave		
37th Ave		
31st Ave		
28th Ave	San Mateo	FM #2
27th Ave		
25th Ave		
20th Ave		
17th Ave		
Barneson Ave		
12th Ave		
9th Ave		
5th Ave	San Mateo	FM #3
4th Ave		
3rd Ave		
2nd Ave		
Crystal Springs Rd		
Baldwin/Baywood Ave		
Tilton/El Cerrito Ave		
Poplar Ave		
Bellevue Ave		
Peninsula Ave/Park Rd	Burlingame	FM #4
Primrose/Bayswater/Cypress		
Howard Ave		
Burlingame Ave		
Chapin Ave		
Floribunda Ave		
Oak Grove Ave		
Sanchez Ave		
Carmelita Ave		
Broadway		
Lincoln Ave		
Hillside Dr		
Adeline/Oxford/Cambridge		
Ray/Rosedale		

Signalized Intersection	City	Field Master Number
Trousdale Dr	Millbrae	FM #5
Murchison Dr		
Millbrae Ave		
Hillcrest Blvd		
Silva Ave		
Meadow Glen Ave	San Bruno	FM #6
Center St		
Park Place/Santa Inez		
San Felipe Ave		
Crystal Sprints Rd		
Taylor Ave/San Mateo Ave	San Bruno	FM #7
Jenevein Ave		
W. Angus Ave		
San Bruno Ave		
Bayhill Dr		

5.1.3. Testing Vehicles

In order to support the field operational test of the integrated ATSP/DPI system with ACS emulation, fifteen SamTrans buses that provide service on routes 390 and 391 were instrumented with the cell-phone-based AVL system (Figure 5-4) to receive and send second-by-second GPS data as one set of inputs to the ACS emulation process. The other set of inputs to the ACS emulation process is the real-time bus reporting data feeds from the existing AVL/ACS.

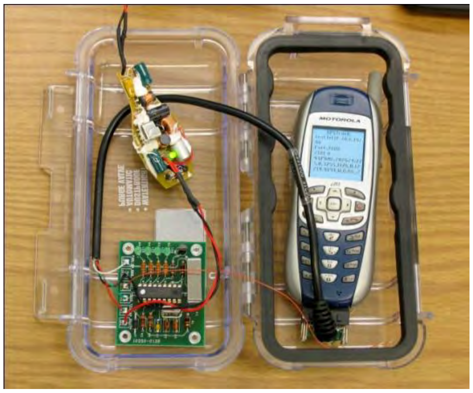


Figure 5-4 Cell-Phone based AVL System

5.1.4. Testing Periods

The field operational test was conducted in two stages: 2-week "silent" field test and 8-week formal field operational test. In addition, 4 weeks "before" data – the scenario of without ATSP, were collected for the evaluation purpose. The "after" data for the scenario of with ATSP were collected during the 8-week field operation test. Table 5-3 summarizes the testing and data collection periods.

Table 5-3 Filed Testing Periods

	Start Date	End Date	Duration
"Before" Data Collection	12/20/2010	01/14/2011	4 weeks
Silent Test	01/03/2011	01/21/2011	2 weeks
Field Operational Test and "After" Data Collection	01/24/2011	03/18/2011	8 weeks

During the "silent" testing period the integrated ATSP/DPI system performed exactly the same as it would be under for field operational test but without sending priority requests to the signal controllers. The objectives of the "silent" test was 1) to test the generations of ACS dynamic polling requests and ATSP requests, and 2) to prepare the integrated system being ready for the formal field operational test.

The average number of ATSP requests per bus one-way trip achieved during the "silent" testing period was compared with that from the previous ATSP filed test conducted in the year 2008. The previous field testing site covers the northern portion of the current fielding test sites with 35 signalized intersections. Second-by-second bus location updates were used for the previous field test while discrete samples induced by dynamic polling were used for the "silent" test.

Table 5-4 summarizes the comparison results for northbound and southbound bus trips. The number of ATSP requests per bus one-way trip under dynamic polling is very consistent with that using second-by-second bus location updates. This indicates that the integrated system does not downgrade the ATSP performance with the use of discrete AVL/ACS feeds compared with the use of second-by-second GPS data.

Table 5-4 Comparison of Number of ATSP Requests per Bus One-Way Trip

Number of Signalized Intersections	Number of ATSP Requests (Northbound)			Number of ATSP Requests (Southbound)		
	Early Green	Green Extension	Total	Early Green	Green Extension	Total
50 (current FOT)	8.0	3.9	11.9	8.1	4.0	12.1
35 (previous FOT)	6.5	3.5	10.0	4.7	2.8	7.5

5.1.5. Sample Size

Pre-processing was applied to filter out bus trips that have large GPS location errors and pick trips were made between 7AM to 7PM, when the traffic signal control systems were providing coordinated control. Table 5-5 summarizes the bus trip sample size used for the evaluation study.

Table 5-5 Bus Trip Sample Size

Scenario	Route 390		Route 391		Total	
	Southbound	Northbound	Southbound	Northbound	Southbound	Northbound
Without ATSP	120	168	137	131	257	299
With ATSP	276	283	238	197	514	480

5.2. Field Testing Results

This subsection summarizes the field testing results regarding the performance of emulated dynamic polling, performance of ATSP with dynamic polling and the performance of prediction of arrival times at bus-stops.

5.2.1. Performance of Emulated Dynamic Polling

The performance of the existing ACS regular bus polling is shown in Figure 5-5. Samples are from routes 390 and 391 in-service buses. The distributions of the regular polling rate for southbound and northbound are almost identical. About 75% of time the polling rate is around 2 minutes. The second highest peak at 4 minutes approximately accounts for 15%. This happened when a regular polling failed in receiving a bus location update but succeeded in the following polling cycle. The polling rate can be larger than 4 minutes while the chance is low (<7%). This verifies that the protocols used for an ACS emulator are appropriate.

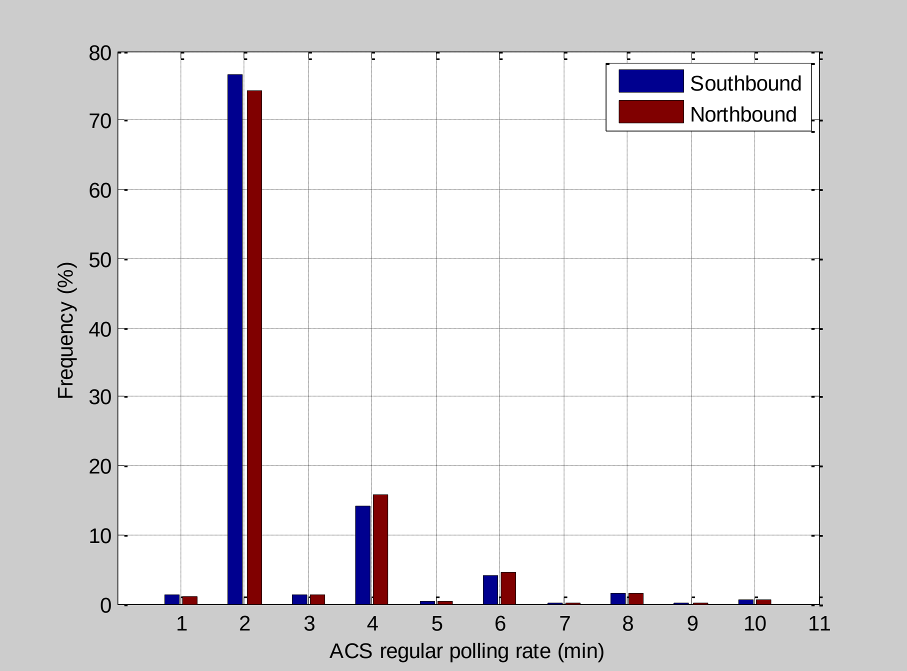


Figure 5-5 Distribution of ACS Regular Polling Rate for Bus Routes 390 and 391

The performance of dynamic polling was investigated in terms of the number of inserted polling requests per bus one-way trip, polling latency and the requested polling type.

Figure 5-6 depicts the distributions of the number of inserted polls per bus trip for southbound and northbound. The distribution has formed a bell shape curve for each direction. The mean and median values for southbound trips are 119 and 120, respectively, and that for northbound trips are 107 and 108, respectively. The number of inserted polls on southbound trips is slightly higher than that on northbound trips due to longer trip time along the ATSP corridor. The average trip time for a southbound trip is 54.6 minutes, about 1 minute longer than for northbound trips. The average time gap between consecutive inserted polls for individual buses is 26 seconds for southbound and 27 seconds for northbound.

