Multi-Modal Intelligent Traffic Signal System

Final Report

University of Arizona (Lead)
University of California PATH Program
Savari Networks, Inc.
Econolite

Version 1.2

9/5/2016

RECORD OF CHANGES

A – Added, M- Modified, D - Deleted

Version Number	Date	Identification of Figure, Table, or Paragraph	Title or Brief Description	Change Request Number
1.0	5/3/2015	N/A	Initial Draft for Team Review/Development	
1.1	3/24/2016		Integration of CA MMITSS Final Report Material	
1.2	6/9/2016		Draft Submitted for Review (with Line Numbers for Reviewers)	
1.3	9/5/2016		Response to Reviewer Comments (Line Numbers Removed)	

Table of Contents

List of Figures

List of Tables

1 Purpose of Document

This document constitutes the final report for the Multi-Modal Intelligent Traffic Signal Systems (MMITSS) Phase II Development and Field Testing Project. The purpose of this document is to capture project details and experience in addition to the documentation in the MMITSS Concept of Operations, MMITSS System Requirements, and MMITSS Detailed Design Documents including laboratory and field-testing procedures and experience, simulation modeling approach and tools developed, general impressions and findings, and recommendations for future MMITSS development and deployment opportunities.

The Multi-Modal Intelligent Traffic Signal System (MMITSS) project is part of the Connected Vehicle Pooled Fund Study (CV PFS) entitled “Program to Support the Development and Deployment of Connected Vehicle System Applications.” The CV PFS was developed by a group of state and local transportation agencies and the Federal Highway Administration (FHWA). The Virginia Department of Transportation (VDOT) serves as the lead agency and is assisted by the University of Virginia’s Center for Transportation Studies, which serves as the technical and administrative lead for the PFS. Maricopa County Department of Transportation and the California Department of Transportation provided significant cost share as well as support for the field implementation and testing throughout the project. The MMITSS project includes several applications in one of the mobility bundles under the United States Department of Transportation’s Dynamic Mobility Application (DMA) program.

The MMITSS project was divided into four technical segments. The development of the Concept of Operations (ConOps), including the solicitation of Stakeholder inputs and feedback, was the first technical stage. The reviewed Stakeholder inputs and ConOps were used to develop, define, and populate the MMITSS system requirements in the second technical stage. In the third stage, the system requirements and prior research were used to define the MMITSS system design. The design effort utilized the California test bed and the Arizona test bed as the target implementation networks. The final stage included Detailed Design, System Implementation and Integration, Field Integration and Testing, and a System Demonstration in each of the California and Arizona test beds.

2 Background and Overview

Traditional approaches to traffic signal control have been based on low fidelity detection and less intelligent control techniques that typically are not well informed about the state of the traffic and are not necessarily modally aware. Use of special devices for emergency vehicle preemption and transit signal priority are exceptions. The state of the practice in traffic control today is actuated traffic signal control where different conflicting movements of vehicles are controlled by phases that are called by fixed-location detectors when vehicles are present at the intersection. Typically, intersections are operated in a time-based coordinated operation with a signal timing plan that has been designed to accommodate different volumes of flow based on time-of-day and day-of-week traffic flow patterns. At the intersection level, traffic control decisions are made based on vehicle detections that call and extend phases, or from pedestrian push buttons that call for the pedestrian phase intervals (Walk, Pedestrian Clearance, and Don’t Walk). At the section (or arterial) level, collections of traffic signals can be coordinated to allow vehicles to progress along an arterial or desired path of vehicle travel. Each intersection of the coordinated arterial operates using the same actuated control principles as single, isolated intersections, but is constrained by the time-of-day signal timing plans that can provide coordination through a fixed cycle length, offset, and a set of phase splits. Emergency vehicle preemption and transit signal priority can modify the coordinated as well as the actuated control to provide some (e.g. high, moderate, or low) degree of priority at signals for certain classes of vehicles. These vehicles communicate with the traffic signal controller by sending a message that will call a special preemption or priority control logic and timing feature on the traffic controller. The advent of connected vehicle (CV) technology and systems provides the first real opportunity for multi-modal control where intelligence can be used to provide cooperative services and priority to each mode and to allow multiple priority requests to be simultaneously considered. For the purpose of this project, these modes include general passenger vehicles, pedestrians, transit, freight, and emergency vehicles.

Connected vehicle technologies enable vehicles to exchange information with each other (vehicle-to-vehicle [V2V]) and with the roadside infrastructure (vehicle-to- infrastructure [V2I]) in real time. These technologies also enable nomadic devices such as smart phones and tablets carried by pedestrians and bicyclists to exchange real-time information about their movements with the roadside infrastructure. At the national level, when the Vehicle-Infrastructure Integration (VII) program began work on what is now known as connected vehicle technologies, the focus was entirely on the use of 5.9 GHz dedicated short-range communication (DSRC) technology. In recent years, the focus has broadened to include a full array of wireless technologies such as 3G and 4G/LTE cellular, Wi-Fi, and Bluetooth. Throughout this report, the mobile devices in vehicles are referred to as On-Board Equipment (OBE), those carried by non-motorized travelers are referred to as nomadic devices, and infrastructure-based devices are referred to as Roadside Equipment (RSE). Although these terms were originally associated with DSRC technology, here they can refer to any of the wireless technologies.

In a connected vehicle environment, an RSE associated with an intersection signal controller broadcasts a MAP and SPaT (signal phase and timing) message. An equipped vehicle (e.g. OBE equipped) that enters the radio range of the RSE will receive the MAP and SPaT data and will actively broadcast basic safety messages (BSMs). If a vehicle is eligible, and authorized, for signal priority/preemption, the vehicle may send a signal request message (SRM) to request signal priority or preemption and receive a signal status message (SSM) with the acknowledgements of priority requests and the status of active priority/preemption request(s). A nomadic device carried by a pedestrian will also receive the MAP and SPaT data. A pedestrian may use a nomadic device to send a signal request message (SRM) which in turn will place a pedestrian call to the associated intersection signal controller and receive a signal status message (SSM) that acknowledges the receipt of the pedestrian SRM. Pedestrian requests could be used to provide priority, hence reducing the waiting time of pedestrians.

Real-time message exchanges between connected travelers (e.g. OBEs and nomadic devices) and roadside infrastructure (e.g. RSEs and associated signal controllers) allow traffic signal control and signal priority for multiple modes (pedestrians, general vehicles, transit, freight, and emergency vehicles) to be managed within an integrated framework. Vehicles that are eligible for signal priority communicate their desire for priority and critical information to the roadside infrastructure. The allocation of priority levels is determined by involved Stakeholders (e.g., local agencies, transit operators, and freight operators) enabling the effective management of signal priority control utilizing the SRMs. Different levels of priority for eligible vehicles, whether multi-modal or within the same mode, can be assigned based on the local interpretation of signal priority importance and usefulness. For example, emergency vehicles in Active Response Mode should have a higher level of priority than transit and freight vehicles; transit vehicles could be assigned a higher level-of-priority than freight vehicles in one corridor and freight vehicles can be assigned a higher level of priority than transit vehicles in another corridor where the truck volume is high. Implementing levels of priority provides the flexibility to configure the policy for signal priority control based on the local or regional characteristics and transportation management goals.

3 Scope of Project

The Multi-Modal Intelligent Traffic Signal System (MMITSS) project is part of the Connected Vehicle Pooled Fund Study (CV PFS) entitled “Program to Support the Development and Deployment of Connected Vehicle System Applications.” The CV PFS was developed by a group of state and local transportation agencies and the Federal Highway Administration (FHWA). The Virginia Department of Transportation (VDOT) serves as the lead agency and is assisted by the University of Virginia’s Center for Transportation Studies, which serves as the technical and administrative lead for the PFS.

The US DOT has identified six mobility application bundles (http://www.its.dot.gov/research_archives/dma/dma_plan.htm) under the Dynamic Mobility Applications (DMA) program for the connected vehicle environment where high-fidelity data from vehicles, infrastructure, pedestrians, etc. can be shared through wireless communications. Each bundle contains a set of related applications that are focused on similar outcomes. Since a major focus of the CV PFS members – who are the actual owners and operators of transportation infrastructure – lies in traffic signal related applications, the CV PFS team is leading the project entitled “Multi-Modal Intelligent Traffic Signal System” in cooperation with US DOT’s Dynamic Mobility Applications Program. As one of the six DMA application bundles, MMITSS includes five applications: Intelligent Traffic Signal Control (I-SIG), Transit Signal Priority (TSP), Mobile Accessible Pedestrian Signal System (PED-SIG), Freight Signal Priority (FSP), and Emergency Vehicle Preemption (PREEMPT).

The MMITSS project is divided into four technical segments. The development of the ConOps, including the solicitation of Stakeholder inputs and feedback, was the first technical stage. The reviewed Stakeholder inputs and Concept of Operations were used to develop, define, and populate the MMITSS system requirements in the second technical stage. In the third stage, the system requirements and prior research were used to define the MMITSS system design. The design effort utilized the California Test Bed and the Maricopa County Test Bed as the target implementation networks. The last stage included implementation, integration, deployment, and field-testing. This report summarizes the entire project process.

During the development process, it was determined that the requirements of the MMITSS system to be implemented in the California Test Bed were different than those for the Arizona Test Bed. The primary differences were in the hardware architecture, including the existing traffic signal controllers used in each network, algorithms for transit priority and intelligent signal control. The California testbed utilize the Caltrans Type 2070 controllers with AB3418 protocol over serial RS-232 communications. The Arizona testbed utilize Econolite ASC/3 and Cobalt controllers with NTCIP (National Transportation Communications for ITS Protocol) over Ethernet communications. Both testbeds utilize loop detectors for vehicle detection and push-buttons for pedestrian-crossing requests. Each of these controllers offers different signal timing logic (control software) and require different communications interfaces. The PATH team elected to utilize an alternative software architecture and alternative algorithms that they felt better satisfied their needs and system architecture. There are many common components, but the core software architecture and control algorithms are different.

Both testbeds utilize Savari RSE units (StreetWave version 3.1) for managing all of the 5.9 GHz DSRC communications between the vehicles and the roadside infrastructure. The California testbed chose the “thin” RSE deployment model, where the core MMITSS intersection-level applications are hosted by a Linux-based computer (e.g. the MMITSS Roadside Processor – MRP), which is installed inside the traffic signal cabinet and connected with the RSE using Ethernet. The Arizona testbed chose the deployment model that integrates the MRP with the RSE.

The Arizona testbed implemented adaptive signal control and the California testbed chose to enhance the existing coordinated-actuated signal control with a common cycle length among coordinated signals and fixed phase sequence at each intersection. As a result, the algorithms implemented for intelligent signal control and signal priority are different for the two testbeds. This Final Report captures the experience from both the Arizona implementation and the California implementation.

4 Project Summary

The MMITSS project was conducted using formal Systems Engineering processes starting from engaging stakeholders to identify their needs, development of use cases, and development of a full systems Concept of Operations (ConOps), development of a system architecture and detailed system design, implementation, laboratory testing, and field testing and demonstration. This section of the MMITSS Final Report documents each of these steps in the project process.

4.1 Stakeholder Needs and Use Case Development

Stakeholder engagement was an important aspect throughout the MMITSS project. Starting from the initial vision of the CV Pooled Fund members, as captured in the Request for Proposals, to the MMITSS team proposal, through several hosted field demonstrations (in Arizona), engagement in professional and scholarly conference, and specific project activities including a stakeholder workshop and several project reviews with the CV Pooled Fund members, the stakeholders input was a valuable part of the project.

As it relates to the MMITSS project, the term *Stakeholder* was used to reference a potential user, operator, designer, facilitator, supplier, integrator, owner, or related functional role of the proposed system. In addition, this term was used to include experts in specific aspects of the system, including interfaces, standards, policy and decision support, operations, support, and maintenance of the resulting system.

The MMITSS Stakeholder Meeting was conducted on June 4, 2012. A detailed stakeholder report is available at http://www.cts.virginia.edu/wp-content/uploads/2014/05/PFS_MMITSS02_Task2.2.pdf. Stakeholders were asked to review a collection of scenarios and provide input, feedback, and guidance on user needs, transformative goals, and potential system performance measures. MMITSS Stakeholders provided coverage at local, state, federal, industry, and supplier levels for five transportation areas: (1) intelligent traffic signal systems, (2) transit, (3) pedestrian mobility, (4) freight, and (5) emergency vehicles.

The stakeholder needs were used to develop a formal Concept of Operations that included use cases based on the scenarios that were used to guide the stakeholder needs process. Individual use cases were discussed for each of the core MMITSS applications including:

- ISIG: Intelligent Traffic Signal Control
- TSP: Transit Signal Priority
- PED-SIG: Pedestrian Mobility
- FSP: Freight Signal Priority
- EVP: Emergency Vehicle Priority (Preemption).

In addition, the stakeholders and Pooled Fund panel suggested two overarching operation scenarios that highlighted the multi-modal capabilities of MMITSS. Table 1 summarizes the operational scenarios and use cases identified. The two MMITSS Operational Scenarios assume that a system of traffic signals is divided into two traffic control sections. Network Section 1 is assumed to be a commercial corridor where freight vehicles move goods to and from warehouses or factories to a nearby interstate as illustrated in Figure 1. In this traffic control section, freight vehicles are given a higher consideration (level) for priority than coordination for passenger vehicles, transit, and pedestrians. It is assumed that rail crossings and emergency vehicles have the highest consideration for priority. Network Section 2 is assumed to be a more residential section where transit and pedestrians are considered to have a higher consideration for priority than freight. Again, rail crossings and emergency vehicles have the highest consideration. These overarching operational scenarios are intended to capture the need for MMITSS to be able to implement multi modal priority. The detailed Concept of Operations and Use Cases are available at: http://www.cts.virginia.edu/wp-content/uploads/2014/05/Task4._SystemDesign_3_Revised_v.2.pdf .

Table 1. MMITSS Operational Scenarios and Use Cases

	Operational Scenarios/Use Cases	**Include**	**Defer**	**AZ**	**CA**
11	MMITSS Operational Scenario				
11.0.1	Network Section 1	X		X	X
11.0.2	Network Section 2	X			X
11.1	Intelligent Traffic Signal System Scenarios				
11.1.1	Basic Signal Actuation	X		X	X
11.1.2	Coordinated Section of Signals		X		
11.1.3	Congestion Control		X		
11.1.4	Dilemma Zone Protection	X		X	X
11.2	TSP Operational Scenarios				
11.2.1	Basic TSP Scenario and Variations	X		X	X
	Nearside Bus Stop		X		
	Transit Signal Priority for Left Turn with Protected Signal		X		
11.2.2	Operational Scenarios for Rail Crossings in Urban Areas		X		
11.2.3	Extended TSP Scenario		X		
11.3	Pedestrian Mobility Operational Scenarios				
11.3.1	Unequipped Non-Motorized Traveler	-			
11.3.2	Equipped Non-Motorized Traveler	X		X	X
11.3.3	Equipped Bicyclist		X		
11.3.4	Inclement Weather Accommodations for Non-Motorized Travelers		X		
11.4	Freight Signal Priority Operational Scenarios				
11.4.1	Basic Freight Signal Priority	X		X	X
11.4.2	Coordinated Freight Signal Priority along a Truck Arterial		X		
11.5	Emergency Vehicle Priority				
11.5.1	Single Intersection Priority/Preemptions	X		X	
11.5.2	Route Based Intersection Priority/Preemption		X		

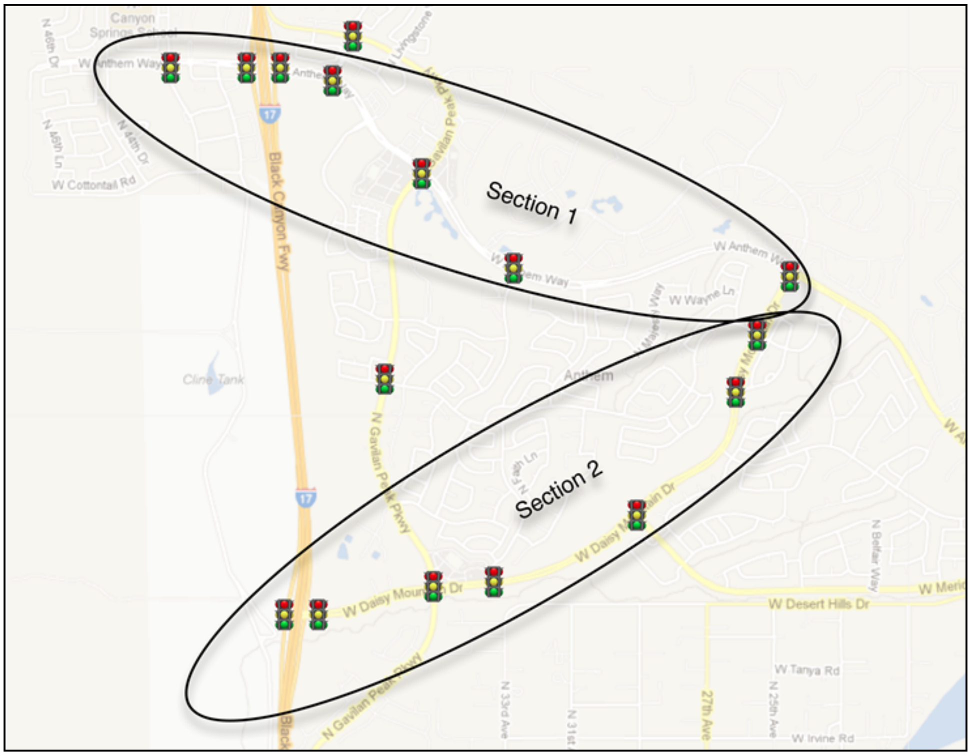


Figure 1. A traffic control system divided into two traffic control sections.