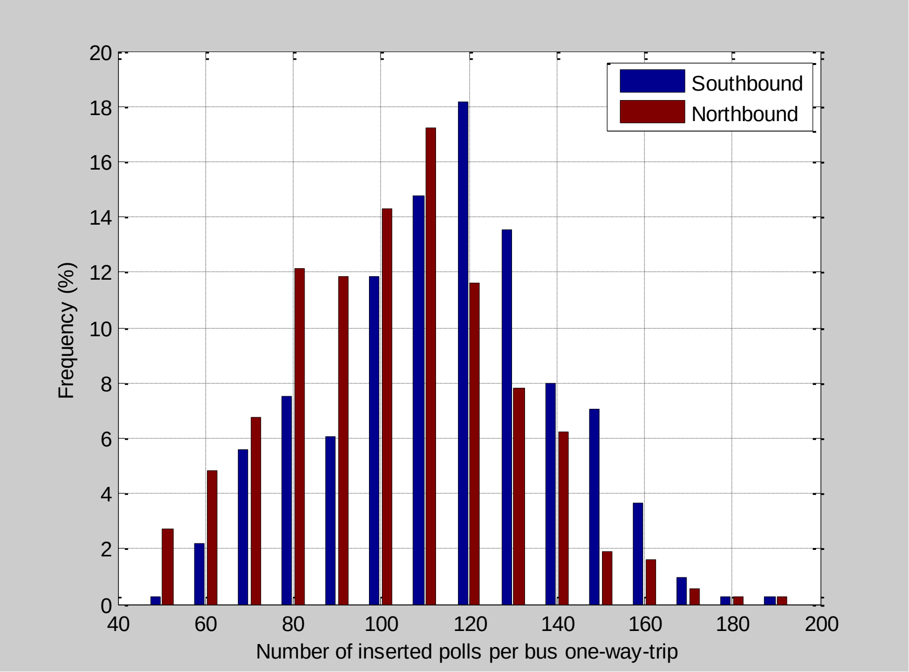


Figure 5-6 Distribution of Number of Inserted ACS Polls per Bus One-Way Trip

The frequency of inserted polls is about 3 times higher than that for regular polls. Polling latency was analyzed to check the load on the contention channel. When the communication channel got overloaded, responses to polling requests would be delayed leading to large polling latency. The distribution of polling latency is shown in Figure 5-7. About 97% of the time, a polling request was responded to within 1 second, meaning there were no multiple requests that compete for the same channel time slot. Request collision did happen though the chance is very low and the longest holding time due to a request collision is about 4 seconds. The cause of the results is strongly related with the characteristics of routes 390 and 391. With a 30-minute time-headway for both routes, there are at most 8 in-service buses simultaneously traveling along the ATSP corridor. With the contention channel providing two time slots in every 1 second interval, the polling request is not likely to be delayed more than 4 seconds.

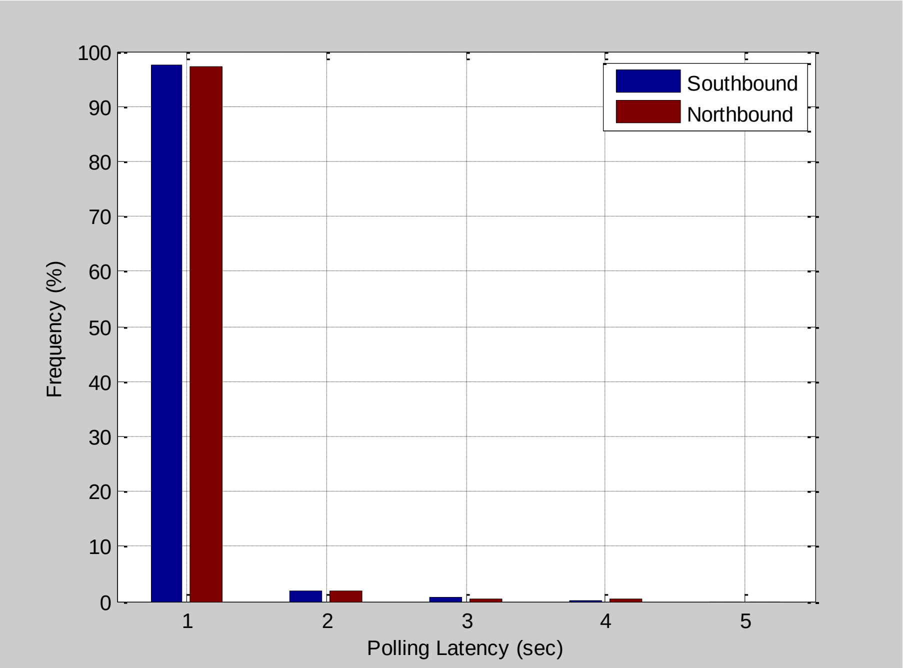


Figure 5-7 Distribution of Polling Latency

The major objective of dynamic polling is to enable ATSP functionality with AVL/ACS, by dynamically inserting check-in check-out polls. Table 5-6 presents the distribution of an inserted polling type. As expected, the green confirming poll has the lowest frequency as it should only be requested when a green extension treatment is need by the bus. Check-out poll has the highest frequency since the response from a check-out poll is used to determine the check-in polling time for the next intersection, therefore more than one check-out polling request could be made to ensure the bus is cleared from the intersection. The frequency of bus-stop check-out poll is compatible with TSP check-in poll as routes 390 and 391 provide local bus service and share 43 bus-stops within the 50-signal ATSP corridor.

Table 5-6 Distribution of Inserted Polling Type

	Check-In Poll	Check-Out Poll	Bus-Stop Check-Out Poll	Green Extension Confirming Poll
Southbound	29.5%	43.2%	25.2%	2.1%
Northbound	31.0%	44.6%	21.9%	2.5%

The results indicate that dynamic polling with ACS emulation performed as expected. Inserted polling requests were processed and responded in a timely manner and the implemented dynamic polling strategies did not overwhelm the existing capacity of the radio communication system.

The immediate question is how the ATSP performed with dynamic polling. This is addressed in the next subsection.

5.2.2. **Performance of ATSP under Emulated Dynamic Polling**

Before-and-after analysis was conducted to evaluate ATSP impacts on transit operations. The measures of effectiveness (MOEs) used for the evaluation are those adopted and most commonly applied in TSP evaluation studies, which include bus trip travel time, total intersection delay, number of stops at prioritized intersections, average running speed, average intersection delay per stopped intersection, and schedule deviation at time-points. Statistic t-test was performed for each MOE to test whether the change of mean value between before-and-after scenarios is statistically significant at the 5% level or not. The evaluation results are summarized in Table 5-7.

Table 5-7 Summary of ATSP Impacts on Transit Operations

MOE	Direction	Mean Value		Change in Mean		p-value of t-test	Statistically Significant at 5% level?
		Without ATSP	With ATSP	Value	%		
Trip Travel Time (minutes)	Southbound	55.3	52.9	-2.4	-4.4%	4.1E-7	✓
	Northbound	53.5	52.8	-0.8	-1.4%	0.022	✓
Total Intersection Delay (seconds)	Southbound	327.9	264.3	-63.6	-19.4%	1.2E-7	✓
	Northbound	362.2	328.8	-33.6	-9.2%	1.7E-4	✓
Number of Stops at Red	Southbound	11.0	10.4	-0.6	-5.9%	0.036	✓
	Northbound	12.6	12.1	-0.5	-4.1%	0.027	✓
Running Speed (MPH)	Southbound	18.7	19.5	0.8	4.3%	7.2E-6	✓
	Northbound	19.7	20.0	0.3	1.5%	0.048	✓
Average Intersection Delay per Stopped Intersection (seconds)	Southbound	30.0	25.5	-4.2	-14.0%	1.1E-6	✓
	Northbound	28.8	27.3	-1.5	-4.9%	0.009	✓
Schedule Deviation at Time-Points (minutes)	Southbound	2.3	1.8	-0.5	-22.9%	0.013	✓
	Northbound	3.4	2.8	-0.6	-17.4%	1.3E-4	✓

The following observations can be made from Table 5-7.

- ATSP saved bus travel time by 4% (145 seconds) for southbound trips and 1% (46 seconds) for northbound trips. Trip travel time savings are statistically significant at the 5% level;
- ATSP reduced bus total intersection delay by 19% (64 seconds) for southbound and by 9% (34 seconds) for northbound. Moreover, the average intersection delay per stopped intersection was reduced by 14% (4 seconds) for southbound and by 5% (2 seconds) for northbound. The changes are statistically significant at the 5% level;
- ATSP increased bus average running by 4% (0.8 MPH) for southbound and by 2% (0.3 MPH) for northbound. The changes are statistically significant at the 5% level;
- ATSP operations reduced number of stops at prioritized intersections by 6% for southbound and 4% for northbound. The changes are statistically significant at the 5% level; and
- ATSP reduced schedule deviation at time-points (positive value indicating the bus is behind the schedule) by 23% (32 seconds) for southbound and by 17% (36 seconds) for northbound. The changes are statistically significant at the 5% level. The percentage of

the buses running over 5 minutes late was reduced by 27% (from 20% before to 15% after) for southbound and by 13% (from 24% to 21%) for northbound.

Routes 390 and 391 provide schedule-based bus service. ATSP reduced bus intersection delay and increased running speed leading buses to arrive earlier at the time-points. As a result, sometime bus operators waited longer at time-points in order to meet the on-time performance requirement. This time-point holding due to ATSP phenomenon can be observed from Figure 5-8 and Figure 5-9.

Figure 5-8 plots the empirical cumulative distribution function (CDF) of average dwelling time per time-point per trip at 4 southbound time-points along the ATSP corridor. The red curve is for the "without ATSP" scenario and blue curve for "with ATSP" scenario. The mean is 43 seconds for "without ATSP" and 53 seconds for "with ATSP". The percentage change is 22.5%. The p-value of statistic t-test is 0.013, indicating the change is statistically significant at the 5% level.

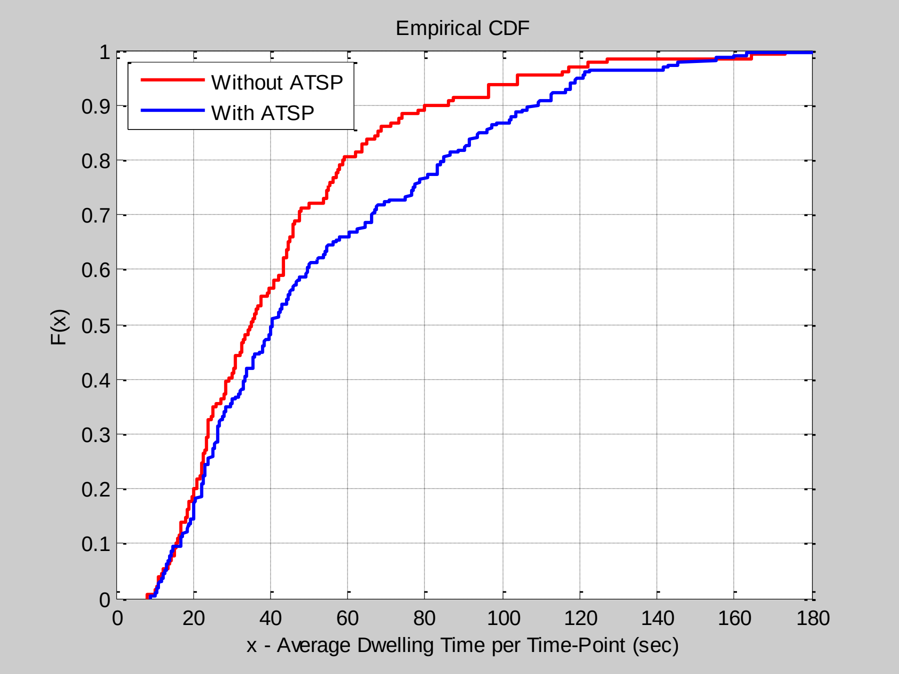


Figure 5-8 Comparison of Average Dwelling Time per Time-Point (Southbound)

Northbound trips experienced similar behavior (see Figure 5-9, also with 4 time-points). The mean is 50 seconds for "without ATSP" and 55 seconds for "with ATSP" (11% change). The change in mean is also statistically significant (p-value = 0.027).

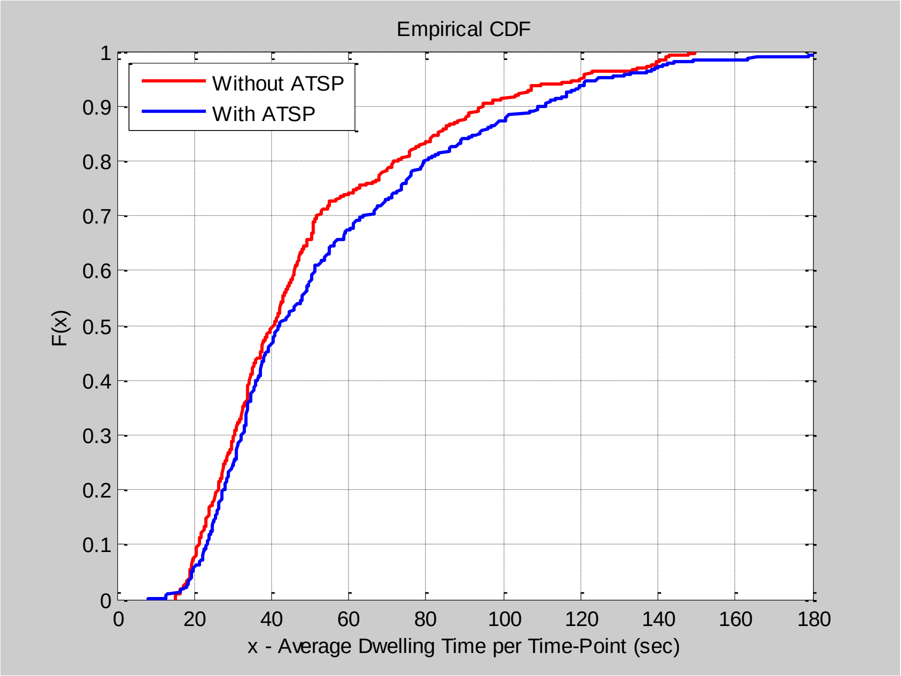


Figure 5-9 Comparison of Average Dwelling Time per Time-Point (Northbound)

If excluding this side impact of ATSP, bus trip travel time (excluding dwelling time at 4 time-points for both "with" and "without" cases) was reduced 5.6% for southbound direction and 2.4% for northbound travel.

Additional empirical CDF plots for before-and-after comparisons of MOEs listed in Table 5-7 are shown in Appendix A.

5.2.3. **Performance of Prediction of Arrival Time at Bus-Stop**

The prediction module keeps updating predicted arrival time at bus-stops as the bus is traveling along the route. Therefore, the prediction error is associated with the time the predicted result was made. The performance of bus-stop arrival time prediction was analyzed at two levels. At the first level, predicted results contributed by inserted polls were filtered out to evaluate how the prediction performed when only using regular ACS polling data at 2-minute polling rate as real-time inputs. At the second level, all polling data were used to study how inserted polls originated for ATSP purposes would help to improve the prediction performance.

Millbrae Station in the northbound direction was selected for the level 1 analysis as it has an in-station kiosk display and the southbound Station at Millbrae is much closer to the northern end of the ATSP corridor than the northbound Station to the southern end. Figure 5-10 depicts the performance of bus-stop arrival time prediction. As a comparison, bus schedule was used as a static arrival time prediction when there is no real-time information available. There are 3 percentile lines (10, 50 and 90 percentile) in Figure 5-10, for both the real-time prediction (blue lines) and static schedule-based prediction (red lines). The 50-percentile line is about the average prediction error, as a function of actual bus travel time to the stop. The area between the 10-

percentile and 90-percentile lines is where 80 percent of observations fall. The vertical gap between these two lines indicates the standard deviation of the prediction error. The larger the gap, the bigger the variance of the prediction error is. When compared with static schedule-based prediction, the 50-percentile lane of real-time prediction is closer to 0 (perfect prediction) for most of the time. The standard error of real-time prediction is always smaller than that of static schedule-based prediction. Moreover, the standard error of real-time prediction becomes smaller when the bus traveling towards the station and eventually converges to 0. This shows the value of dynamic passenger information. It provides better bus arrival information than is offered by published schedules and the information is dynamically getting more reliable.

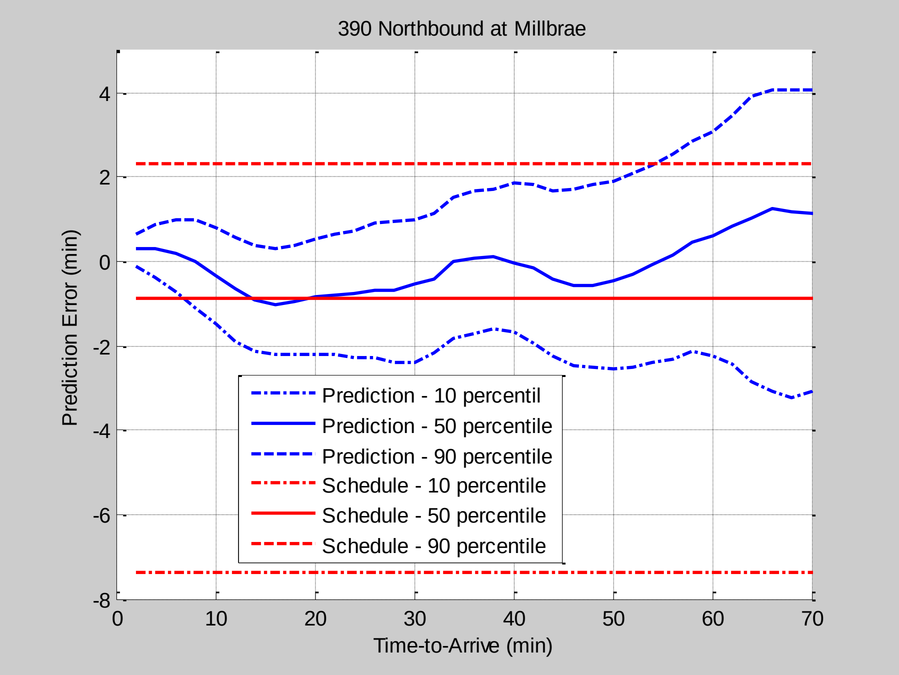


Figure 5-10 Prediction Error vs. Actual Travel Time to Bus-Stop (Case 1)
(At 390 Northbound Millbrae Station)

Figure 5-11 shows a result for the level 2 analysis, still at the northbound Millbrae Station. Three green percentile lines are included for the case that all bus polling data, including regular polls from the existing AVL/ACS and inserted polls enabled from the integrated ATSP/DPI, were used for arrival time prediction. At approximately 35 to 40 minutes of travel to the Millbrae Station, the bus started entering the ATSP corridor and additional inserted polls became available. As shown in Figure 5-11, inserted polls helped to reduce both the prediction error mean and its standard deviation.