Table 1 also includes a summary of the Use Cases that were selected for design, implementation, and field testing based on the characteristics of the two test beds (Arizona and California), resources available for development, and the desire to provide a rich set of multi modal intelligent signal control capabilities. Several use cases were selected to be deferred until future efforts due to resources and network characteristics.

4.2 System Requirements

The MMITSS Stakeholder Needs and Concept of Operations were used to derive the MMITSS System Requirements which provide a listing of requirements and critical, supporting information (e.g., constraints) on the basis of which the MMITSS system was designed. The requirement was not a system specification which is a listing of requirements and constraints on the basis of which a system can be built or mass-produced. A system specification would provide each detail of the design, whereas the MMITSS requirements document allows further specification of implementation detail to support the differences between the systems and infrastructure deployed in the Arizona Test Bed and the California Test Bed.

The MMITSS System Requirements Document was developed using a customized version of the IEEE-1233 Requirements Development Process. The customization incorporates the specific MMITSS tasking for a draft submittal, requirement workshop review, and final submittal. These elements, along with the corresponding deliverable (CDRL) and timeline, are shown in Figure 2. As noted on the left side of the figure, the initial source of Customer/Stakeholder, Environment, Customer, and Technical Community requirement inputs are based on the MMITSS ConOps, which provides traceability to specific Stakeholder inputs in the ConOps.

Using this requirements development process, an "Initial Set of Requirements" was defined during a meeting of University of Arizona and PATH team members in Phoenix, Arizona on October 8-10, 2012. After this initial definition, the "Refine Requirements Definition" was achieved by executing the requirement review process shown in Figure 3. The first step of this review process called for each MMITSS subject matter expert to swap requirements in an effort to ensure proper coverage. The review process employed an iterative approach with the MMITSS systems engineer to ensure compliance with the MMITSS Requirements Checklist and acceptable practices listed in the reference section of the MMITSS System Requirements Document.

Figure 2 also conveys that the contents of the draft requirements document served as the basis of the MMITSS Requirements Walkthrough scheduled on December 4, 2012 in Irvine, CA. This working meeting provided details on the systems engineering methodology employed by MMITSS, the mapping of MMITSS use cases-to-requirements, data and interface requirements, functional requirements analysis, and traceability of requirements to the MMITSS ConOps and Stakeholder feedback in greater detail than draft requirements document. The Final MMITSS Requirements Document was intended to show the results of the requirements development processes, analyses, and integration of Sponsor feedback from the Requirements Walkthrough meeting on December 4, 2012.

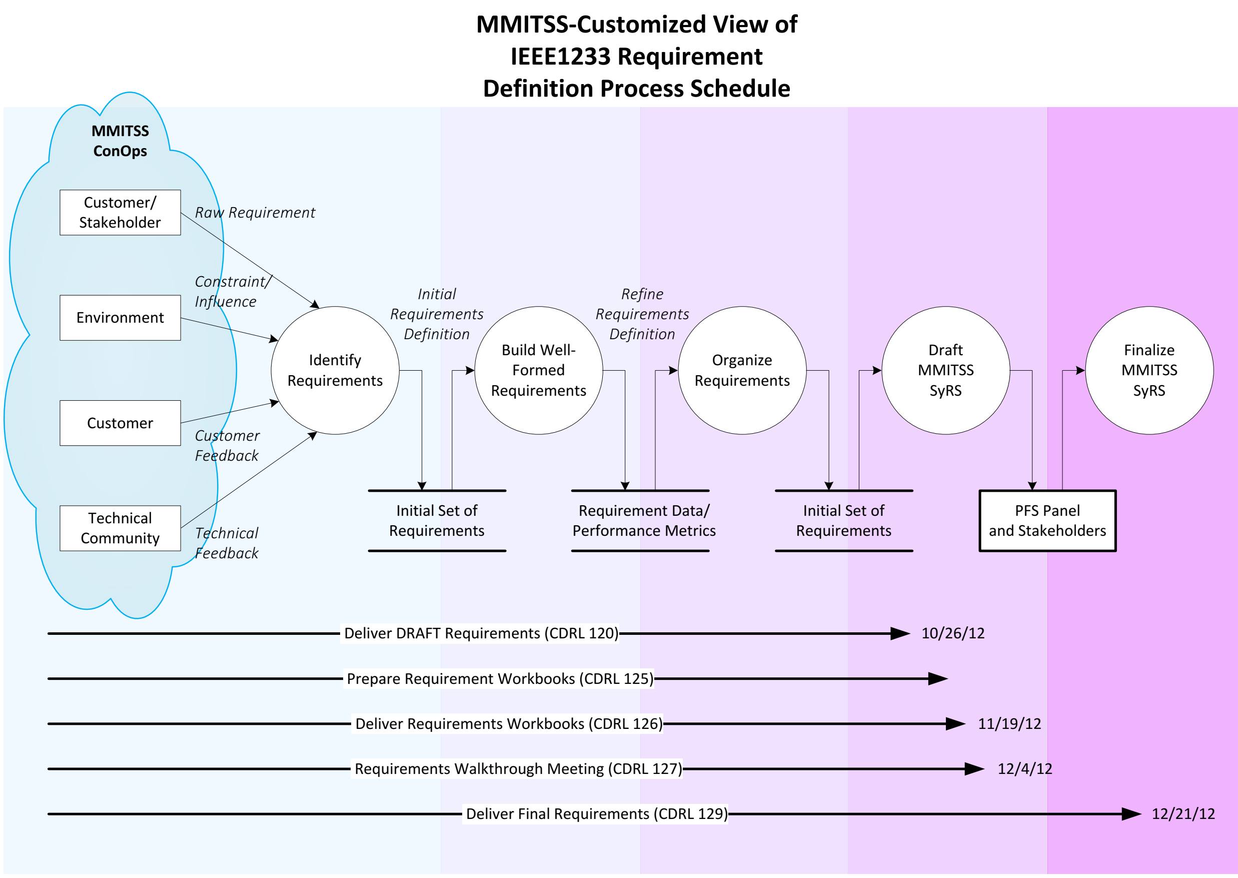


Figure 2. MMITSS-Customized View of IEEE 1233 Requirement Definition Process.

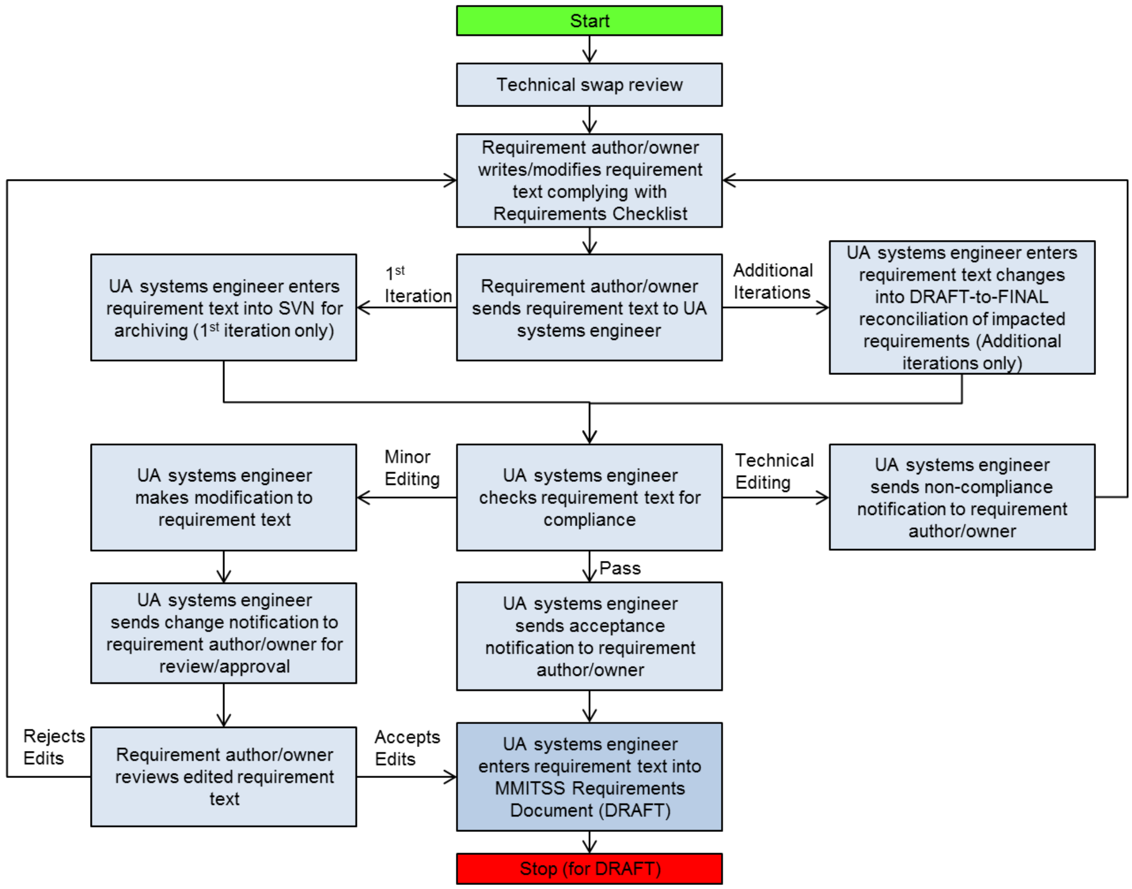


Figure 3. MMITSS Requirement Review Process.

The general categorization of requirements is shown in Figure 4 that highlights the categories of requirements.

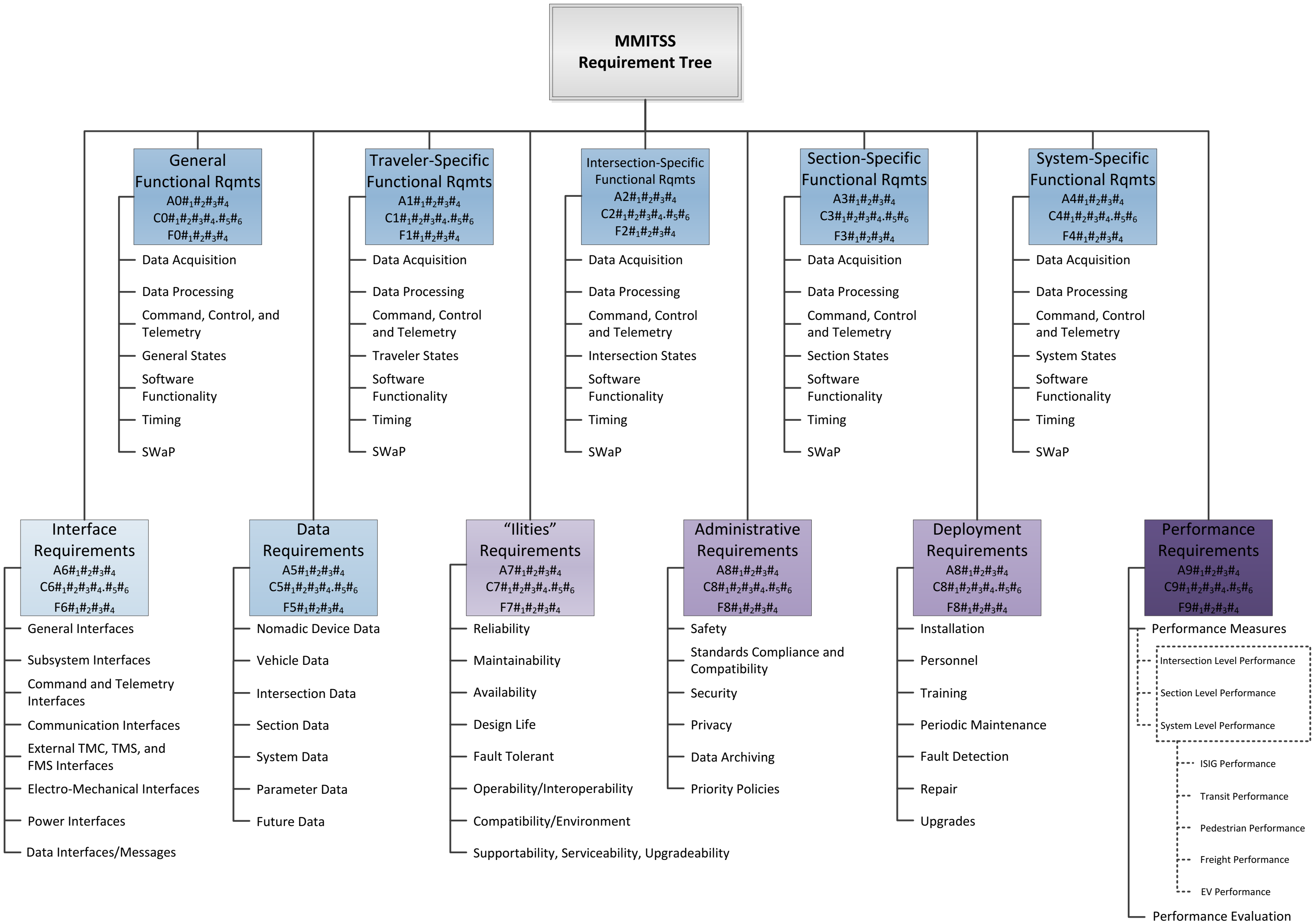


Figure 4. MMITSS Requirements Tree.

As shown in Table 2, the requirement numbering scheme defined a unique requirement identification (RQID).

Table 2. Summary of Requirement ID Nomenclature.

Requirement ID	Explanation / Description
A$\#_1\#_2\#_3\#_4$	System Level Requirement:
B$\#_1\#_2\#_3\#_4$	Research/Prototype
C$\#_1\#_2\#_3\#_4.\#_5\#_6\#_7$	Child/Derived Requirement, where $\#_1\#_2\#_3\#_4$ is identical to the parent requirement, $\#_5$ indicates the allocated area of the child, and $\#_6\#_7$ is the consecutive child numbering value
F$\#_1\#_2\#_3\#_4$ F$\#_1\#_2\#_3\#_4.\#_5\#_6\#_7$	Future ("should" and "will") Future Child Format
$\#_1$	Requirements Structure Diagram: 0=General, 1=Traveler, 2=Intersection, 3=Section; 4=System, 5=Data, 6=Interface, 7="Ilities", 8= Administrative/Deployment, 9=Performance
$\#_2$	0=Multiple, 1=ISIG, 2=Transit, 3=Pedestrian, 4=Freight, 5=EV
$\#_3\#_4$	Consecutive Requirement Number: 01, 02,…
$\#_5$	Child allocation to: 0=All Vehicles, 1=ISIG, 2=Transit, 3=Pedestrian, 4=Freight, 5=EV
$\#_6\#_7$	Consecutive *Child* Requirement Numbering: 01, 02,…

The highest level of a requirement is designated by an RQID that starts with either an "A" or an "F." An RQID starting with an "A" is applicable to requirements that must be implemented or functional within five years of the development and deployment of MMITSS. An RQID that begin with an "F" implies that the requirement must be implemented in the time period associated with the use of "will" or "should" in the requirement text, which is 10 years and 25 years, respectively. An RQID starting with "B" is at the same hierarchical level as "A" and "F" requirements, but represents a derived requirement identified in the Detailed Design.

An "A,", "B", or "F" requirement can be a single requirement or a parent requirement. As shown in Table 2, a child requirement can begin with a "C" for "F" depending on the implied implementation time frame corresponding with the use of "shall" or "will" within the requirement text, respectively. A child requirement will have the same numerical specification for the first four numbers ($\#_1\#_2\#_3\#_4$) as the parent. This is similar to the convention of parents and children having the same last name in many Western cultures. The number specification located to the right of the radix point indicates characteristics of the child.

The MMITSS System Requirements were used, in conjunction with the System Architecture (see Section 4.3.1), to define the High Level and Detailed System Design. Each requirement was assigned to one or more system components (software) as shown in Table 3. The system components were designed to satisfy the associated requirements. The system components are discussed in greater detail in Section 4.3.2.

Table 3. Mapping of System Requirements to System Components (Software).