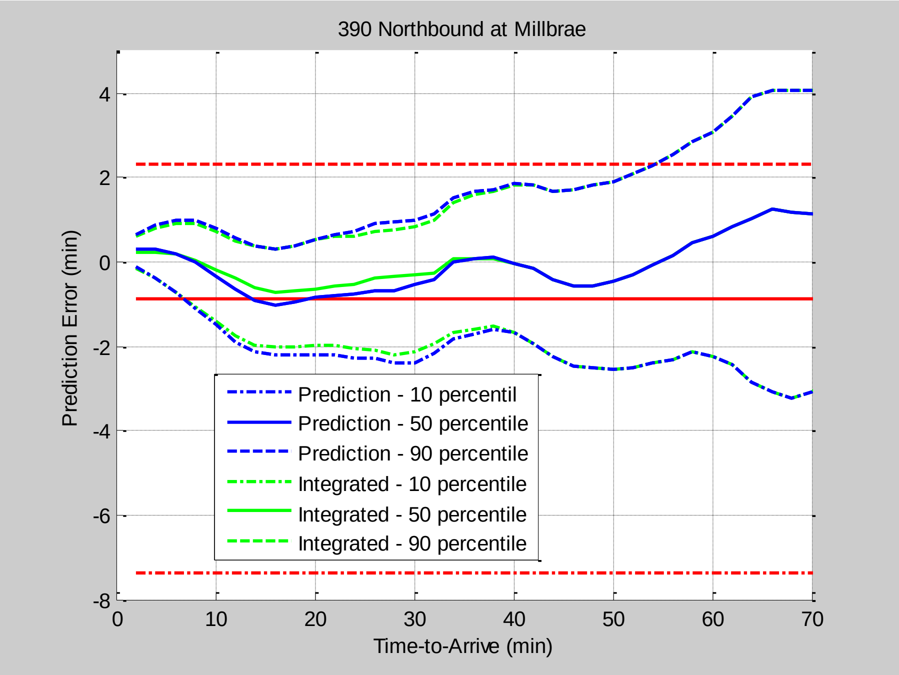


Figure 5-11 Prediction Error vs. Actual Travel Time to Bus-Stop (Case 2) (At 390 Northbound Millbrae Station)

In order to quantify the significance of inserted polls on the accuracy of arrival time prediction, mean absolute error (MAE) was compared for the two cases: 1) using regular polls only and 2) also using inserted polls. The mean absolute error is a commonly used accuracy measure for forecasts and prediction. It is given by

$$\mathrm{MAE} = \tfrac{1}{n}\sum_{i=1}^{n}|e_i|$$

where n is the number of samples and e_i is the prediction error of sample i.

A comparison of MAE performance was made for 4 time-points inside the ATSP corridor, namely Millbrae, Burlingame at ECR, 5th Ave at ECR and Hillsdale at ECR. The order of these 4 time-points is from north to south. The comparison at Hillsdale did not apply as it is too close to the southern end of the ATSP corridor. To make a fair comparison, only prediction samples that the bus was traveling inside the ATSP corridor were applied for the analysis. The results are summarized in Table 5-8. On average, inserted polls reduced MAE by 16%.

Table 5-8 MAE Comparison

Time-Point	MAE (Southbound, sec)			MAE(Northbound, sec)		
	Without Inserted Polls	With Inserted Polls	Percentage Change	Without Inserted Polls	With Inserted Polls	Percentage Change
Millbrae	89.8	76.6	-14.7%	102.9	82.9	-19.5%
Burlingame	105.4	89.2	-15.3%	79.1	66.0	-16.6%
5th Ave	103.9	88.3	-15.1%	73.0	61.4	-15.8%
Hillsdale	107.0	89.9	-16.0%	-	-	-

6. Summary and Conclusion

Automatic Vehicle Location (AVL) and Advanced Communication Systems (ACS), Transit Signal Priority (TSP) and Dynamic Passenger Information (DPI) have been widely deployed and have contributed to improved service quality and reliability. These systems share many common critical functions including in communication, onboard processing, vehicle locating, and capability of prediction of bus arrival time. However, they are separately implemented in the real world, resulting in redundant capital investments and additional costs for operations and maintenance. Integrated implementation will provide the full benefits enabled by the ITS technologies and will substantially improve the cost effectiveness. The study conducted by PATH reported herein investigated approaches to integrate ATSP and DPI systems with AVL/ACS.

Open system architecture is essential for the integrated system. Under this project, an integrated open architecture was developed that incorporate the following functional elements:

- The dynamic polling strategies enable the existing ACS communication link to provide more frequent bus location updates for the operation of ATSP and DPI;
- A flexible and scalable database has also been developed to support the open architecture ITS applications;
- Data pre-processing tools have been in place to query, parse and process data from different sources, such as AVL/ACS, 511 XML and Google transit feeds, and saved to a database;
- Bus arrival time prediction model has added transit operation factors in the prediction algorithm such that it can serve both short-term prediction for ATSP and long-term prediction for DPI;
- Adaptive Transit Signal priority algorithm triggers signal priority based on transit needs and traffic conditions; and
- Web APIs has been developed to deliver passengers dynamic information in various ways, pre-trip planning, information querying by smart phones and information display at bus-stops. DPI information has been displayed at Millbrae kiosks.

In order to field test the dynamic polling strategies without requiring manufacturer to make changes of the software on the buses, an ACS emulation environment has been developed to facilitate necessary additional bus location updates to support ATSP operation. Field testing results (two SamTrans bus routes along El Camino Real) showed the emulated dynamic polling under ACS emulation provided necessary additional bus location updates for the needs of ATSP without overwhelming the channel capacity of the existing ACS. Ninety-seven percent (97%) of inserted polls were processed without the collision of multiple competing polling requests. Even when collision of multiple polling requests did happen, the delay in responding to polling request was within 4 seconds.

Prediction of arrival time at bus-stops was fairly accurate, within +/- 2 minutes when the bus is 40 minutes away. Inserted polls for ATSP purposes contributed to better predicted results and typically reduced the absolute error by 16% at each of the time-points within the ATSP corridor.

The integrated ATSP and DPI have been tested at 50 intersections along El Camino Real through an emulated polling environment. The results indicate that ACS can indeed accommodate the ATSP communication needs. ATSP under emulated dynamic polling benefited transit services in several aspects:

- Bus travel time was reduced by 4.4% for southbound trips and by 1.4% for northbound trips;
- Bus total intersection delay was reduced by 19.4% for southbound and by 9.2% for northbound;
- Number of stops at prioritized intersections was reduced by 5.9% for southbound and by 4.1% for northbound; and
- Bus running speed was increased by 4.3% for southbound and by 1.5% for northbound;

All these changes are statistically significant at the 5% level. Moreover, the percentage of buses running over 5 minutes behind the schedule was reduced by 27%. The evaluation results also confirmed that the time-point holding phenomena due to TSP. With TSP savings at intersections, the bus tended to wait longer at the time-points. The average dwelling time per time-point per trip was increased by 22.5% for southbound and by 11% for northbound. The additional waiting time at the time points cancelled the time saved at the intersection by TSP. Excluding the dwelling time at time-points for both the "before" and "after" scenarios, bus trip travel time could be reduced by 5.6% for southbound and by 2.4% for northbound.

This project has successfully addressed a number of issues that remained for eventual integration of ATSP/DPI with AVL/ACS, including the open system architecture and the integrated database management to meet transit agencies' various needs, the interface with existing ACS for facilitating TSP, and the information process and dissemination methods. Field operational test results with emulated dynamic polling demonstrated that when integrated with ATSP and shared TSP operation status data, better DPI information will be achieved, and that there is a great potential for integrating ATSP/DPI and AVL/ACS.

References

AASHTO/ITE/NEMA. (2008). *National transportation communications for ITS protocol: Object Definitions for Signal Control and Prioritization, v01.*

ABDEL-ATY, M., KITAMURA, R., & JOVANIS, P. (1996). Investigating Effect of Advanced Traveler Information on Commuter Tendency To Use Transit. *Transportation Research Record* (1550), 65-72.

Adler, J. L., & Blue, J. (1998). Toward the design of intelligent traveler information systems. *Transportation Research Part C: Emerging Technologies , Vol. 6* (3), 157 – 172.

Aftabuzzaman, M., Currie, G., & Sarvi, M. (2010). Evaluating the Congestion Relief Impacts of Public Transport in Monetary Terms. *Journal of Public Transportation , 13* (1).

California PATH. (2011). *DPI information at bus stops*. Retrieved from ATSP/DPI server: http://tlab.path.berkeley.edu:8080/DPIStops/lists.html

California PATH. (2010). *Real-time DPI at Millbrae station*. Retrieved from ATSP/DPI server: http://tlab.path.berkeley.edu:8080/SamTransKioskMillbrae/index.htm

Craigslist. (n.d.). *MySQL Customer: craigslist*. Retrieved March 2011, from MySQL: http://www.mysql.com/customers/view/?id=531

D' Angelo, M., Al-Deek, H. M., & Wang, M. C. (1999). Travel-Time Prediction for Freeway Corridors. *Transportation Research Record* (1676), 184-191.

Dziekan, K., & Kottenhoff, K. (2007). Dynamic at-stop real-time information displays for public transport: effects on customers. *Transportation Research Part A: Policy and Practice , 41* (6), 489-501.

Ferris, B., Watkins, K., & Borning, A. (2011). OneBusAway: Behavioral and Satisfaction Changes Resulting from Providing Real-Time Arrival Information for Public Transit. *Paper #11-0440, TRB 90th Annual Meeing.* Washington, D. C.: Transportation Research Board.

FTA. (2006). *Advanced Public Transportation Systems: State of the Art Update 2006.* Washington, D.C.: Report #FTA-NJ-26-7062-06.1, FTA.

FTA. (2008). *Transit Signal Priority Research Tools.* FTA-CA-26-7049-2008.1, Federal Transit Administration.

Google. (2010, January 11). *General Transit Feed Specification*. Retrieved March 22, 2011, from Google Code: http://code.google.com/transit/spec/transit_feed_specification.html

Google. (2011, March 14). *List of publicly-accessible transit data feeds* . Retrieved March 22, 2011, from Google Code: http://code.google.com/p/googletransitdatafeed/wiki/PublicFeeds

Grotenhuis, J. -W., Wiegmans, B. W., & Rietve, P. (2007). The desired quality of integrated multimodal travel information in public transport: Customer needs for time and effort savings. *Transport Policy , 14* (1), 27 – 38.

Harford, J. (2006). Congestion, Pollution, and Benefit-to-Cost Ratios of US Public Transit Systems. *Transportation Research Part D: Transport and Environment , 11* (1), 45-58.

Jasper, K., Armstrong, C., Golembiewski, G., & Miller, S. (2011). *National Evaluation of the SafeTrip-21 Initiative: California Connected Traveler Test Bed, Draft Evaluation Report: Networked Traveler – Transit/Smart Parking.* US DOT, ITS Joint Program Office.

Kenyon, S., & Lyons, G. (2003). The value of integrated multimodal traveller information and its potential contribution to modal change. *Transportation Research Part F: Traffic Psychology and Behaviour , 6* (1), 1-21.

Li, J.-Q., Zhou, K., Zhang, L., & Zhang, W.-B. (2010). A Multimodal Trip Planning System Incorporating the Park-and-Ride Mode and Real-time Traffic and Transit Information. *Proceedings of ITS World Congress.* Busan, Korea.

Litman, T. (2011). *Evaluating Public Transit Benefits and Costs: Best Practices Guidebook.* Victoria Transport Policy Institute.

Lyons, G., Jain, J., & Holley, D. (2007). The use of travel time by rail passengers in Great Britain. *Transportation Research Part A: Policy and Practice , 41* (1), 107-120.

Nextbus. (n.d.). *Nextbus Agencies*. Retrieved March 2011, from http://www.nextbus.com/predictor/agencySelector.jsp

PATH Report. (2008). *Field Evaluation of San Pablo Corridor Transit Signal Priority (TSP) System.* Report # UCB-ITS-PWP-2008-07, California PATH Program, UC Berkeley.

PATH Report. (2006). *Field Evaluation of VTA Rapid 522 Corridor Transit Signal Priority System.* California PATH Program, UC Berkeley.

PATH Report. (2010). *Field Operational Tests of Adaptive Transit Signal Priority Systems (ATSP).* PATH Report #UCB-ITS-PRR-2010-35, California PATH Program, UC Berkeley.

PATH Report. (2010). *Field Operational Tests of Adaptive Transit Signal Priority Systems (ATSP).* PATH Report #UCB-ITS-PRR-2010-35, California PATH Program, UC Berkeley.

Schrank, D., & Lomax, T. (2009). *2009 Urban Mobility Report.* Texas Transportation Institute.

Shalaby, A., & Farhan, A. (2003). Bus Travel Time Prediction Model for Dynamic Operations Control and Passenger. *TRB Annual Meeting.* Washington, D.C.

Smith, H., & Hemily, B. (2005). *Transit Signal Priority (TSP): A Planning and Implementation Handbook.* ITS America.

Song, H.-L. (1994). Automatic vehicle location in cellular communications systems. *IEEE Transactions on Vehicular Technology , 43* (4), 902-908.

TRB. (2008). *AVL Systems for Bus Transit: Update.* TCRP SYNTHESIS 73, Transportation Research Board.

TRB. (2003a). *Real-Time Bus Arrival Information Systems: A Synthesis of Transit Practice.* TCRP SYNTHESIS 48, Transportation Research Board.

TRB. (2003b). *Strategies for Improved Traveler Information.* TCRP Report 92, Transportation Research Board.

Turnbull, K., & Pratt, R. (2003). *Transit Information and Promotion: Traveler Response to Transport System Changes: Chapter 11.* TCRP Report 95, Transportation Research Board.

US DOT. (2006). *National Strategy to Reduce Congestion on America's Transportation Network. United States Department of Transportation.*

van Exel, N., & Rietveld, P. (2010). Perceptions of public transport travel time and their effect on choice-sets among car drivers. *Journal of Transport and Land Use , 2* (3/4), 75-86.

Wener, R., & Evans, G. (2003). Comparing stress of car and train commuters. *Transportation Research Part F: Traffic Psychology and Behaviour , 6* (1), 1-21.

Wikepedia. (n.d.). *MySQL*. Retrieved March 2011, from Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/MySQL

Wikipedia. (n.d.). *Ajax (programming)*. Retrieved March 2011, from Wikipedia: http://en.wikipedia.org/wiki/Ajax_(programming)

Wilbur Smith Associates / Harley & Associates. (2006). *Hub review findings - Millbrae BART/Caltrain Station.* MTC Regional Transit Hub Performance Review Project. MTC.

Winston, C., & Langer, A. (2006). *The Effect of Government Highway Spending on Road Users' Congestion Costs.* Brookings Institution.

Zhang, L., LI, J.-Q., Zhang, W.-b., Zhou, K., Datta Gupta, S., Miller, M. A., et al. (2011). *SafeTrip-21 Connected Traveler: Networked Traveler Transit and Smart Parking.* California PATH, University of California, Berkeley.

Zhang, L., Li, J.-Q., Zhou, K., Datta Gupta, S., Li, M., Zhang, W.-b., et al. (2011). Design and Implementation of a Traveler Information Tool with Integrated Real-time Transit Information and Multi-modal Trip Planning. *TRB Annual Meeting.* Washington, DC.

Appendix A. Comparison Results of ATSP Benefits

The measures of effectiveness (MOEs) for evaluating ATSP impacts on transit operations, as presented in Section 5.2.2, include bus trip travel time, total intersection delay, number of stops at prioritized intersections, average running speed, average intersection delay per stopped intersection, and schedule deviation at time-points. The before-and-after comparison of each of the MOEs is presented in this Appendix.

The "before" scenario is the case of "without ATSP" and after "with ATSP". Empirical cumulative distribution function (CDF) plots for both scenarios depict the changes with respect to MOEs.

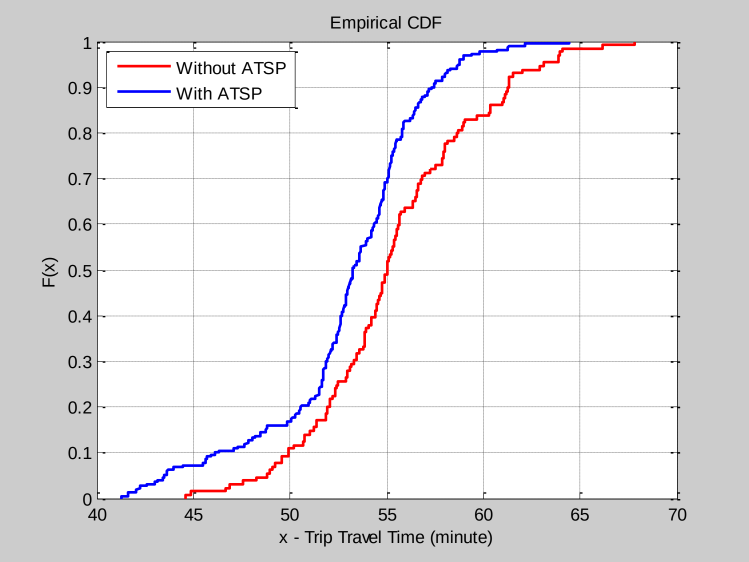


Figure A-1 Comparison of Bus Trip Time (Southbound)