RQID	Requirement Title *Short Text*	Traceability	Assigned System Component(s)
C1303.301	Certified Nomadic Device Application	13.3.3; *ConOps §4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.3*	AuthorizedSpecalUserService
C2006.001	Track Equipped Vehicles Near Intersection	13.3.1, 13.3.2, 13.3.4, 13.3.5; *11.0, §11.1.1, §11.1.4, §11.2.1, §11.2.2, §11.4.1, §11.5.1*	MRP_EquippedVehicleTracker
C2007.001	Estimate Intersection Expected Time of Arrival of Equipped Vehicles	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.1, §11.2, §11.4, §11.5*	MRP_EquippedVehicleTracker
C2007.202	Estimate Intersection Expected Time of Arrival of Equipped Transit Vehicles	13.3.2; *§11.1, §11.2*	MRP_EquippedVehicleTracker
C2007.404	Estimate Intersection Expected Time of Arrival of Equipped Freight Vehicles	13.3.4; *§11.0.1, §11.1, §11.4*	MRP_EquippedVehicleTracker
C2007.505	Estimate Intersection Expected Time of Arrival of Emergency Vehicles	13.3.5; *§11.1, §11.5*	MRP_EquippedVehicleTracker
C2008.001	Estimate Intersection Expected Time of Departure of Equipped Vehicles	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.1, §11.2, §11.4, §11.5*	MRP_EquippedVehicleTracker
C2008.202	Estimate Intersection Expected Time of Departure of Equipped Transit Vehicles	13.3.2; *§11.2*	MRP_EquippedVehicleTracker
C2008.303	Estimate Intersection Expected Time of Departure of Equipped Non-Motorized Traveler	13.3.3; *§11.3*	MRP_EquippedVehicleTracker
C2008.404	Estimate Intersection Expected Time of Departure of Equipped Freight Vehicles	13.3.4; *§11.4*	MRP_EquippedVehicleTracker
C2008.505	Estimate Intersection Expected Time of Departure of Emergency Vehicles	13.3.5; *§11.5*	MRP_EquippedVehicleTracker
F2019.001	Estimate Required Stopping Distance for Passenger Vehicles	11.1.4 *§5, §8, §11.0.1, §11.1.4*	MRP_EquippedVehicleTracker
F2019.202	Estimate Required Stopping Distance for Transit Vehicles	11.1.4 *§5, §8, §11.0.1, §11.1.4*	MRP_EquippedVehicleTracker
F2019.403	Estimate Required Stopping Distance for Freight Vehicles	11.1.4 *§5, §8, §11.0.1, §11.1.4*	MRP_EquippedVehicleTracker
A2103	Determine Traffic Signal Phase for Tracked Vehicle	13.3.1 (for 11.1.1.3)	MRP_EquippedVehicleTracker MRP_NomadicDeviceTracker
C2004.001	Provide Intersection Signal Phase and Timing Data to Equipped Vehicles	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; *§4.1.3, §5, §8, §9.1, §11.0, §11.4.1, §11.5.1*	MRP_MAP_SPaT_Broadcast
C2005.001	Provide Geometric Intersection Description (GID) Data to Equipped Vehicles	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§4.1.3, §5, §8, §9.1, §11.0, §11.2.1, §11.4.1, §11.5.1*	MRP_MAP_SPaT_Broadcast
C2005.302	Provide Geometric Intersection Description (GID) Data to Equipped Travelers	13.3.3; *4.1.3, §5, §8, §9.1, §11.0, §11.2.1*	MRP_MAP_SPaT_Broadcast

A1002	Vehicle Configuration to Acquire Basic Status Data from Intersection	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1*	MRP_MAP_SPaT_Broadcast, OBE_MAP_SPaT_Receiver
C2006.302	Track Equipped Non-Motorized Traveler Near Intersection	13.3.3; §*11.0, §11.3.2*	MRP_NomadicDeviceTracker
F2006.003	Determine Possible Conflicts In Travel Path	13.3.1, 13.3.2, 13.3.4, 13.3.5; *11.0, §11.1.1, §11.1.4, §11.2.1, §11.2.2, §11.4.1, §11.5.1*	MRP_NomadicDeviceTracker MRP_EquippedVehicleTracker
C2001.302	Acquire Equipped Non-motorized Traveler Status Data	13.3.3; *§8, §11.0, §11.3.2, §11.3.3*	MRP_NomadicDeviceTracker NomadicDeviceTracker
C2013.001	Estimate Intersection Traffic Counts	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2013.002	Estimate Intersection Traffic Count Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2013.003	Estimate Intersection Queue Length	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2013.004	Estimate Intersection Queue Length Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2013.005	Estimate Intersection Delay	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2013.006	Estimate Intersection Delay Variability	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2013.007	Estimate Intersection Throughput	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2013.008	Estimate Intersection Throughput Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11*	MRP_PerformanceObserver
C2015.001	Update Estimates of Intersection Traffic Counts	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver
C2015.002	Update Estimates of Intersection Traffic Count Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver

C2015.003	Update Estimates of Intersection Queue Length	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver
C2015.004	Update Estimates of Intersection Queue Length Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver
C2015.005	Update Estimates of Intersection Delay	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver
C2015.006	Update Estimates of Intersection Delay Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver
C2015.007	Update Estimates of Intersection Throughput	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver
C2015.008	Update Estimates of Intersection Throughput Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_PerformanceObserver
C8101.101	Archive Intersection Level Performance Measures	11.1.2 (13.3.1) *§11.0, §11.1.2, §12.7*	MRP_PerformanceObserver
F2002.001	Acquire Equipped Vehicles Signal Request Data	13.3.1; *§8, §11.1*	MRP_PriorityRequestServer
C2002.202	Acquire Equipped Transit Vehicles Signal Request Data	13.3.2; *§8, §11.0, §11.2*	MRP_PriorityRequestServer
C2002.303	Acquire Equipped Non-Motorized Travelers Signal Request Data	13.3.3; *§8, §11.0, §11.3*	MRP_PriorityRequestServer
C2002.404	Acquire Equipped Freight Vehicles Signal Request Data	13.3.4; *§8, §11.0, §11.4*	MRP_PriorityRequestServer
C2002.505	Acquire Emergency Vehicles Signal Request Data	13.3.5; *§8, §11.0, §11.5*	MRP_PriorityRequestServer
F2003	Acquire Intended Travel Path from Equipped Vehicles and Travelers	13.3.4, 13.3.5; *§11.0, §11.4.1, §11.5.1*	MRP_PriorityRequestServer
C2003.201	Acquire Intended Travel Path from Equipped Transit Vehicles	13.3.2 *§11.2*	MRP_PriorityRequestServer
C2003.302	Acquire Intended Travel Path from Equipped Pedestrian	13.3.3 *§11.3.2, §11.3.3*	MRP_PriorityRequestServer
F2003.303	Acquire Intended Travel Path from Equipped Bicyclists	13.3.3 *§11.3.2, §11.3.3*	MRP_PriorityRequestServer

C2009.001	Provide Signal Status Data to Equipped Vehicles	13.3.2, 13.3.4, 13.3.5; §8, §11.0, §11.2.1 §11.2.3 §11.4.1, §11.4.2, §11.5.1	MRP_PriorityRequestServer
F2010.001	Process Signal Request Message from Equipped Vehicle	13.3.2, 13.3.4, 13.3.5; §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1	MRP_PriorityRequestServer
C2010.202	Process Signal Request Message from Equipped Transit Vehicle	13.3.2; §11.0.2, §11.2.1, §11.2.2, §11.2.3	MRP_PriorityRequestServer
C2010.303	Process Signal Request Message from Equipped Non-Motorized Traveler	13.3.3; §11.3.2, §11.3.3	MRP_PriorityRequestServer
C2010.404	Process Signal Request Message from Equipped Freight Vehicle	13.3.4; §11.0.1, §11.3.4	MRP_PriorityRequestServer
C2010.505	Process Signal Request Message from Emergency Vehicle	13.3.5; §11.0, §11.5	MRP_PriorityRequestServer
C2004.302	Provide Signal Phase and Timing Data to Equipped Non-Motorized Travelers	13.3.3; §4.1.3, §5, §8, §9.1, §11.3;	MRP_SignalStatus_Nomadic
C2009.302	Provide Signal Status Data to Equipped Non-Motorized Travelers	13.3.3; §8, §11.0, §11.3.2	MRP_SignalStatus_Nomadic
A1301	Traveler Configuration to Acquire Basic Status Data from Intersection	13.3.3; §5, §8, §11.1, §11.3	MRP_SignalStatus_Nomadic Nomadic_SignalStatusReciever
A1302	Traveler Configuration to Acquire Signal Status Data from Intersection	13.3.3; §5, §8, §11.1, §11.3	MRP_SignalStatus_Nomadic Nomadic_SignalStatusReciever
C2011.001	Control Signal Actuation for Equipped Vehicles	13.3.1, 13.3.2, 13.3.4, 13.3.5; §11.1.1, §11.1.4, §11.2.1, §11.2.2, §11.4.1, §11.5.1	MRP_TrafficControl
C2011.302	Control Signal Actuation for Equipped Travelers	13.3.3; §11.3.3	MRP_TrafficControl
C2011.303	Call Pedestrian Phase and Interval	13.3.3; §11.3.2, §11.3.3	MRP_TrafficControl
C2011.304	Provide Pedestrian Clearance Extension	13.3.3; §11.3.3	MRP_TrafficControl
C2011.005	Call the Signal Phase Associated with a Tracked Equipped Vehicle	13.3.1; §11.1.2, §11.1.3, §11.1.4	MRP_TrafficControl
C2011.006	Provide Dynamic Passage Interval for Tracked Equipped Vehicles	13.3.1; §11.1.2, §11.1.3, §11.1.4	MRP_TrafficControl
C2012.001	Acquire Intersection Signal Intervals	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; §11	MRP_TrafficControl
C2012.002	Acquire Intersection Active Interval Information	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; §11	MRP_TrafficControl

C2014.001	Match Tracked Vehicle Location With Field Sensor Location	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.2.2, §11.2.3, §11.4.1, §11.5.1*	MRP_TrafficControl
C2101.001	Acquire Intersection Field Sensor Detection – Traffic Counts	13.3.1,13.3.2, 13.3.4, 13.3.5; *§11.0, §11.1.1, §11.1.4, §12.7.1*	MRP_TrafficControl
C2101.002	Acquire Intersection Field Sensor Detection – Vehicle Occupancy	13.3.1,13.3.2, 13.3.4, 13.3.5; *§11.0, §11.1.1, §11.1.4, §12.7.1*	MRP_TrafficControl
C2101.003	Acquire Intersection Field Sensor Detection – Vehicle Detector Status	13.3.1,13.3.2, 13.3.4, 13.3.5; *§11.0, §11.1.1, §11.1.4, §12.7.1*	MRP_TrafficControl
C2101.004	Acquire Intersection Field Sensor Detection – Vehicle Detector Alarms	13.3.1,13.3.2, 13.3.4, 13.3.5; *§11.0, §11.1.1, §11.1.4, §12.7.1*	MRP_TrafficControl
C2101.305	Acquire Intersection Field Sensor Detection – Pedestrian Data Element	13.3.3; *§11.0.2, §11.1.1, §11.3, §12.7.3*	MRP_TrafficControl
A2106	Control Infrastructure Advance Warning Flashers	11.1.4; *§11.0.1, §11.1.4*	MRP_TrafficControl
C1303.302	Authorized Nomadic Device Application	13.3.3; *ConOps §4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.3*	Nomadic_PriorityRequestGenerator AuthorizedSpecalUserService Nomadic_PriorityDataServer
A1303	Nomadic Device Application	13.3.3; *ConOps §4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.3*	NomadicMMITSSApp
C2001.001	Acquire Equipped Vehicle Status Data	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§5, §8, §11.1.1, §11.1.3, §11.2.1, §11.4.1, §11.5.1*	OBE_BSMData_Transmitter MRP_EquippedVehicleTracker
C1003.201	Determine Transit Vehicle Eligibility	Use Case 13.3.2; *§4, §4.1.2, §4.1.4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.2*	OBE_PriorityRequestGenerator

C1003.402	Determine Freight Vehicle Eligibility	Use Case 13.3.4; §4, §4.1.2, §4.1.4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.4	OBE_PriorityRequestGenerator
C1003.503	Determine Emergency Vehicle Eligibility	Use Case 13.3.5; §4, §4.1.2, §4.1.4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.5	OBE_PriorityRequestGenerator
C1004.201	Determine Transit Vehicle Level of Priority	Use Case 13.3.2; §4, §4.1.2, §4.1.4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.2	OBE_PriorityRequestGenerator
C1004.402	Determine Freight Vehicle Level of Priority	Use Case 13.3.4; §4, §4.1.2, §4.1.4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.4	OBE_PriorityRequestGenerator
C1004.503	Determine Emergency Vehicle Level of Priority	Use Case 13.3.5; §4, §4.1.2, §4.1.4, §4.1.5, §5, §9.3.4, §11.0, §11.0.1, §11.0.2, §11.5	OBE_PriorityRequestGenerator
A2501	Acquire Active Response Mode Status of Emergency Vehicle	11.5.1;	OBE_PriorityRequestGenerator MRP_PriorityRequestServer
C3003.001	Identify the Leading and Trailing Vehicles of a Platoon	11.1.2 (13.3.1); §11.0, §5, §4.1.1	Section_Coordinator
C3003.002	Estimate Platoon Size	11.1.2 (13.3.1); §4.1.1	Section_Coordinator
C3003.003	Identify a Platoon of Vehicles	11.1.2 (13.3.1); §4.1.1	Section_Coordinator
C3003.004	Track a Platoon of Vehicles	11.1.2 (13.3.1); §4.1.1	Section_Coordinator
C3003.005	Estimate Platoon Travel Time Between Intersections	11.1.2 (13.3.1); §11.1.2	Section_Coordinator
C3003.006	Estimate Number of Stops of a Platoon in a Section	11.1.2 (13.3.1); §11.1.2	Section_Coordinator
A3004	Identify an Intersection with Inappropriate Offset	11.1.2 (13.3.1); §11.0, §11.1.2	Section_Coordinator
C3005.001	Calculate Appropriate Intersection Offset(s) Based on Queue Discharging Time and Expected Travel Time	11.1.2 (13.3.1); §11.0, §11.1.2	Section_Coordinator
C3005.002	Calculate Intersection Offsets for Multiple Directions of Travel	11.1.2 (13.3.1); §11.0, §11.1.2	Section_Coordinator
A3006	Acquire Basic Vehicle Status Information	11.1.2 (13.3.1); §11.0, §4.1.1	Section_Coordinator
A3101	Control Signal Timing Plans to Accommodate Priority, Coordination, and Congestion Control	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; §11, §11.1.2, §11.4.2, §11.5.2	Section_Coordinator
C3101.001	Command Signal Coordination	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; §11, §11.1.2, §11.4.2, §11.5.2	Section_Coordinator

C3102.001	Determine Section Signal Timing	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; §4.2, §5, §11, §11.1.2, §11.4.2, §11.5.2	Section_Coordinator
C3102.002	Determine Section Signal Coordination Timing	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; §4.2, §5, §11, §11.1.2, §11.4.2, §11.5.2	Section_Coordinator
A3103	Set Intersection Offset(s) at Each Intersection in the Section	13.3.1, 13.3.4, 13.3.5; §11.0, §11.1.2, §11.4.2, §11.5.2	Section_Coordinator
C3002.001	Acquire Intersection Performance Measure Data - Queue Length	13.3.1; §11	Section_PerformanceMonitor
C3002.002	Acquire Intersection Performance Measure Data - Queue Length Variability	13.3.1; §11	Section_PerformanceMonitor
C3002.003	Acquire Intersection Performance Measure Data - Delay	13.3.1; §11	Section_PerformanceMonitor
C3002.004	Acquire Intersection Performance Measure Data – Delay Variability	13.3.1; §11	Section_PerformanceMonitor
C3002.005	Acquire Intersection Performance Measure Data - Throughput	13.3.1; §11	Section_PerformanceMonitor
C3002.006	Acquire Intersection Performance Measure Data – Throughput Variability	13.3.1; §11	Section_PerformanceMonitor
C3002.007	Acquire Intersection Performance Measure Data – Traffic Counts	13.3.1; §11	Section_PerformanceMonitor
C3002.008	Acquire Intersection Performance Measure Data – Traffic Count Variability	13.3.1; §11	Section_PerformanceMonitor
C8101.102	Archive Section Level Performance Measures	11.1.2 (13.3.1) §11.0, §11.1.2, §12.7	Section_PerformanceMonitor
C3001.202	Acquire Active Transit Vehicle Priority Requests in a Section	13.3.2; §11.2	Section_PriorityRequestServer
C3001.403	Acquire Active Freight Vehicle Priority Requests in a Section	13.3.4; §11.4	Section_PriorityRequestServer
C3001.504	Acquire Active Emergency Vehicle Priority Requests in a Section	13.3.5; §11.5	Section_PriorityRequestServer
C4003.201	Acquire Transit Route Information	13.3.2; §11.0, §11.2.3	Section_PriorityRequestServer
C4003.402	Acquire Freight Route Information	13.3.4; §11.0, §11.4.2	Section_PriorityRequestServer
A4004	Store Available Vehicle Route Data	13.3.2, 13.3.4; §11.0, §11.0.1, §11.0.2	Section_PriorityRequestServer
C4004.201	Store Available Transit Route Data	13.3.2; §11.0, §11.0.1, §11.0.2	Section_PriorityRequestServer
C4004.402	Store Available Freight Route Data	13.3.4; §11.0, §11.0.1, §11.0.2	Section_PriorityRequestServer
A4101	Assign MMITSS Intersection ID	13.3.1; §11.1.2, §11.1.3	System_ConfigurationManager
A4102	Assign MMITSS Section ID	13.3.1 (for 11.1.2); §11.1.2, §11.1.3	System_ConfigurationManager
A8001	Support N-Level Priority Policy	13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.5; §11.0, §11.1.1, §11.2, §11.3, §11.4, §11.5	System_N_LevelPriorityConfigurationManager