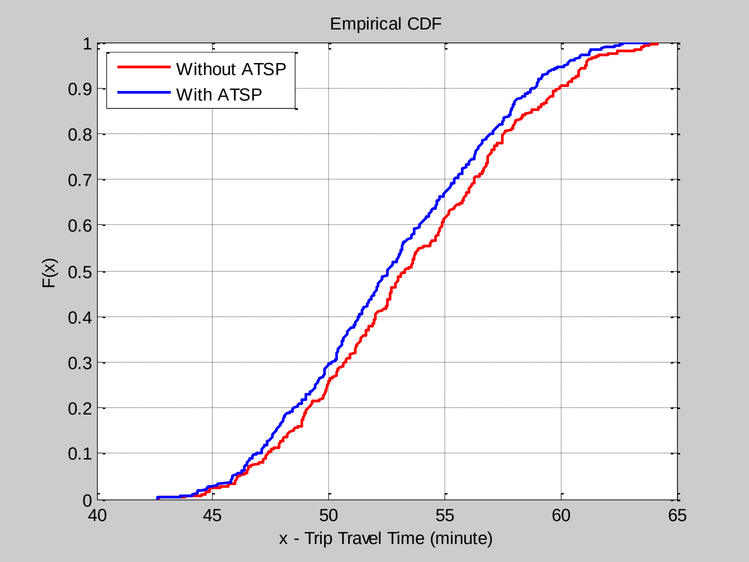


Figure A-2 Comparison of Bus Trip Time (Northbound)

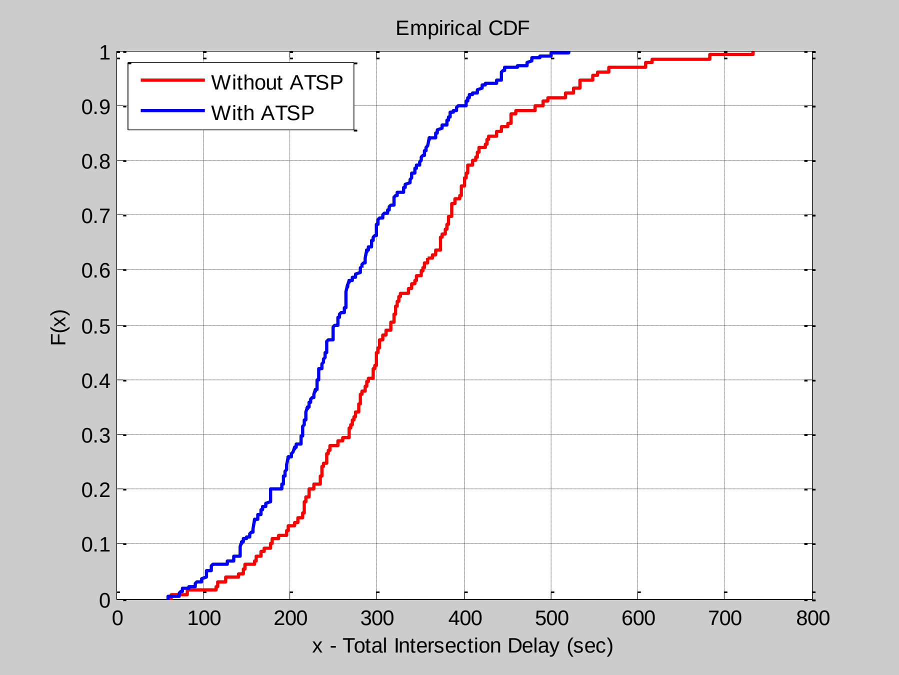


Figure A-3 Comparison of Total Intersection Delay (Southbound)

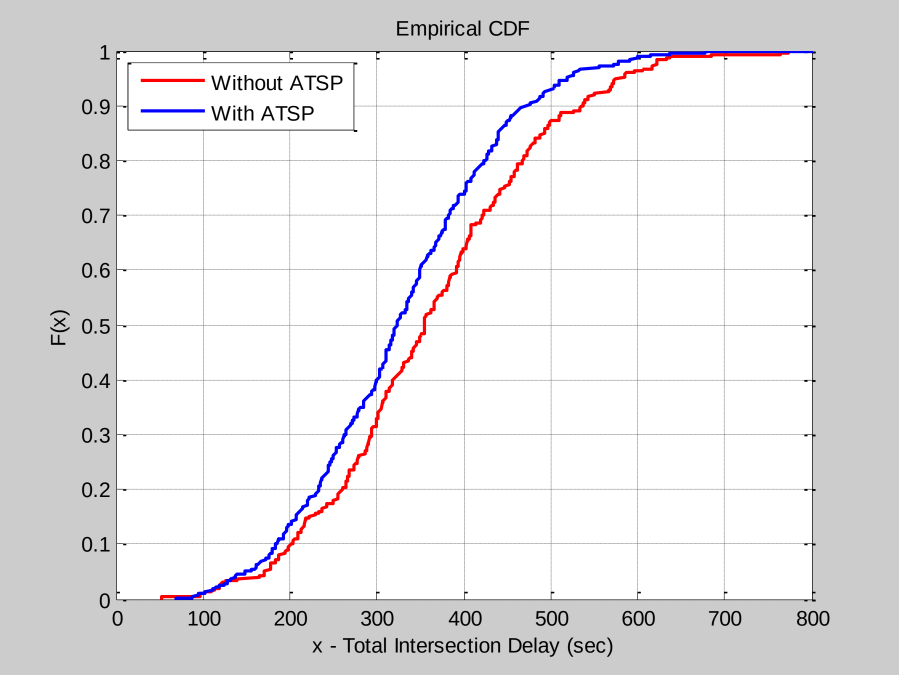


Figure A-4 Comparison of Total Intersection Delay (Northbound)

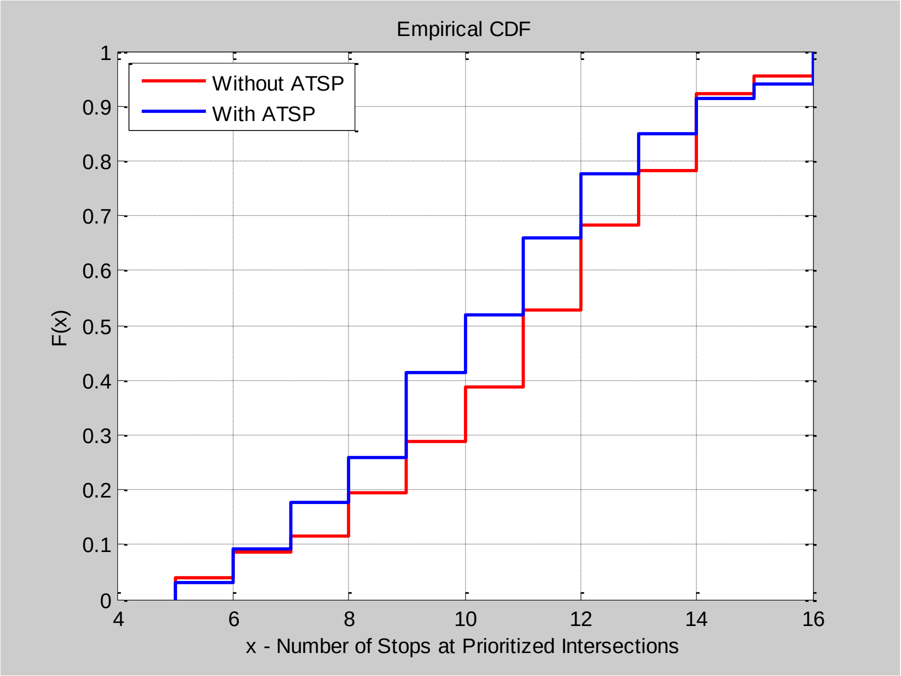


Figure A-5 Comparison of Number of Stops due to Traffic Signals (Southbound)

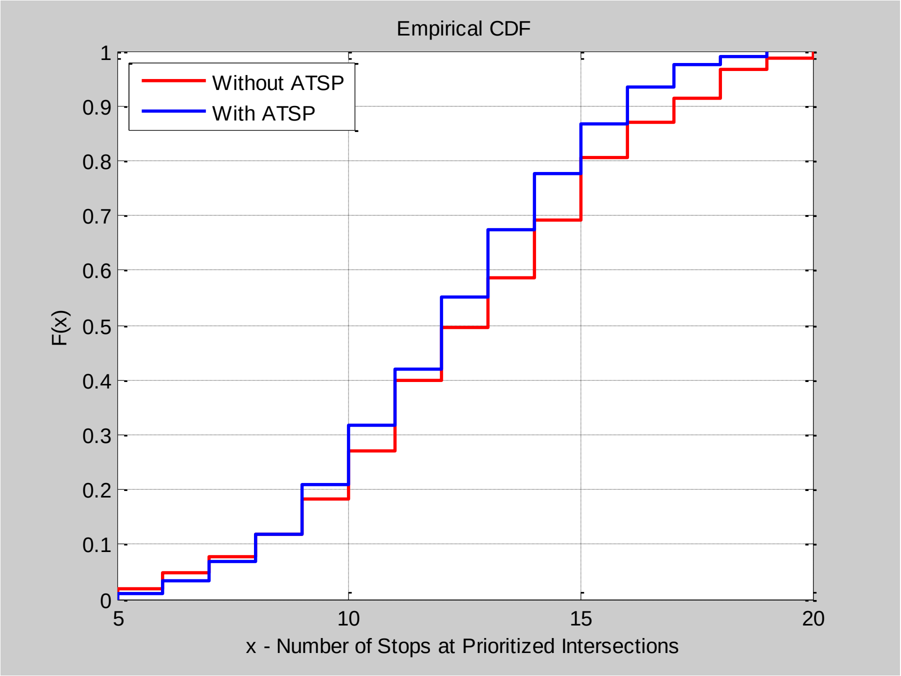


Figure A-6 Comparison of Number of Stops due to Traffic Signals (Northbound)