A8002	Support Fleet Management Systems (FMS) Vehicle Priority Policies	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.0, §11.1.1, §11.2, §11.4, §11.5*	System_N_LevelPriorityConfigurationManager
C8002.402	Support Freight Management Vehicle Priority Policy	13.3.1, 13.3.4 *§11.0, §11.1.1, §11.4*	System_N_LevelPriorityConfigurationManager
C8002.503	Support Emergency Vehicle Management Priority Policy	13.3.1, 13.3.5 *§11.0, §11.1.1, §11.5*	System_N_LevelPriorityConfigurationManager
A4103	Acquire Section Level Performance Measures	13.3.1 (for 11.1.2); *§11.0, §11.1.2; §12.7*	System_PerformanceMonitor
C4103.001	Acquire Section Vehicle Delay	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.0, §11.2.3, §12.7*	System_PerformanceMonitor
C4103.002	Acquire Section Vehicle Delay Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.0, §11.2.3, §12.7*	System_PerformanceMonitor
C4103.003	Acquire Section Vehicle Total Travel Time	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.0, §11.2.3, §12.7*	System_PerformanceMonitor
C4103.004	Acquire Section Vehicle Travel Time Variability	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.0, §11.2.3, §12.7*	System_PerformanceMonitor
C4103.405	Acquire Section Freight Vehicle Travel Time Variability	13.3.4; *§11.0, §12.7*	System_PerformanceMonitor
C4103.006	Section Vehicle Number of Stops	13.3.1, 13.3.2, 13.3.4, 13.3.5; *§11.0, §11.2.3, §12.7*	System_PerformanceMonitor
C4103.007	Acquire Section Vehicle Throughput	13.3.1; *§11.0, §12.7*	System_PerformanceMonitor
C8101.103	Archive System Level Performance Measures	11.1.2 (13.3.1) *§11.0, §11.1.2, §12.7*	System_PerformanceMonitor

4.3 System Design and Architecture

4.3.1 System Architecture

The Physical MMITSS architecture is shown in Figure 5 as a UML Deployment Diagram. In the Unified Modeling Language (UML), nodes are shown as 3D blocks and represent physical devices that have a processor (at least one), memory, and physical interfaces (e.g. Ethernet, RS-232, or wireless – 3G/4G, 5.9 GHz DSRC, or other such as CAN-bus). The basic architecture is applicable at each MMITSS controlled intersection. In a system that consists of several intersections, each intersection would have an RSE Radio and traffic control equipment as shown in the deployment diagram. There would generally be only one instance of the Traffic Management System, but there could be multiple instances of the Fleet Management System (e.g. Transit Fleet, commercial truck fleet, emergency vehicles).

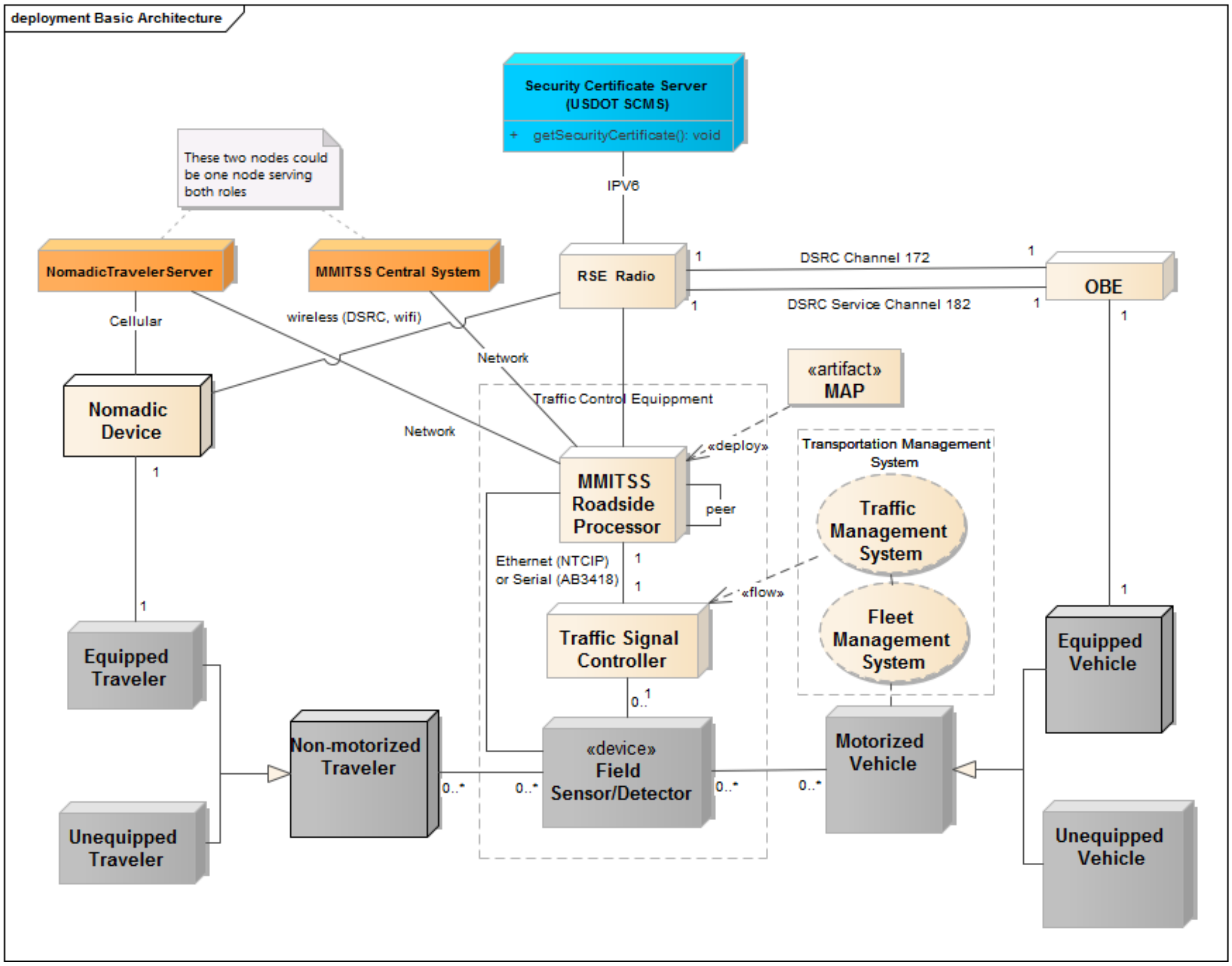


Figure 5. MMITSS Physical Architecture.

The nodes have been shaded such that the light colored nodes are part of the connected vehicle system. Traffic Management and Fleet Management systems (or nodes that can be modified or assigned MMITSS responsibilities) and the gray colored nodes represent the vehicles and travelers. The Security Certificate Server (e.g. USDOT Security Credential Management System – SCMS) is interfaced to the RSE using IPv6 and is used to provide security certificates to trusted OBEs. The orange colored nodes are the MMITSS Central System and Nomadic Traveler Server as described below. These two nodes may be realized as a single node for the test bed implementations.

In this view of the system, there are two types of travelers – motorized vehicles and non-motorized travelers. Motorized vehicles consist of passenger vehicles, trucks, transit vehicles, emergency vehicles, and motorcycles. This type of traveler includes any vehicle that must be licensed to operate on the public roadway. Non-motorized travelers include pedestrians, bicyclists, and other modes such as equestrians that are not required to be licensed to operate on the public roadway. These travelers are either unequipped or equipped, meaning that they have some type of OBE (On-Board Equipment) or nomadic device that is connected vehicle (or MMITSS) aware and can operate as part of the traffic control system.

Motorized vehicles can be part of a fleet management system such as a transit management system, commercial freight management system, emergency vehicle dispatch system, or taxi dispatch, which is shown as a UML collaboration (oval in Figure 5) meaning that a collection of entities work together to perform the traffic management functions, but there may be many different systems involved in this collaboration.

The infrastructure based traffic signal control equipment consists of the traffic signal controller, field sensors/detectors, an optional MMITSS Roadside Processor (MRP) and the Roadside Equipment (RSE). The RSE Radio is the hardware device that is responsible for managing all of the 5.9 GHz DSRC communications between the vehicles and the infrastructure. Two channels are utilized for communications between the RSE and the vehicle on-board equipment (OBE). Channel 172 is called the high-availability, low-latency channel and is used by vehicle for broadcasting basic safety messages (BSM) and by the infrastructure (RSE) for broadcasting the map message (MAP) and the signal phase and timing message (SPaT). Channel 182 is used for exchanging the signal request messages (SRM) and signal status messages (SSM).

The OBE is a hardware device deployed on the vehicle. MMITSS will be developed and tested using Savari MobilWave units (called aftermarket safety devices - ASD) for the OBE. These units are general purpose and provide a powerful and flexible platform for development and testing. The California testbed also utilized Arada OBE. The OBE devices are interoperable since the messages exchanged with the RSE are based on the SAE J2735 (2009) standard.

There are two traffic signal controllers models that will be utilized in the field installations: Econolite ASC/3 (or Cobalt) controllers are used in Arizona and Type 2070 controllers are used in California. Each of these controllers offers different signal timing logic (software) and require different communications interfaces. The Econolite ASC/3 and the Cobalt controllers are based on NEMA standards and support NTCIP over Ethernet communications. The Caltrans Type 2070 controllers are based on a Caltrans standard and support AB3418 over serial RS-232 communications. Both networks utilize loop detectors for vehicle detection. The MMITSS Roadside Processor (MRP) is a Linux based general-purpose computer (see Section 4.6.1.1 below for details) [Note: this devices was specified for the California test bed].

The traffic signal controllers can be part of a larger traffic management system that controls and organizes groups (sections) of signals. The larger traffic management system is shown as an UML collaboration in Figure 5. The RSE is a general communications processing node that coordinates messages from the various modes of travelers. The MMITSS Roadside Processor (MRP) is a general purpose computer that can host the core intersection level infrastructure applications for MMITSS. The RSE contains (deploys) the MAP artifact, which is the digital description of the intersection geometry and associated traffic control definitions.

Both motorized and non-motorized travelers can be detected by the Field Sensor/Detector node at the intersections using a variety of detection technologies, including inductive loop detectors, video detection, microwave, radar, pedestrian push button, etc. The detection system at an intersection provides information to the traffic signal controller that stimulates the control algorithms. For example, a vehicle that triggers a detector will call a signal control phase for service or extension. A pedestrian may activate a pedestrian push button to request the traffic signal pedestrian interval associated with a crosswalk movement. The direct communications path from the field sensor to the MRP allows the communication of information from the sensor. This information might include vehicle count from presence detectors or possibly vehicle trajectories from radar based sensors in the future. [Implementation Note: No direct sensor data was used in the MMITSS implementation.]

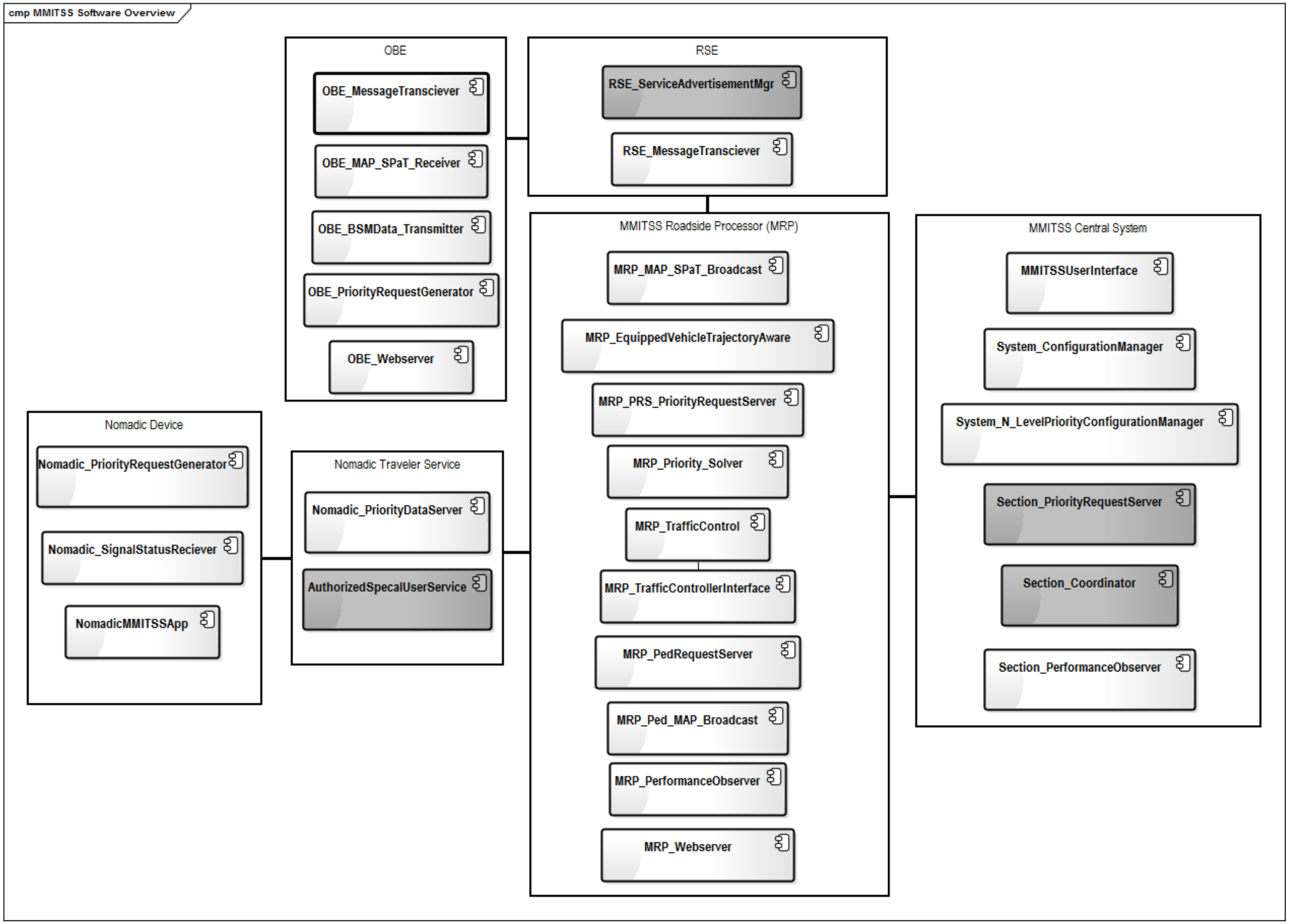


Figure 6. MMITSS System (Software) Architecture.

4.3.2 System Design (Software)

The MMITSS System Design (software) is show as a UML Component Diagram in Figure 6. There are Boundaries (boxes) around groups of components that relate to the nodes in the Deployment Model (Figure 5). Each component is a software process (program) that runs and communicates with other local (same node) or remote (different node) components.

This design was implemented for the Arizona Test bed, but an alternative software architecture was selected for the California Test Bed and is discussed in Section 4.6

Table 4. Summaries of the Basic Component Responsibilities.