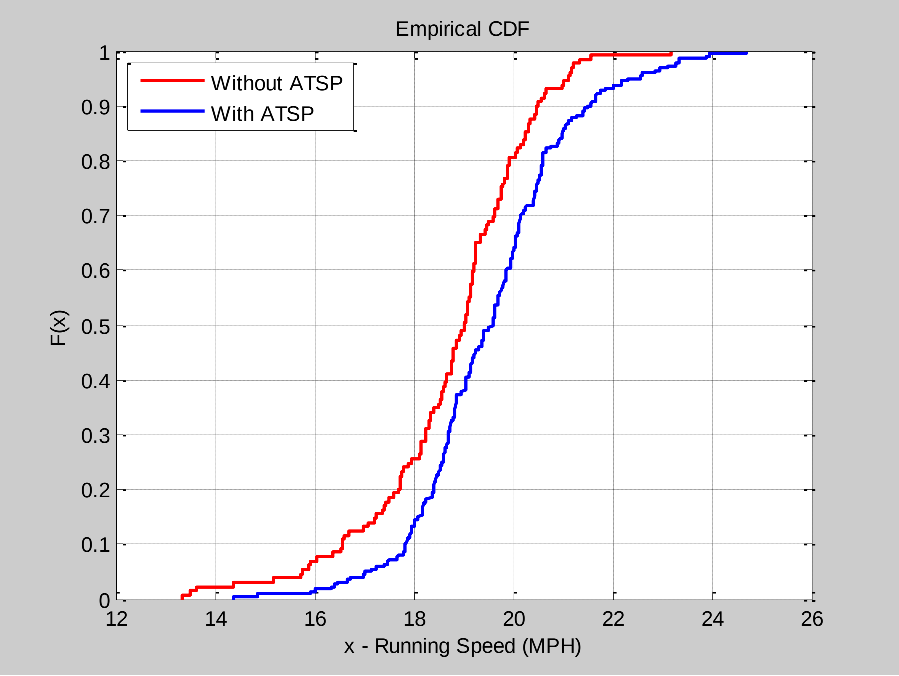


Figure A-7 Comparison of Bus Running Speed (Southbound)

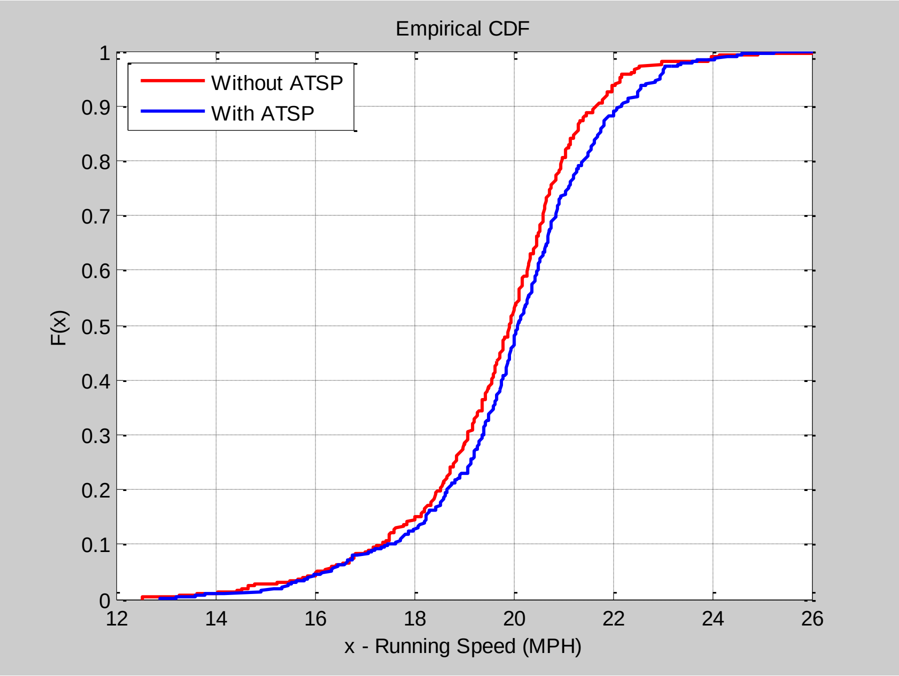


Figure A-8 Comparison of Bus Running Speed (Northbound)

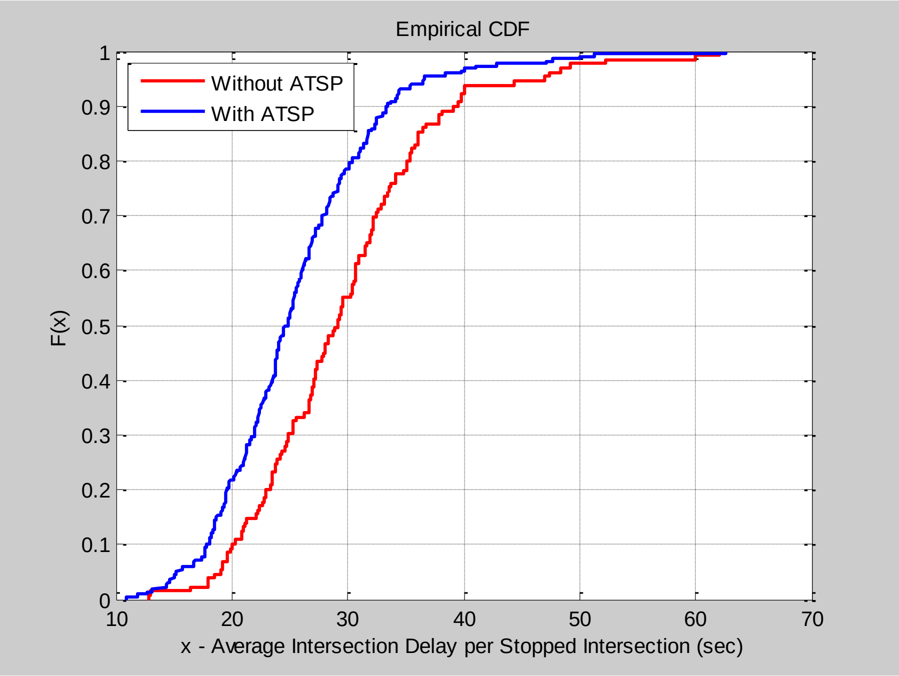


Figure A-9 Comparison of Average Intersection Delay per Stop at Red (Southbound)

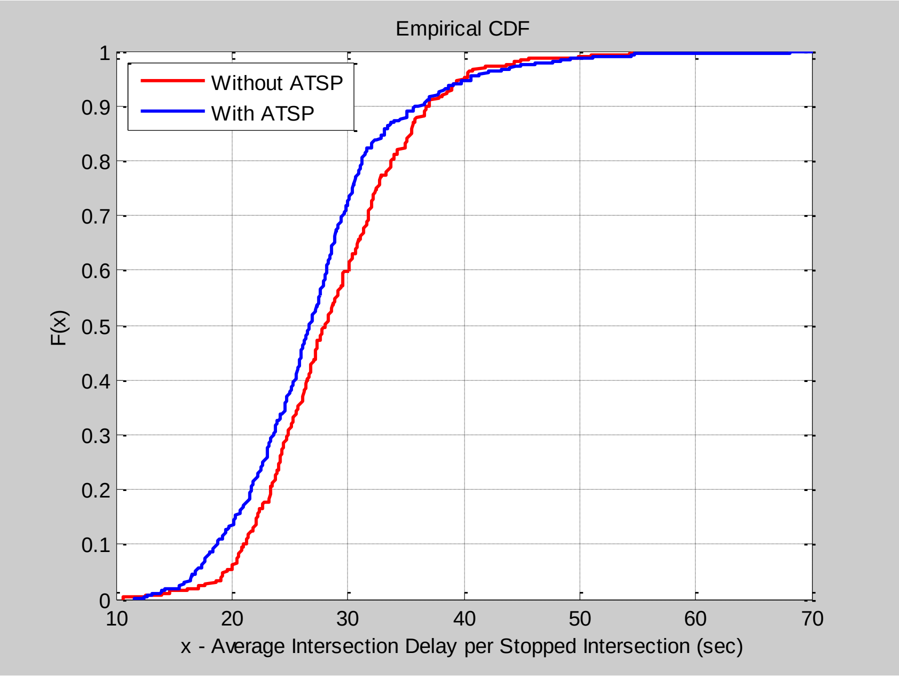


Figure A-10 Comparison of Average Intersection Delay per Stop at Red (Northbound)