System Component	Basic Responsibility
RSE_ServiceAdvertisementMgr[1]	Advertise available services, e.g. MMITSS
RSE_MessageTransceiver	Transmit and receive WAVE messages from the roadside
OBE_MessageTransceiver	Transmit and receive WAVE messages from the vehicle (identical to the RSE_MessageTransceiver but implemented on the OBE)
OBE_BSMData_Transmitter	Send vehicle Basic Safety Messages
OBE_MAP_SPaT_Receiver	Receive MAP and SPaT data on a vehicle
MRP_MAP_SPaT_Broadcast	Broadcast MAP and SPaT data from the intersection
MRP_EquippedVehicleTrajectoryAware	Be aware of the trajectories of equipped vehicles that are present in DSRC radio range (e.g. 300 meters)
System_ConfigurationManager	Set up sections (collections of traffic signals) and system level functions
MMITSSUserInterface	Display system information and status
System_N_LevelPriorityConfigurationManager	Configuration manager for N-Level priority policy
OBE_Webserver	Tool for visualization of data available on a vehicle. The data includes priority related information – submitted requests, pending requests, active requests, and MAP and SPaT information. Web pages were used for the implementation to allow flexibility in the design.
Nomadic_SignalStatusReciever	Nomadic device application that receives traffic signal status data
NomadicMMITSSApp	The application that provides MMITSS capabilities on the nomadic device. This application must be downloadable from the appropriate "store"
OBE_PriorityRequestGenerator	The vehicle based component that is responsible for sending a priority request
MRP_TrafficControllerInterface	The component responsible for communications with the traffic signal controller (AZ – NTCIP, CA –

	AB3418)
MRP_TrafficControl	The component responsible for traffic control logic – phase calls, phase extension, dilemma zone (AZ – NTCIP and adaptive control, CA – Caltrans software using AB3418 and coordinated actuated control)
AuthorizedSpecialUserService	A service to support the authorization of nomadic devices and provide authorization for special (e.g. disabled) travelers
Nomadic_PriorityDataServer	A service (cloud based) component that relays data from the MMITSS to the nomadic device.
Nomadic_PriorityRequestGenerator	The nomadic device component responsible for sending Ped Requests (either ped call or special service request)
MRP_PriorityRequestServer[2]	This component is responsible for managing all priority requests received at each intersection, including selection of the appropriate service strategy. The AZ and CA testbeds support different priority strategies so this component will be custom for each network.
MRP_PerformanceObserver	Acquires data and estimates intersection level performance measures
Section_Coordinator	Responsible for section level traffic control including coordination
Section_PerformanceObserver	Acquires intersection level data and estimates section level performance measures
Section_PriorityRequestServer	Responsible for section level priority control strategies
System_PerformanceObserver	Acquires section level data and estimates system level performance measures
MRP_PedRequestServer	Responsible for receiving and serving ped requests
MRP_PedMAP_Broadcast	Responsible for sending the Ped MAP data to the Nomadic device
MRP_PrioritySolver (AZ)	Responsible for solving the optimal scheduling problem for sets of active priority requests (used in the Arizona Test Bed)
MRP_Webserver	Tool for visualization of data available at the intersection The data includes priority related information – active requests, pending requests, MAP and SPaT information, and Performance Measures at the Intersection level. Web pages were used for the implementation to allow flexibility in the design.
System_PerformanceObserver	Acquires section level data and estimates system level performance measures

1

4.4 Architecture Differences in the Arizona and California Prototypes

The primary difference between the Arizona MMITSS and the California MMITSS is summarized in Table 5. The different implementations of MMITS allows flexibility for the deployment in other sites that might want to utilize the different traffic controller models, traffic control strategies, and software architectures.

Table 5. Comparison of Arizona MMITSS and California MMITSS

	Arizona Test Bed	California Test Bed
Test Bed Traffic	Suburban intersections (6)	Major arterial in congested intersections (11)
Traffic Controller and Software	NEMA Econolite ACS/3 controller and software using the NTCIP 1202 standard for communications	Model 2070 controller and Caltrans TSCP software using AB3418 protocol for communications
System Architecture	MRP integrated with RSE; Peer-to-peer data communication between software components	Standalone MRP; Data manager for data synchronization and distribution
Desired Operation Mode	Adaptive signal control	Coordinated-actuated signal control
Traffic Control Algorithms	New priority control algorithms that use connected vehicle data integrated with coordination and actuated or adaptive signal control	Traffic control algorithms based on the Caltrans 2070 firmware with connected vehicle data used for collecting priority requests and traffic state estimation.

The Arizona MMITSS directly embodies connected vehicle technology and data while utilizing a combination of new priority control algorithms based on the NCHRP 3-66 research (http://apps.trb.org/cmsfeed/TRBNetProjectDisplay.asp?ProjectID=820) and the RHODES traffic adaptive signal control system (Mirchandani & Head, 2001). The Arizona MMITSS traffic control algorithms are specifically based on the availability of connected vehicle data while realizing that the market penetration of equipped passenger vehicles will initially be low, but that special fleets of connected vehicles, such as EV, transit, and trucks, will likely be the first benefactors of this new technology.

The California MMITSS provides an enhancement for the existing coordinated-actuated control system by integrating the connected vehicles and travelers with the traffic and priority control.

4.5 Design and Implementation of Arizona MMITSS Prototype

The Arizona MMITSS implementation followed the high level design described in Section 4.3. The deployment utilized the processing capabilities of the Savari RSE and didn't utilize a separate roadside processor (MRP). Since the software was designed as a collection of software components that share data as needed, e.g. a peer-to-peer data architecture, the different software components can be built and deployed on any set of nodes (e.g. a processor with memory) including the Savari RSE or a separate MRP. The details of the Arizona MMITSS prototype can be found in the Detailed Design Report (http://www.cts.virginia.edu/wp-content/uploads/2014/05/Task4._SystemDesign_3_Revised_v.2.pdf) and in the US DOT Open Source Application Development Portal (http://www.itsforge.net/).

Aside from the main MMITSS development process, Econolite developed a Connected Vehicle Processor Board that is a stand alone Linux computer than can serve the role of the MPR. This CVPB is field hardened so that it can survive the hot temperatures during the Arizona summers and it resides inside of the Econolite Cobalt controller. The Arizona MMITSS prototype was built for the Econolite CVPB and successfully tested at one intersection in the Arizona Test Bed.

4.5.1 Overview of the Arizona MMITSS Prototype

The Arizona MMITSS (AZ MMITSS) prototype was developed with the philosophy that connected vehicle data, both vehicle basic safety messages and priority request messages, would provide significant additional information that could be used to develop a new generation of traffic signal control algorithms. Goodall (2013) demonstrated that connected vehicle data could improve traffic signal control performance. Together, with recent advances in adaptive traffic signal control, such as OPAC (Gartner, 1983), RHODES (Mirchandani and Head, 2001), and ACS Lite (Gettman, et. al., 2006) and the results of the NCHRP 3-66 project (Urbanik, et. al, 2003), there has been significant advancement in the development of traffic control algorithms for both general vehicles (adaptive signal control) and for special classes of vehicle (priority control). In addition, it was recognized that the market penetration rate of connected vehicles would be low for several years, but priority eligible vehicle fleets could be good candidates for early adoption projects – such as transit priority corridors, freight priority corridors, and emergency vehicle systems. If MMITSS could provide effective priority control with low market penetration rates for personal vehicles, then the entire traffic management system could benefit as the market penetration increases.

One aspect of this strategy was the recognition that actuated traffic signal control would continue to be the standard at most traffic signals, so AZ MMITSS should be able to integrate innovative priority control with actuated signal control. Coordination is also a critical traffic control strategy and AZ MMITSS should be able to integrate priority control with coordination and actuation. As such, coordination was considered to be a form of priority which could be addressed in a unified priority control framework.

An important concept adopted in the AZ MMITSS prototype was the concept of multiple levels of importance for different modes in traffic control. An operating agency should have the ability to determine which mode of priority should be the most important and which should be the least important when serving multiple requests at one time. For example, in one corridor, coordination might be the most important consideration during the AM and PM peak commute times or during an incident on a parallel freeway. In the off-peak times, the priority might favor transit vehicles in an urban shopping/school area or might favor freight in a commercial warehouse/factory area. The ability to establish a “priority policy” would allow the operating agency to have a powerful tool for traffic management.

Finally, the recent development and agency adoption of the NTCIP communication standards has provided a favorable path for deployment of AZ MMITSS. NTCIP compliant traffic signal controllers can easily provide the current signal timing plan data to AZ MMITSS and can interface with the AZ MMITSS algorithms. The standards provide a well-defined and common interface that is now widely used across the United States. Most major traffic signal controller manufacturers support NTCIP and NTCIP provides the interfaces necessary for working with the controllers.

In the following sections, an overview of the MMITSS system is presented. The goal in this section is to familiarize the reader with the design and algorithm concepts used in the AZ MMITSS prototype. Details of the software implementation can be found in the MMITSS Detailed Design document. The following section is divided into two main subsections. The first presents the AZ MMITSS architecture and describes the software components that provide the platform for the AZ MMITSS traffic control algorithms. The second section provides an introduction to the adaptive signal control, the priority control, and the performance observer components. The AZ MMITSS Central System was not fully developed and tested in the Phase II prototype effort, but represent key architecture and approach components that would be required in any deployment.

4.5.2 Architecture of AZ MMITSS

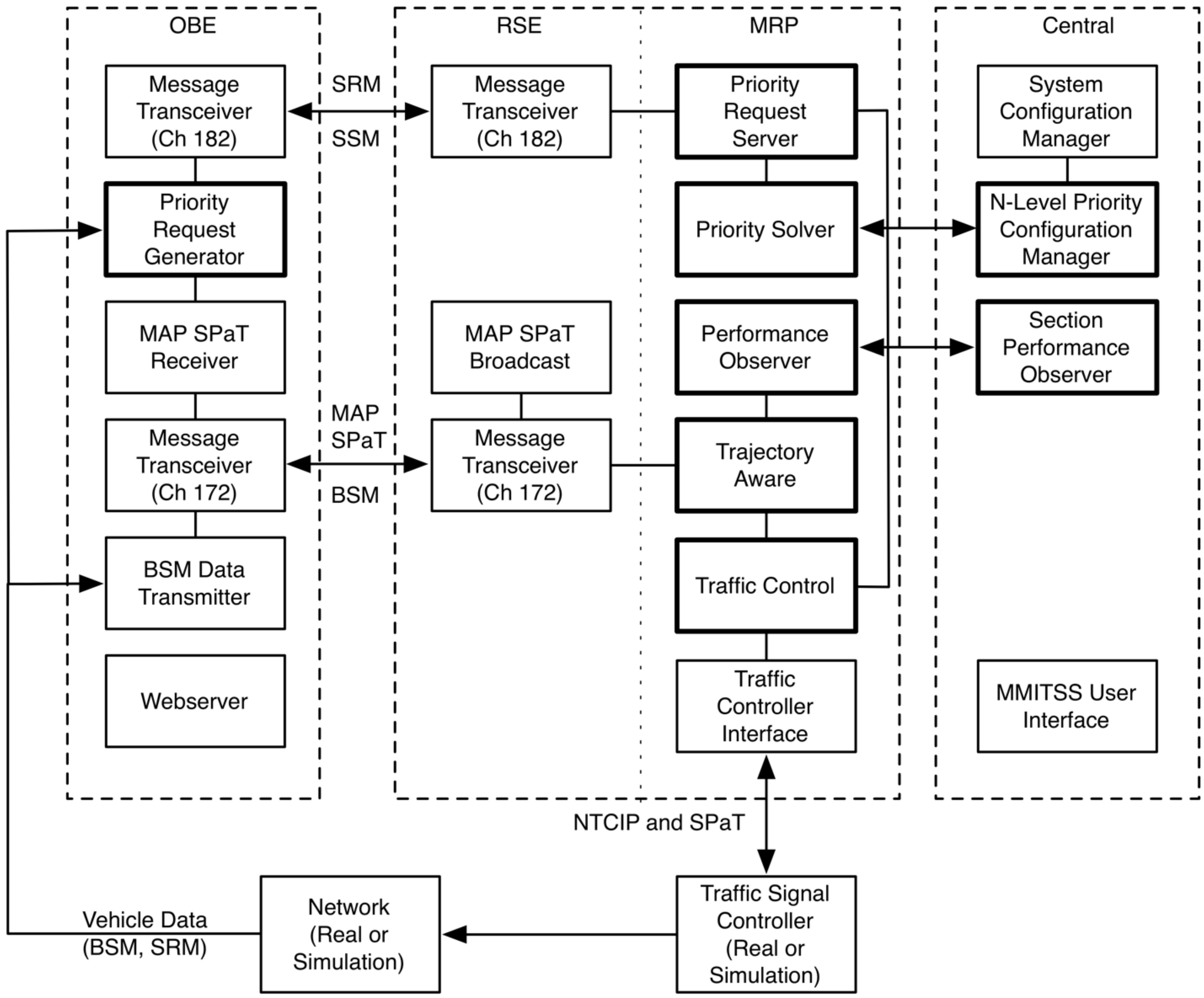


Figure 7. Architecture of AZ MMITSS System

The AZ MMITSS architecture is shown in Figure 7. The architecture was designed based on a component based, or distributed, software design where the components use peer-to-peer communications (sockets) to share information. Figure 7 shows three key MMITSS architecture elements: the on-board Equipment (OBE), roadside processor and equipment (RSE and MRP), and the Central system. Software components are deployed on each of the key architecture elements. There are two kinds of components: MMTISS traffic control algorithms (shown with think black borders) and MMTISS interface and general connected vehicle components. The interface components allow the MMITSS system to acquire data and actuate controls, as well as to interface to the two DSRC communication channels that are used. The MMITSS traffic control components implement the new control algorithms developed to use the connected vehicle data.

4.5.3 Architecture Components

The architecture includes six software components that run on the OBE, ten software components that run on the RSE or on the MMITSS Roadside Processor (MRP), and four that run on the central system (a Windows based PC). In the current implementation, all RSE and MRP components are run on the RSE (Savari Streetwave), but a prototype MRP (developed by Econolite during the development of their Connected Vehicle Co-Processor (CVCP) product) was successfully tested with the AZ MMITSS algorithms. The CA MMITSS system uses a dedicated MRP (see Section 4.6.1.1 below). The peer-to-peer socket interfaces allow the components to be installed on any processor in the subnetwork and configured to communicate with the desired peer component(s). For example, all the algorithm components (thick black lines in Figure 7) could be run on a separate processor, or some could be on the RSE and some on the MRP, or all on the RSE – as was tested in the Arizona Field Test.

During the detailed design process, several software architectures were investigated. The distributed architecture (as opposed to the central data manager architecture described in Section XX that was used in the CA MMITSS system) was selected for the AZ MMITSS system. The critical difference for the AZ MMITSS system was the desire (need) to the software components to operate asynchronously so that streams of data, including BSMs from connected vehicles and SPaT data from the traffic signal controller, could be continuously processed while the other components, such as the Priority Request Server, Priority Solver, Traffic Control, could be allowed to process. This architecture can be implemented on the RSE which does not support threading and in a distributed fashion on one or more MRPs.

Applications, including adaptive signal control and dilemma zone protection, utilize only BSMs while coordination and signal priority utilize SPaT, MAP, SRM and SSM. Two DSRC channels 172 and 182 are used to send/receive different types of messages. Channel 172 is the safety channel, which is used to transmit safety related messages (e.g., BSM, SPaT and MAP). Channel 182 is one of the available service channels, which is used to transmit application specific messages (e.g., SRM and SSM for priority control).

This architecture has been implemented in both simulation (VISSIM 6.0 with the Econolite ASC/3 software-in-the-loop (SIL) signal controller) and in the field using NTCIP compliant signal controllers. The simulation implementation is described in Section 6.1.1.

4.5.3.1 Message Transceiver (OBE and RSE)

The Message Transceiver application is configurable to utilize either of the two 5.9GHz radios. One is configured to use channel 172 and the other to use channel 182. Channel 172 is used by the OBE to broadcast basic safety messages (BSM) and to receive MAP and SPaT messages that are broadcast by the RSE (MAP SPaT Broadcast component). Channel 182 is used to broadcast and receive Signal Request Messages (SRM) and Signal Status Messages (SSM). The Message Transceiver interfaces to the native IEEE 1609 WAVE stack on the Savari RSE, but could also be implemented on other roadside units.

4.5.3.2 MAP SPaT Receiver (OBE)

The MAP SPaT Receiver is deployed on the OBE and is responsible for receiving and unpacking the J2735 (200(9) MAP and SPaT messages and making the data available to the Priority Request Generator. The MAP SPaT receiver can supply MAP and SPaT data to other components on the OBE if desired. The MAP data is stored in a file with labeled with the name of the intersection for instances where there may be more than one active MAP received.

4.5.3.3 BSM Data Transmitter (OBE)

The BSM Data Transmitter is responsible for building the J2735 (2009) Basic Safety Message (BSM). For the MMITSS prototype, the BSM contains a temporary identification number (ID, the vehicle's GPS position, speed (from GPS) and the vehicle type, which is read from a file and can be changed for different vehicle types. The message is built 10 times per second and sent to the Message Transceiver for broadcast on Channel 172. Currently, the BSM Data Transmitter does not read vehicle data from the vehicle's CAN bus and it does not do any position correction. The position accuracy was sufficient for field testing in the Arizona Connected Vehicle Test Bed where lane level positions were generally received.

4.5.3.4 Traffic Controller Interface (RSE, MRP)

The Traffic Control Interface component is responsible for all communications wth the traffic signal controller. There are three main communication interactions between the traffic signal controller and the Traffic Controller Interface component: 1) register and receive SPaT data, 2) query the controller for configuration data, and 3) command the controller to implement the desired traffic control schedule. All communications with the traffic controller (Econolite ASC/3 or Cobalt) are through the local Ethernet network. The Econolite ASC/3 controller is configured with the IP address and port where it can stream SPaT messages. The SPaT messages are defined according to the Battelle (2012) Signal Phase and Timing specification that was developed for FHWA. SPaT data is received, unpacked and made available to the Traffic Control and Priority Request Server components 10 times per second.

The controller configuration data is queried when it is started and provides the Phase Timing Interval Data defined in Table 6. [Note: the frequency of this query can be changed to allow MMITSS to be aware of changes to the controller configuration by a traffic technician using the controller front panel or by a central traffic management system.] In addition, Detector Parameter configuration is read on start up, and Detector Data (Phase Call, Occupancy, and Volume) are read once per second. This data is made available to Traffic Control, Priority Request Server, and the Performance Observer components as input for their functions. All controller configuration and detector data is queried using NTCIP 1202 objects (2005).

The commands from AZ MMITSS to the traffic signal controller are implemented using NTCIP Phase Control Objects (Table 6). AZ MMITSS utilizes the controller configuration together with the data from vehicle detectors, connected vehicle basic safety messages and signal request messages to determine an optimal signal timing schedule that consists of the phase sequence and durations. The Traffic Control Interface component monitors the controller status and sends the appropriate NTCIP Phase Control command to the controller to implement the signal timing schedule. The optimal signal timing schedule is updated by the MMITSS Traffic Control and Priority Request Server components as new information is available. As the schedule is updated the Traffic Control Interface component issues the commands to the traffic signal controller.

Table 6 Controller's Configuration Data

Data Object	Parameters	NTCIP 1202 Protocol Objects	Usage
PhaseTimingIntervals	PhaseNumber PhaseWalk PhasePedestrianClear PhaseMinimumGreen PhasePassage PhaseMaximum PhaseYellowChange PhaseRedClear	Phase Parameters	Input to *MRP_TrafficControl* & *MRP_PRS_PriorityServer* for phase allocation
DetectorParameters	DetectorID DetectorType DetectorCallPhase NTCIPOccupancy NTCIPVolume	Detector Parameters	Input to *MRP_PerformanceObser* for performance estimation *MRP_TrafficControl* for phase allocation
PhaseControl	PhaseOmit PhaseHold PhaseForceOff VehicleCall PedCall	Phase Control	Output from MRP_TrafficControlInterface to the traffic signal controller

4.5.3.5 Webserver

A webserver is used on the OBE to provide a simple and flexible visualization of the status of MMITSS during field testing. Figure 8 shows an example of the MMITSS visualization. The webpage shows the current signal status by phase and pedestrian interval as well as the status of the MAP and the Priority Request that has been broadcast. It also shows two priority requests that were received by the Priority Request Server and acknowledged through the [revised] Signal Status Message. The emergency vehicle request is overriding the transit vehicle request – as indicated by the red 'x' in the Active column of the table. The in-lane and out-lane, requested phase(s), estimated arrival time, time of desired service, estimated departure time, and vehicle status is shown. All of this data is available through the [revised] J2735 Signal Status Message (SSM).

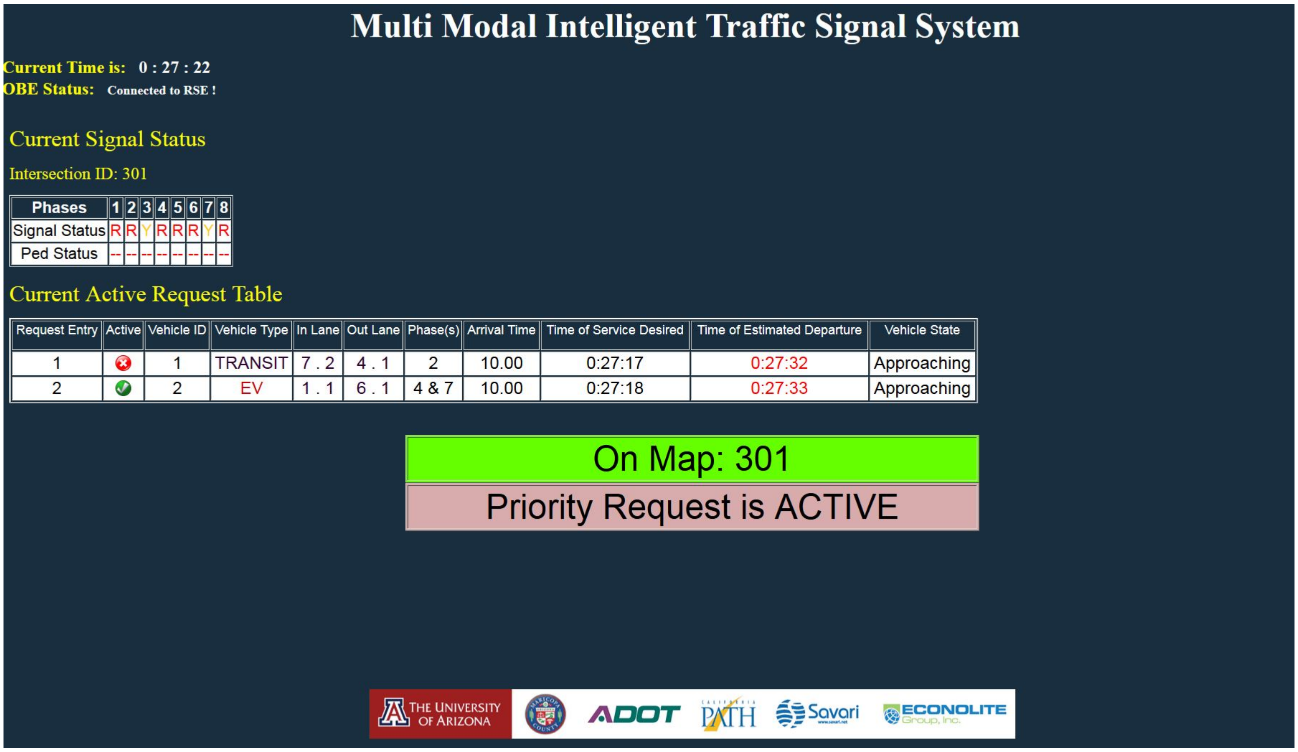


Figure 8 Example of the web page based user interface

4.5.4 Traffic Control Components

4.5.4.1 Priority Request Generator (OBE)

The Priority Request Generator (PRG) is deployed on the OBE. The PRG embodies the vehicle based logic that determines when to send a priority request and the contents of the request message. In the AZ MMITSS prototype it is assumed that any priority eligible vehicle, e.g. emergency vehicles, transit vehicles, and trucks, can send a request when they are approaching an equipped intersection. This assumption would require additional development in a deployment. For example, an emergency vehicle would only request EV priority when the lights and sirens are in use. A transit vehicle might only send a request when it was behind schedule with significant occupancy.

The Priority Request Generator (PRG) checks to see if the MAP SPaT Receiver has new map data. Once a MAP, or several MAPs, are received the PRG determines which MAP describes the intersection that the vehicle is approaching and which lane/approach the vehicle is located. Once this is determined, the PRG will compute the travel time to the stop bar (from the MAP data and the current position) and will build and send the Signal Request Message. The PRG will wait for a Signal Status Message (SSM) to confirm receipt of the request. If no SSM is received, or an SSM is received and the vehicle's request isn't in the table of active requests, the PRG will update the SRM and send it again. If the vehicle speed changes significantly, the estimated time of arrival is updated and an updated request is sent. Once the vehicle crosses the stop bar, a cancel request is sent. Once a SSM is received showing the canceled request is no longer active, the PRG will check any available MAPs or wait for a new MAP to be received.

4.5.4.2 Trajectory Aware (RSE, MRP)

The Equipped Vehicle Trajectory Aware component is responsible for collecting data from each of the connected vehicles that is broadcasting Basic Safety Messages (BSM) when in the range of an RSE.

Each BSM received by the Message Transceiver (Channel 172) is forwarded to the Trajectory Aware component. The Trajectory Aware component checks to see if the vehicle is within the geo-fence area created by the MAP GPS waypoints and creates a record for each vehicle's temporary ID that is received and within the geo-fence. Checking the geo-fence location of each vehicle allow the system to filter out vehicles that are in nearby parking lots or other non-traffic controller facilities.

Each record is time stamped and contains the position, speed, vehicle type information, traffic control phases, and estimated time of arrival (ETA) at the stop bar (assuming current speed). If a vehicle changes its temporary ID, a new record will be created for the new vehicle ID. Vehicle data is retained for 5 minutes after the vehicle leaves the RSE range or after it changes its temporary ID. This data is deleted as part of the privacy assurance requirements in the design. The collection of all vehicles that are sending BSM is available to any other AZ MMITSS component that requests data about the location, and trajectory history, if each connected vehicles. Trajectory Aware provides data for Traffic Control and the Performance Observer.

4.5.4.3 Traffic Control (RSE, MRP)

The Traffic Control component is responsible for the allocation of green time to the traffic control phases based on the information available in Trajectory Aware and from vehicle detector data from the Traffic Control Interface component. Traffic Control was developed to provide traffic adaptive control using concepts from the COP algorithm (Sen and Head, 1997) that was developed as part of the RHODES (Mirchandani and Head, 2001) and to provide dilemma zone protection. The AZ MMITSS Traffic Control component is based on three algorithms: 1) Estimation of the location of unequipped vehicles (called EVLS), 2) a phase allocation algorithm that determines the duration of the green phases based on the data from trajectory aware and the estimated locations of unequipped vehicles, and 3) a dilemma zone protection algorithm.

EVLS Algorithm

Due to the low penetration rate of the connected vehicles, an algorithm that estimates the states of unequipped vehicle (EVLS) based on connected vehicle data is developed to construct a complete arrival table for the phase allocation algorithm. The road segment before the intersection is divided into three regions: queuing region, slow-down region and free-flow region as shown in Figure 9. Different traffic flow models are applied to each region to estimate unequipped vehicle states. Shockwave theory is applied in queuing region to estimate the queue length. In slow-down region, Wiedemann's car following model is applied to insert unequipped vehicles and calculate their location and speed. In free-flow region, vehicles are randomly inserted on the roadway based on market penetration rate.

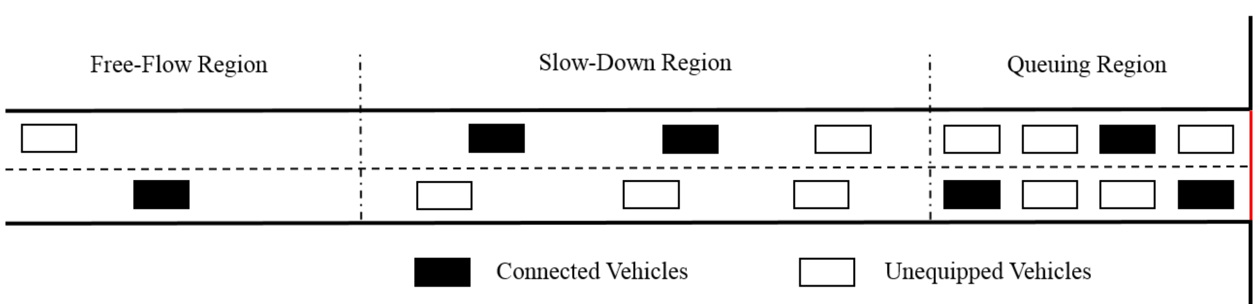


Figure 9. EVLS Algorithm Illustration

A trajectory-based arrival table is constructed based on ETAs and traffic control phases of all vehicles after executing EVLS algorithm. After Traffic Control computes the trajectory-based arrival table, it will combine the data with detector data that is acquired from the Traffic Controller Interface component to create a modified arrival table that will be the input for phase allocation algorithm. Figure 10 shows the structure of the arrival table. The arrival table is a two dimensional matrix with time and phase respectively. The value of each cell is the number of vehicles that will arrive at the stop bar at time point t requesting phase p.

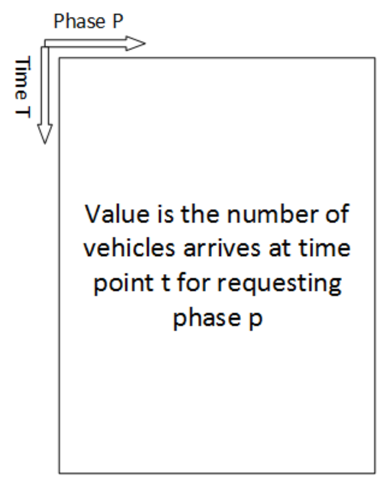


Figure 10. Arrival Table Structure

The Phase Allocation Algorithm

The phase allocation algorithm provides the adaptive control capabilities that the stakeholders desired. It is based on an improvement of controlled optimization of phases algorithm (COP) (Sen and Head, 1997) which is used in the RHODES adaptive traffic control system (Mirchandani and Head, 2001). The algorithm is based on dynamic programming that can result in optimal phase sequences and timing considering a variety of performance indices such as delay, stops and queue length. The algorithm applies a rolling-horizon approach where the optimization problem is solved at the end of each stage of implementation in order to include more recent arrival data.

The original COP algorithm was based on a sequence of stages, e.g. A, B, C, D, where a stage could represent phases 1 and 5 (denoted A), or phases 2 and 6 (denoted B), but it didn't support flexible, or dual ring, phase sequences. The AZ MMITSS phase allocation algorithm applies a two-level optimization scheme based on the dual ring controller as shown in Figure 11. At the upper level, a dynamic programming framework similar to COP is applied to each barrier group (e.g. phases 1 and 2 in ring 1 and phases 5 and 6 in ring 2) The sequence of barrier groups is assumed to be fixed, but the order of phases within each ring in each barrier group can vary. Initially, two stages of barriers with a total time horizon of 120 seconds are planned, but this could be modified to include three or more barrier groups depending on the time horizon and availability of data.

First, the minimum and maximum allowed barrier group lengths are calculated according to variety of signal. Those parameters include the minimum green, maximum green, yellow interval and red clearance time of each phase. For the first barrier group, the elapsed time since the start of green of the current and previously served phases is taken into consideration. If one of the barrier group phases has no demand, then that phase can be skipped, otherwise, it will be served for at least the minimum green time. To determine the value function of a barrier group with a fixed length, the lower optimization level considers the phase duration and sequence within that barrier group. The four phases in one barrier are further divided into two rings. For each ring, the combinations of the two phase lengths and sequences are iterated to find the minimum delay combination. This combination is considered as the optimal phase split and sequence for that barrier length.

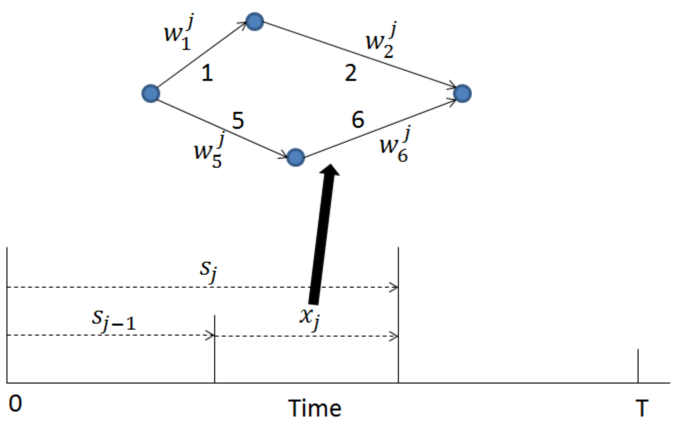


Figure 11. Two-level Optimization of Phase Allocation

The output of the phase allocation algorithm is a signal schedule for three barriers groups with phase sequence and duration. The schedule is a list of event which consists of control time, control command (e.g. call, hold, force-off, and omit) and which phase to control. The control commands are sent to Traffic Controller Interface component and then to signal controller.

Dilemma Zone Protection

The objective of the Traffic Control component not only considers the efficiency part of the intersection operation such as delay or number of stops, but also the safety. The number of vehicles in the dilemma zone (NVDZ) is used to represent safety. An analytical model (Feng et.al, 2014) was developed to predict the NVDZ at the onset of the yellow change interval. In a connected vehicle environment, NVDZ can be directly obtained by vehicle trajectory data from Trajectory Aware component. The phase allocation algorithm is solved first and the optimal schedule is provided as input to the Dilemma Zone algorithm. Then green time can be adjusted around the optimal duration to extend or terminate the signal control phase to reduce the NVDZ. An economic model was used to minimize the total combined delay of vehicle delay and NVDZ.

4.5.4.4 Priority Request Server (RSE, MRP)

The Priority Request Server component and the Priority Solver component provide the multi-modal priority control for priority requests from multiple vehicles. The AZ MMITSS approach to priority utilizes a new approach to priority and preemption rather than implementing legacy algorithms that exist in current traffic signal controllers. The new approach is based on research from the NCHRP 3-66 project "Traffic signal state transition logic using enhanced sensor information" (Urbanik, et. al., 2003) that investigated improved approaches to priority control. The algorithms implemented in the AZ MMITSS system use information in the connected vehicle Signal Requests Messages that are received from multiple vehicles of multiple modes. At any time, there may be several active requests that are simultaneously considered through a mathematical programming model that captures the ring-barrier and phase-interval based structure of the traffic signal controller. The formulation considers how to best serve all active requests based on a policy (N-level policy) set by the operating agency by selecting the mode priority weights in the mathematical programming model. The model is formed and managed by the Priority Request Server and solved by the Priority Solver. The Priority Solver uses an open source optimization solver – the GNU Linear Programming Kit (GLPK) to solve the optimization each time a new request is received, updated, or cancelled. The output of the Priority Request Server is a flexible signal timing schedule that the Traffic Control component can take as input when allocating phase green time based on the estimated vehicle arrivals.