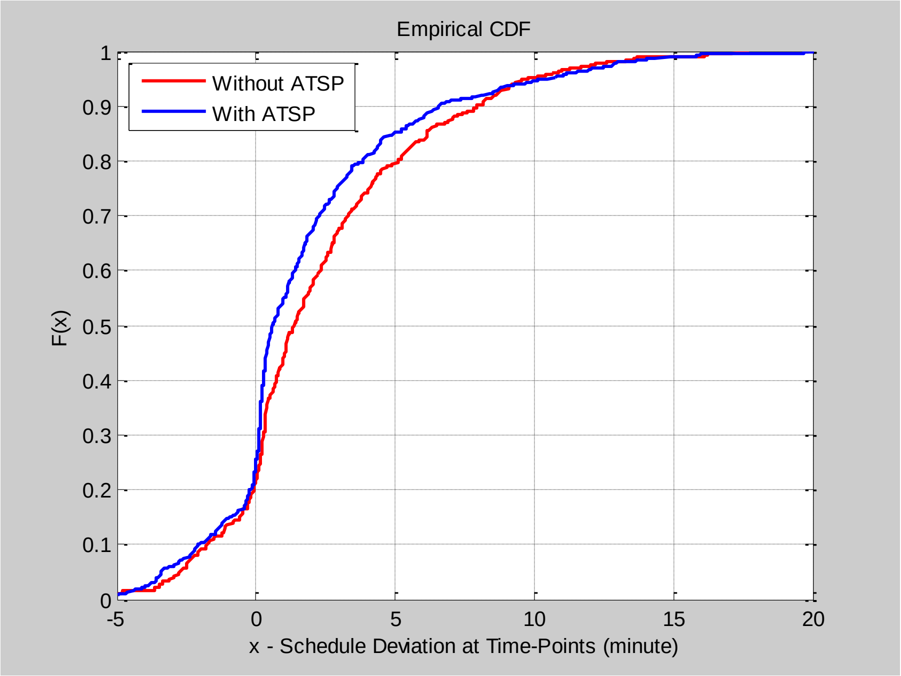


Figure A-11 Comparison of Schedule Deviation at Time-Points (Southbound)

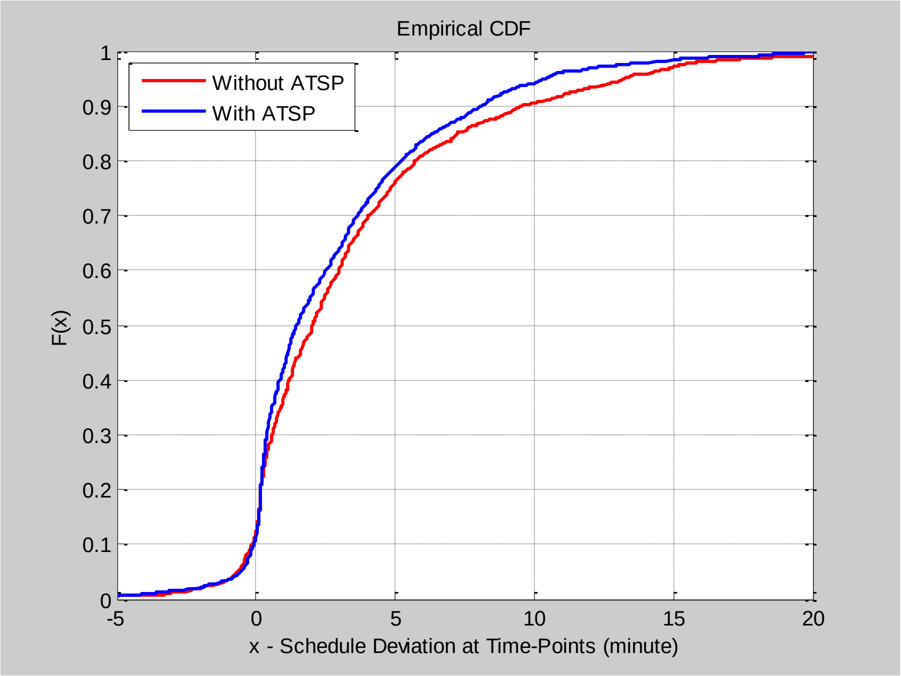


Figure A-12 Comparison of Schedule Deviation at Time-Points (Northbound)

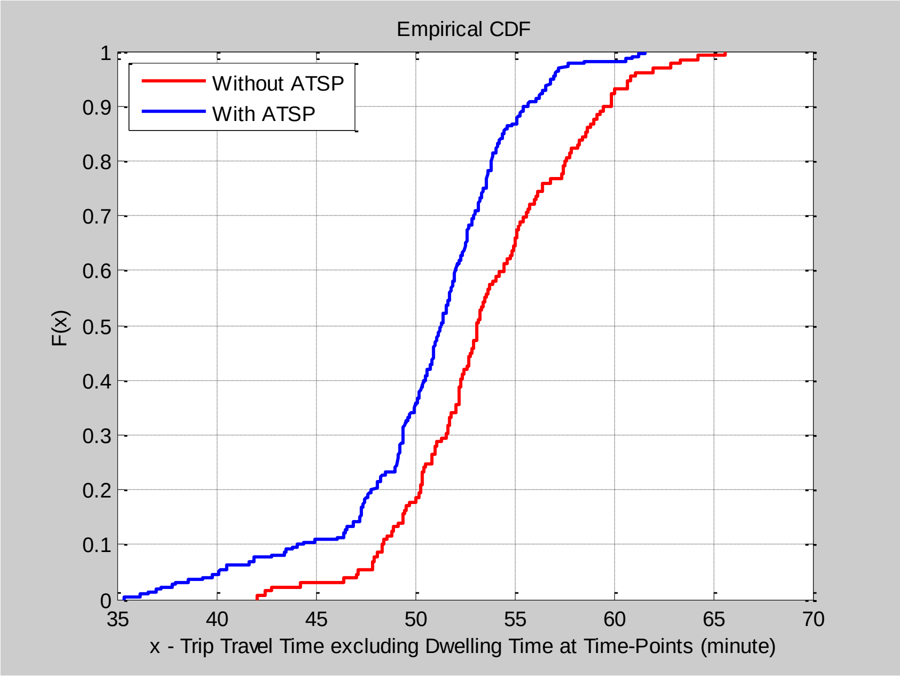


Figure A-13 Comparison of Bus Trip Travel Time (Excluding Dwelling Time at Time-Points, Southbound)

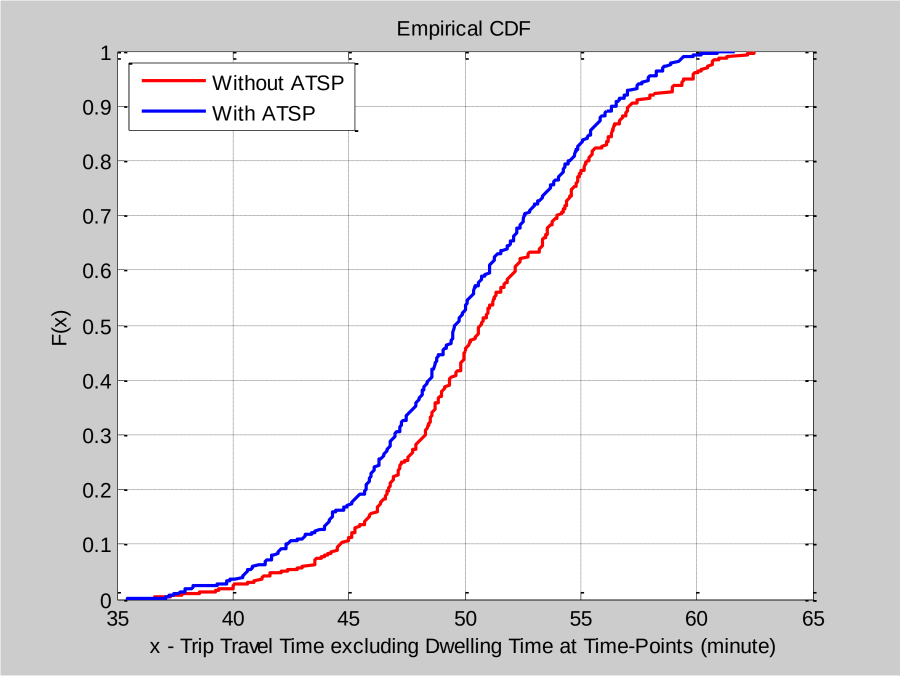


Figure A-14 Comparison of Bus Trip Travel Time (Excluding Dwelling Time at Time-Points, Northbound)