The following section describes the mathematical programming model and is followed by a description of the flexible implementation model.

4.5.4.4.1 Mathematical Programming Model

It is assumed that the sequence of phases in each ring is fixed. Table 1 summarizes the notation of the mathematical model.

The main structure of the mathematical model is as follows.

Minimize *Priority Eligible Vehicle Request Delay + Coordination Request Delay*
s.t.
Precedence Constraints
Phase Duration & Interval Constraints
Priority Request Delay Evaluation Constraints

The objective of the mathematical model is to minimize the total weighted delay for different priority eligible vehicles and virtual coordination requests. There are three sets of constraints including precedence, phase duration and intervals, priority vehicle delay and coordination delay evaluation constraints. This mathematical model is designed to be solved over a rolling horizon. When a priority eligible vehicle enters the range of the RSE, it sends its estimated arrival time (ETA) and in-lane/out-lane information. The PRS computes the desired service phase and updates the table of active priority requests. When the vehicle changes its status (e.g. joining the queue, leaving the queue, or leaving the intersection), the model is solved again. The rolling horizon in this model is set to two cycles.

Precedence Constraints

The standard NEMA dual-ring, eight-phase structure is considered as shown in Figure 12. Each ring in the controller contains 4 phases, Figure 12(b). Although the model is based on the dual-ring eight-phase structure, it is possible to reformulate the model to deal with missing phases that occur at T-intersections and/or other structures. The dual-ring controller can be modeled by a precedence graph as depicted in Figure 12 (c). Arcs in the precedence graph represents the duration of phases, while nodes represent the phase transitions. Phase intervals can be easily visualized in the precedence graph by decomposing each arc into its respective interval precedence graph (Head, et. al., 2006).

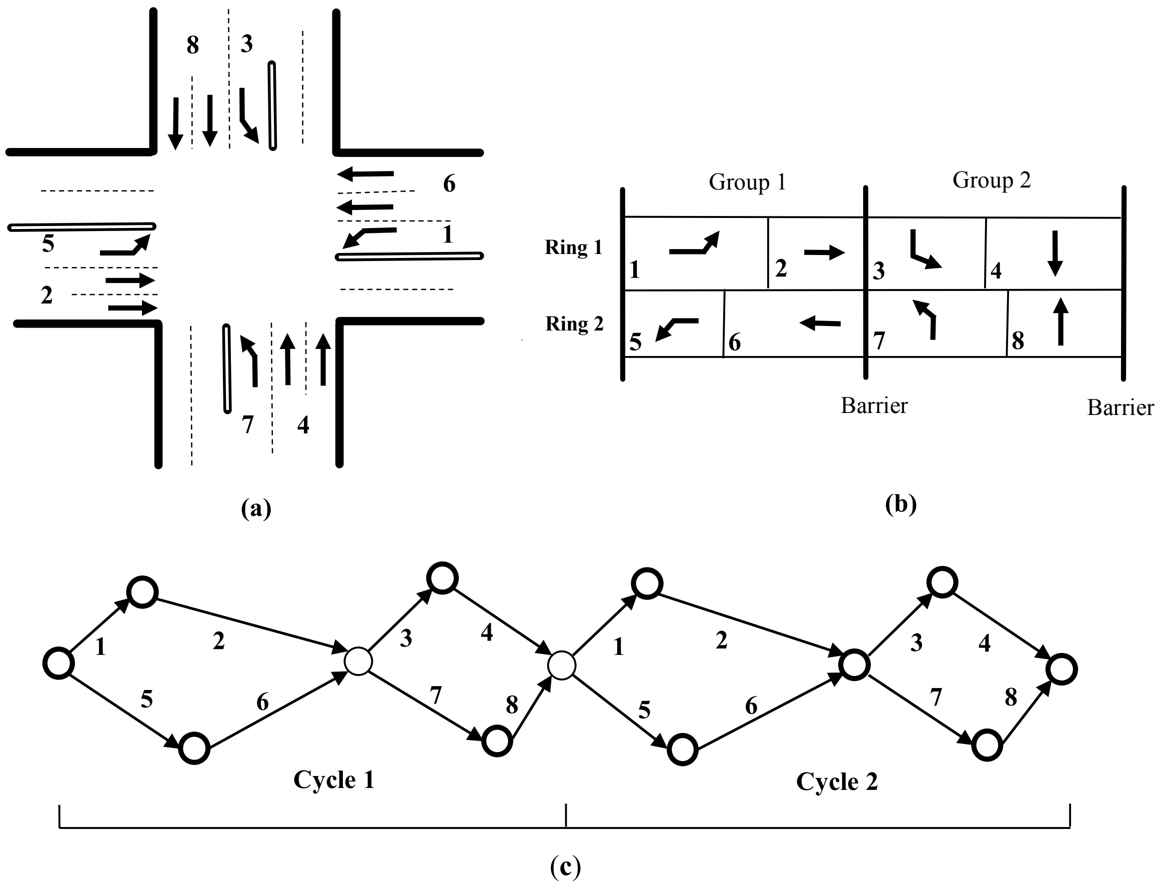


Figure 12. Dual-ring, eight-phase controller, and precedence diagram

The precedence graph defines the phase precedence relationships and green time feasible region. Traditionally, a signal plan is modeled by pre-allocated splits v_p^k, where $v_p^k = g_p^k + y_p + r_p$ However, in the proposed model, the allocated green time g_p^k and the splits v_p^k are decision variables while the clearance times $y_p + r_p$ are a predefined fix value for each phase. Constraints (1)-(13) model the precedence graph of dual-ring eight-phase controller in Figure 1 (c).

$$v_p^k = g_p^k + y_p + r_p \qquad \forall p,k \tag{1}$$

$$t_{p_1+1}^k = t_{p_1}^k + v_{p_1}^k \qquad \forall k \tag{2}$$

$$t_{p_1+2}^k = t_{p_1+1}^k + v_{p_1+1}^k \qquad \forall k \tag{3}$$

$$t_{p_1+3}^k = t_{p_1+2}^k + v_{p_1+2}^k \qquad \forall k \tag{4}$$

$$t_{p_2+1}^k = t_{p_2}^k + v_{p_2}^k \qquad \forall k \tag{5}$$

$$t_{p_2+2}^k = t_{p_2+1}^k + v_{p_2+1}^k \qquad \forall k \tag{6}$$

$$t^{k}_{p_2+3} = t^{k}_{p_2+2} + v^{k}_{p_2+2} \qquad \forall k \tag{7}$$

$$t^{k}_{p_1} = t^{k}_{p_2} + v^{k}_{p_2} \qquad \forall k, ((p_1 + p_2) \bmod 9) = 0 \tag{8}$$

$$t^{k+1}_{p_1} = t^{k}_{p_1+3} + v^{k}_{p_1+3} \qquad \forall k \tag{9}$$

$$t^{k+1}_{p_1} = t^{k}_{p_2+3} + v^{k}_{p_2+3} \qquad \forall k \tag{10}$$

$$t^{k+1}_{p_2} = t^{k}_{p_2+3} + v^{k}_{p_2+3} \qquad \forall k \tag{11}$$

$$t^{k+1}_{p_2} = t^{k}_{p_1+3} + v^{k}_{p_1+3} \qquad \forall k \tag{12}$$

$$v^{k}_{p} = g^{k}_{p} + y_p + r_p \qquad \forall p, k \tag{13}$$

Phase Duration & Interval Constraints

Since priority requests can be received at any point in the cycle, the optimization problem may be formulated at any time during any phase. Hence, it is necessary to capture the elapsed green time of the current phase. Constraints (14) and (15) address this need. If the phase is in the clearance interval, it is important to know how long it will take to start the next phase in the ring. This is modeled by constraints (16) and (17). The minimum and maximum green times are adjusted based on the elapsed time by constraint (18) and (19). If the phase is a coordinated phase and has returned to green early, the maximum green time limit is relaxed in constraint (20). Otherwise, there is a maximum green time for each phase.

$$t^{1}_{p_1} = ElapsGrn_{p_1} \qquad ,if\ p_1 \quad is \quad not \quad in \quad clearance \tag{14}$$

$$t^{1}_{p_2} = ElapsGrn_{p_2} \qquad ,if\ p_2 \quad is \quad not \quad in \quad clearance \tag{15}$$

$$t^{1}_{p_1} = Init_{p_1} \qquad ,if\ p_1 \quad is \quad in \quad clearance \tag{16}$$

$$t^{1}_{p_2} = Init_{p_2} \qquad ,if\ p_2 \quad is \quad in \quad clearance \tag{17}$$

$$\max\{0, \mathrm{gmin}_{\mathrm{p_1}} - ElapsGrn_{p_1}\} \le \mathrm{g}^{1}_{\mathrm{p_1}} \le \mathrm{gmax}_{\mathrm{p_1}} - ElapsGrn_{p_1} \qquad ,if \qquad I_{p_1} = 0 \tag{18}$$

$$\max\{0, \mathrm{gmin}_{\mathrm{p_2}} - ElapsGrn_{p_2}\} \le \mathrm{g}^{1}_{\mathrm{p_2}} \le \mathrm{gmax}_{\mathrm{p_2}} - ElapsGrn_{p_2} \qquad ,if \qquad I_{p2} = 0 \tag{19}$$

$$\mathrm{gmin}_{\mathrm{p}} \le \mathrm{g}^{1}_{\mathrm{p}} \qquad \forall p, p \in CP, I_p = 1 \tag{20}$$

$$\mathrm{gmin}_{\mathrm{p}} \le \mathrm{g}^{k}_{\mathrm{p}} \le \mathrm{gmax}_{\mathrm{p}} \qquad \forall p, k, p \notin CP, p \ne p_1, p \ne p_2 \tag{21}$$

Priority Request Delay Evaluation Constraints

Since there is uncertainty about traffic in front of the priority vehicle, the arrival time to stop bar is considered as a range (latest-earliest arrival time) to increase the robustness of the model. The range is calculated based on the speed of the vehicle, the distance to the stop bar, and vehicle mode. Constraints (22)-(27) calculate the delay of priority eligible vehicle requests. If the request can be served in the current cycle, the end of the requested phase in the current cycle $(t^{1}_{p(j_m)} + g^{1}_{p(j_m)})$, should be larger than the latest arrival time of the request to the intersection, u_{j_m} (right side of constraint (23)). In this case, the time between the starting time of requested phase in the current cycle and the earliest arrival time of the request (right side of constraint (22)) is smaller than the time interval between the starting time of requested phase in the next cycle and the earliest arrival time of the request constraint (right side of constraint (25)). Since d_{j_m} is being minimized in the objective function, from constraint (25), θ_{j_m} is forced to be zero. Therefore, constraints (25), (26), and (27) are relaxed. The delay of the priority request is calculated by constraint (22).

If the request is to be served in the next cycle, the requested phase in the current cycle should be terminated before the latest arrival time of the request (right side of constraint (23)). Therefore, constraint (23) forces θ_{j_m} to be 1 and constraint (24) is relaxed. Constraint (26) ensures the allocated green time to the requested phase in the next cycle is at least more than the time interval between the latest arrival time of the request and the earliest arrival time of the request. Constraint (27) guarantees that the end of the requested phase in the next cycle is more than the latest arrival time of the priority request. Constraint (25) calculates the delay of the priority request.

$$d_{j_m} \geq t^k_{p(j_m)} - l_{j_m} \qquad \forall m, j_m, k=1 \tag{22}$$

$$M\theta_{j_m} \geq u_{j_m} - (t^k_{p(j_m)} + g^k_{p(j_m)}) \qquad \forall m, j_m, k=1 \tag{23}$$

$$g^k_{p(j_m)} \geq (u_{j_m} - l_{j_m})(1-\theta_{j_m}) \qquad \forall m, j_m, k=1 \tag{24}$$

$$d_{j_m} \geq (t^k_{p(j_m)} - l_{j_m})\theta_{j_m} \qquad \forall m, j_m, k=2 \tag{25}$$

$$g^k_{p(j_m)} \geq (u_{j_m} - l_{j_m})\theta_{j_m} \qquad \forall m, j_m, k=2 \tag{26}$$

$$t^k_{p(j_m)} + g^k_{p(j_m)} \geq u_{j_m}\theta_{j_m} \qquad \forall m, j_m, k=2 \tag{27}$$

In constraint (25), the term $t^k_{p(j_m)}\theta_{j_m}$ is a multiplication of a real and a binary decision variable. In order to linearize this term, an intermediate variable $x_{j_m} = t^k_{p(j_m)}\theta_{j_m}$ is introduced and non-equalities (29), (30) and (31) are added to the constraints. These four constraints ensure that x_{j_m} is either 0 (when $\theta_{j_m} = 0$) or $t^k_{p(j_m)}$(when $\theta_{j_m} = 1$). Also, constraint (25) is replaced with constraint (28).

$$d_{j_m} \geq x_{j_m} - l_{j_m}\theta_{j_m} \qquad \forall m, j_m \tag{28}$$

$$M\theta_{j_m} \geq x_{j_m} \qquad \forall m, j_m \tag{29}$$

$$x_{j_m} \geq t^k_{p(j_m)} - M(1-\theta_{j_m}) \qquad \forall m, j_m, k=2 \tag{30}$$

$$t^k_{p(j_m)} + M(1-\theta_{j_m}) \geq x_{j_m} \qquad \forall m, j_m, k=2 \tag{31}$$

In this model, coordination is considered as a form of priority through the use of virtual coordination requests. Given a pre-calculated signal coordination plan, the coordinated phase split, the offset and cycle length are defined. The total number of coordination requests is at most two because a two cycle model is assumed. Every time the optimization problem is formulated, the earliest (l) and latest $(u$) arrival time of the coordination request for the next two cycles are calculated based on the given coordination plan and the current time in the cycle. Coordination request delay for coordinated phase *p* in the first cycle (c_p^1) and in the second cycle (c_p^2) are calculated using constraints similar to (22) – (31) while d_{j_m} is substituted with c_p^1 and c_p^2.

Objective Function

The objective of the mathematical programming model is to minimize the delay of the priority requests as shown in equation (32):

$$\min_{t,g,\theta,\mu} \quad \alpha\left(\sum_{m\in TM, j\in J_m} w^m d_{j_m}\right)+\beta\left(\sum_{k,p\in CP} c_p^k\right) \tag{32}$$

The two main terms form a weighted summation of weighted sum of priority request delay and coordination request delay. In the first term, requests for each mode have a predefined weight w^m that reflect the importance of that mode. The second term is the summation of coordination requests delay over the next two cycles. α and β are weights that show the importance for priority eligible vehicles compared to the coordination requests.

4.5.4.5 Flexible Implementation Algorithm

Since the real-time traffic demand for each phase is unknown due to the stochastic nature of traffic flow, a flexible implementation algorithm was designed to consider non-priority eligible vehicles. This algorithm reduces the negative impacts on regular vehicles because it allows the signal to operate based on either adaptive control or actuated control while ensuring the priority requests are served with minimum delay. Integration of the mathematical programming formulation and actuated control will result in the phases being allowed to gap out if no vehicles are detected; to extend if a priority request needs to be served; to force-off the phase when the maximum green extension is reached; or, to force-off the phase when it is needed to serve the priority request in the next phase(s) or cycle.

Given the assumption of a fixed-phase sequence, a phase–time diagram (He, et. al., 2011) is constructed with one horizontal axis representing time and two vertical axes representing the phases in each of the two rings (the left axis is ring 1 and the right axis is ring 2), as shown in Figure 13. The origin denotes the current time and current phase, which is shown as the start of phases P_1 and P_5, but could be any feasible phase combination. The projection of every arc into the time axis determines the phase duration. Any piecewise line starting from the origin represents a signal plan in the phase–time diagram. However, the feasible region of the signal plan is bounded in a cone-shaped area by the shortest path (shortest timing as determined by each phase's minimal green times) and the longest path (longest time as determined by each phase's maximal green times). The black dashed lines in Figure 13, show this feasible region which is determined by the phase minimum and maximum times. A priority request j_m with requested phase $p(j_m)$ is associated with a desired service time interval, $[l_{j_m}\ u_{j_m}]$ which represents the earliest arrival time and latest arrival time of the jth request of mode type *m*. In Figure 13 (c), the thin orange piecewise line represents the timing of ring 1 and the thick blue piecewise line represents the timing of ring 2.

In Figure 13 (c), there are 3 requests. Two of them are mode m=2 and one of them is mode m=5. The first request of mode m=2 has service time interval $[l_{1_2}\ u_{1_2}] = [2.0\ 6.0]$ and the requested phase $p(1_2) = P_2$. This request has delay because the feasible starting time of the requested phase is larger than the earliest arrival time of the request ($7.0 > 2.0$). The second request of mode m=2 has service time interval $[l_{2_2} \ u_{2_2}] = [36.0 \ 41.0]$ and the request phase $p(2_2) = P_7$. This request does not have any delay because the starting time of the requested phase is smaller than the earliest arrival time of the request ($21.0 < 36.0$). The first request of mode m=5 has service time interval $[l_{1_5} \ u_{1_5}] = [26.0 \ 30.0]$ and the requested phase $p(1_5) = P_4$. This request does not have any delay.

Assume there is just one priority request in the request list. If the optimal delay for that request is not zero, there is one and only one optimal signal timing schedule. The optimal signal timing would allocate minimum green time to the phases so that the requested phase starts as soon as possible. But, if the optimal delay is zero, there are multiple optimal solutions for the problem. Figure 13 (a) and (b) show two phase time diagrams for request j_m with service time interval $[l_{j_m} \ u_{j_m}]$ and the requested phase $p(j_m) = P_8$. The only mutual part of these two realizations is that phase P_8 is green during the interval $[l_{j_m} \ u_{j_m}]$. In order to ensure phase P_8 is green during the service time interval, the predecessor phases should be held (phase HOLD) or forced-off (a phase FORCE-OFF) at a specific threshold.

Figure 13 (d) shows an illustration of the set of "backward critical points" for serving request j_m without delay. These points are divided into two groups, backward left (BL) and backward right (BR). The BL points are calculated based on BR points. Equation (33)-(39) explain how BRs and BLs are calculated.

$$BR1 = u_{j_m} \tag{33}$$

$$BR2 = l_{j_m} - (y_{P_7} + r_{P_7}) \tag{34}$$

$$BR3 = BR2 - \mathrm{gmin}_{\mathrm{P}_7} - (y_{P_6} + r_{P_6}) \tag{35}$$

$$BR4 = BR3 - \mathrm{gmin}_{\mathrm{P}_6} - (y_{P_5} + r_{P_5}) \tag{36}$$

$$BL1 = BR1 - \mathrm{gmax}_{\mathrm{P}_8} - (y_{P_7} + r_{P_7}) \tag{37}$$

$$BL2 = BR2 - \mathrm{gmax}_{\mathrm{P}_7} - (y_{P_6} + r_{P_6}) \tag{38}$$

$$BL3 = BR3 - \mathrm{gmax}_{\mathrm{P}_6} - (y_{P_5} + r_{P_5}) \tag{39}$$

gmax and gmin are maximum and minimum time.

As Figure 13(d) shows, some of the BL and BR points are infeasible points. For example, BR4, BL2, and BL3 are out of the feasible region that is created by minimum and maximum green time lines. Figure 13 (e), illustrates the "forward critical points" (Equations (40)-(45)). These points are also divided into two parts, Forward Right (FR) and Forward Left (FL). The FRs are points where each phase has been extended till its max time. The FLs are points where only minimum green time has been allocated to each phase.

$$FR1 = \max(0, \mathrm{gmax}_{\mathrm{P}_5} - Elaps_{P_5} + Init_{P_5}) \tag{40}$$

$$FR2 = FR1 + \mathrm{gmax}_{\mathrm{P6}} - (y_{P_5} + r_{P_5}) \tag{41}$$

$$FR3 = FR2 + \mathrm{gmax}_{\mathrm{P7}} - (y_{P_6} + r_{P_6}) \tag{42}$$

$$FL1 = \max(0, \mathrm{gmin}_{\mathrm{P}_5} - Elaps_{P_5} + Init_{P_5}) \tag{43}$$

$$FL2 = FL1 + \mathrm{gmin}_{\mathrm{P6}} - (y_{P_5} + r_{P_5}) \tag{44}$$

$$FL3 = FL2 + \mathrm{gmin}_{\mathrm{P7}} - (y_{P_6} + r_{P_6}) \tag{45}$$

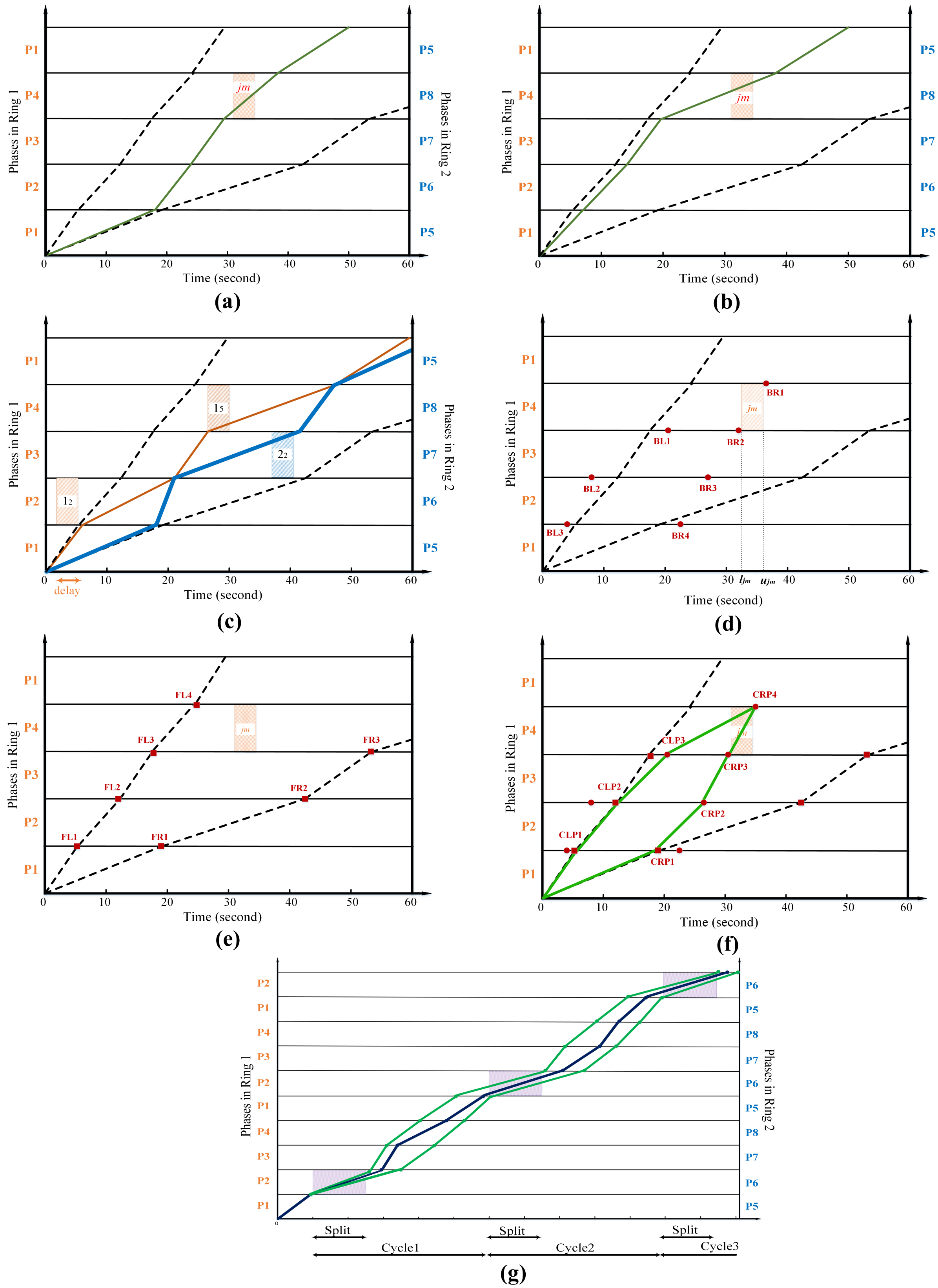


Figure 13. Time – phase diagram and critical points illustration

Critical Points (CP) for serving priority request j_m are points that create a feasible region for signal control that ensures the request is served without delay. They are also divided to two major points, Critical Left Points (CLP) and Critical Right Points (CRP). Figure 13 (f) shows these points and the feasible region that they create for the serving priority request j_m. The right points of this region (CRPs) are the minimum of the BRs and FRs for each phase, while the left points of the region (CLPs) are the maximum of the BLs and FLs for each phase.

In the case of multiple requests, the optimal request-cycle assignment can be sorted based on the earliest time of service. The CP can be obtained following the similar procedure for every request. Figure 13 (g) illustrates the CP for two coordination requests for phases P2 and P6. The bold blue line shows one possible timing realization of phases in ring 2.

4.5.5 Performance Observer Component

The Performance Observer component is responsible for acquiring data from other components, including the Trajectory Aware and the Traffic Controller Interface components, to compute observed performance measures. The Performance Observer includes estimation, managing, and archiving the data. All the acquired performance observations and metrics are collected and reported in terms of peak period, all day, or specially defined time period.

As stated in the design document the Basic Safety Message contains the vehicle temporary ID, latitude, longitude, elevation, speed, heading, and vehicle size. It doesn't have the vehicle type information, although it can be determined according to the Signal Request Message. The high resolution data (i.e. received BSMs every 0.1 second) are used to reconstruct the vehicle trajectories.

Table 7 summarizes the performance measures being calculated and estimated in this component with their associated sources of data.

Table 7. Performance Metrics and associated data sources

Performance Metric	Abbreviation	Unit	Data Source
Travel Time	TT	Second	Trajectory Aware
Delay	D	Second	Trajectory Aware
Travel Time Variability	TTV	Second	Trajectory Aware
Delay Variability	DV	Second	Trajectory Aware
Queue Length	QL	Meter/number of vehicles	Trajectory Aware
Number of Stops	NS		Trajectory Aware
Volume	V		Traffic Controller Interface
Occupancy	O	%	Traffic Controller Interface
DSRC Range	RNG	Meter	Trajectory Aware
Packet Loss Rate	PLR	%	Trajectory Aware
Market Penetration Rate	MPR	%	Trajectory Aware Trajectory Aware Traffic Controller Interface

Equation (1) defines how the travel time for each individual vehicle for a specific phase is calculated by subtracting the time that the vehicle leaves the radio range from the time that the vehicle entered the radio range. This range can be measured separately for each intersection based on the received BSMs on RSE side, and doesn't have to be exactly the same as the DSRC radio range.

$$Travel\ Time\ =\ Leaving\ Time - Entrance\ Time \tag{1}$$

One of the principle concerns confronting the development of connected vehicle applications is ensuring privacy of the public travelers. The connected vehicle system is being developed with this privacy issue as a requirement. According to "Annex E: Traffic Probe Message Use and Operation" in the SAE J2735 (2009) Standard, the data collection of a connected vehicle does not begin until 500m or 120 seconds (whichever occurs first) after the vehicle start up to aid anonymity. If a vehicle changes its temporary ID number while it is being observed, it is not difficult to associate its new ID number with its old number based on the position data that is received every 0.1 seconds. However, making this association would violate the privacy requirement.

Hence, trajectory fragments are utilized to estimate the travel time over a certain road segment (i.e. within the boundaries of DSRC range). When a vehicle randomly generates a new ID, it's considered as a new entity in the system in order to avoid matching the two trajectory fragments. Keeping the trajectories for a short amount of time and assuming the trajectories fragments are independent, are two ways to help ensure no vehicle is tracked.

Figure 14 shows how the Travel Time (TT) and Distance (D) of a single connected vehicle are divided into two separate fragments. At time t_1, the vehicle enters the data collection range (or the DSRC radio range). From the time that the ID is changed (t_2) till the vehicle leaves the data collection range (t_3), it's treated as a different connected vehicle in the system. The two fragments of one trajectory are designated as Type 1 and Type 2 trajectory fragments. The number of trajectory fragments stored in the database is twice the number of vehicles that changed their IDs.

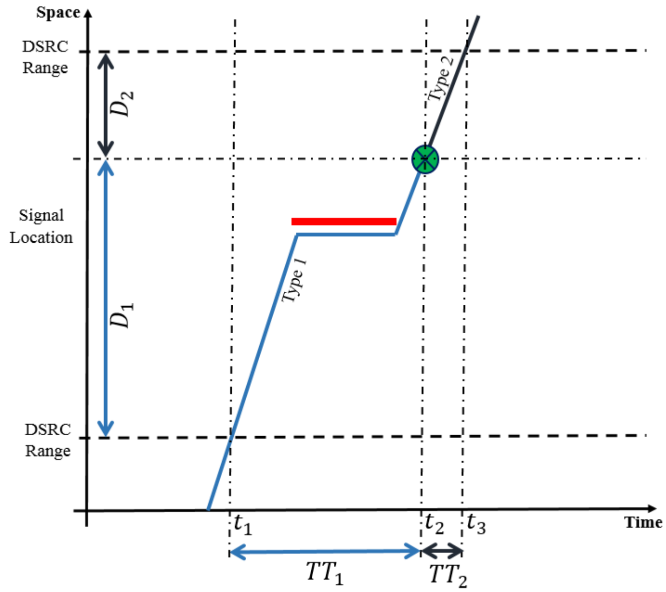


Figure 14. Trajectory Fragments of a single Connected Vehicle with a changed ID

4.5.5.1.1.1 Travel Time Estimation Using the Extended Tardity Function

The average Extended Tardity is calculated based upon the trajectory fragments in the time-space diagram to obtain an estimate of the average link travel time of the vehicles in the system. In this Equation *n* is the number of the vehicles that didn't change their IDs; m_1 is the number of Type 1 vehicles that changed their IDs, and accordingly m_2 is the Type 2 number of cars with changing ID.

$$ETT = \left[\sum_{i=1}^{n} \frac{TT^i}{D^i} + \sum_{j=1}^{m_1} \frac{TT_1^j}{D_1^j} + \sum_{j=1}^{m_2} \frac{TT_2^j}{D_2^j}\right] \times \left[\sum_{i=1}^{n} D^i + \sum_{j=1}^{m_1} D_1^j + \sum_{j=1}^{m_2} D_2^j\right] / \left[n + m_1 + m_2\right]$$

where,

TT^i = Travel Time of Vehicle *i* that doesn't change its ID,
D^i = Distance Traveled by Vehicle *i* that doesn't change its ID,
TT_1^j = Travel Time of Vehicle *j* of Type 1,
TT_2^j = Travel Time of Vehicle *j* of Type 2,
D_1^j = Distance Traveled by Vehicle *j* of Type 1,
D_2^j = Distance Traveled by Vehicle *j* of Type 2.

The estimation of delay depends on two factors. One is the vehicle total travel time and the other is the free flow travel time. The travel time is estimated above, and as shown in equation (2) the free flow travel time is obtained based on the free flow speed and is specified as a pre-determined value in this design. However, there are different definitions of free flow speed recognized by different agencies (e.g. Speed Limit), and they can be implemented to the logic as a configurable variable.

$$Free\ Flow\ Travel\ Time = \frac{Distance\ Range}{Free\ Flow\ Speed} \quad (2)$$

By knowing the values of travel time of each vehicle and free flow travel time at the intersection, delay is calculated as:

$$Delay = Travel\ Time - Free\ Flow\ Travel\ Time$$

Vehicle stopped delay is also measured based on the amount of time that the vehicle's speed is less than a threshold (designed to be 2 miles per hour) before reaching the stop bar.

When the number of vehicles traversing the intersection in a specific movement is known, total travel time and delay for movement "*i*" are calculated as:

$$(Total\ Travel\ Time)_i = Number\ of\ Vehicles \times (Average\ Travel\ Time)_i$$

$$(Total\ Delay)_i = Number\ of\ Vehicles \times (Average\ Delay)_i$$

Based on the simple equations presented above, travel time, delay, and the total values of these measures are calculated for each specific movement. For obtaining the intersection travel time or delay the equations below are used, where *"i"* is an indicator of the available movements at the intersection:

$$\sum_{i=1}^{12} (Total\ Travel\ Time)_i$$

$$\sum_{i=1}^{12} (Total\ Delay)_i$$

The variability of travel time can be related to the Expected Value and Variance through the following relations (TT: Travel Time; FTT: Free Flow Travel Time; D: Delay).

$$TT = FTT + D$$
$$E(TT) = FTT + E(D)$$
$$Var(TT) = E[(FTT + D) - E(FTT + D)]^2 = E[D - E(D)]^2 = Var(D)$$
$$\sigma(TT) = \sigma(D)$$

Travel time is free flow travel time plus delay. The expected value of travel time (average) is the free flow travel time (which is assumed to be deterministic) plus the expected value of delay (average delay). By definition of the variance, it can be shown that the variance of travel time is equal to the variance of delay.

An important performance metric being analyzed at the intersection level is queue length. The functional description of this measure is divided into two parts: Observing and estimating the queue length in real time in terms of the distance of the last stopped vehicle from the stop bar for each lane, and/or measuring the queue in terms of the number of stopped vehicles at each lane. These measures can be easily transformed into each other. For example, according to the data available from BSMs coming from the OBEs (vehicle trajectory database), the length of each individual vehicle is known, and by knowing the number of vehicles in the queue, the overall queue length in terms of the distance from the stop bar can be calculated.

The main goal is to have an accurate and acceptable estimation of this measure when the market penetration rate is low. The algorithm developed addresses the market penetration problem by adding a parameter called Queue Increase Rate. Basically, by having information about the last stopped vehicle a lower bound for the queue length is provided. Then by adding the increase rate factor which is calculated based on the combination of historical and real-time data coming from the loop detectors to the current known queue status, an accurate estimate of the queue length is achieved.

Figure 15 shows the flow chart for queue length estimation algorithm which estimates the queue length by lane either with or without system detectors.

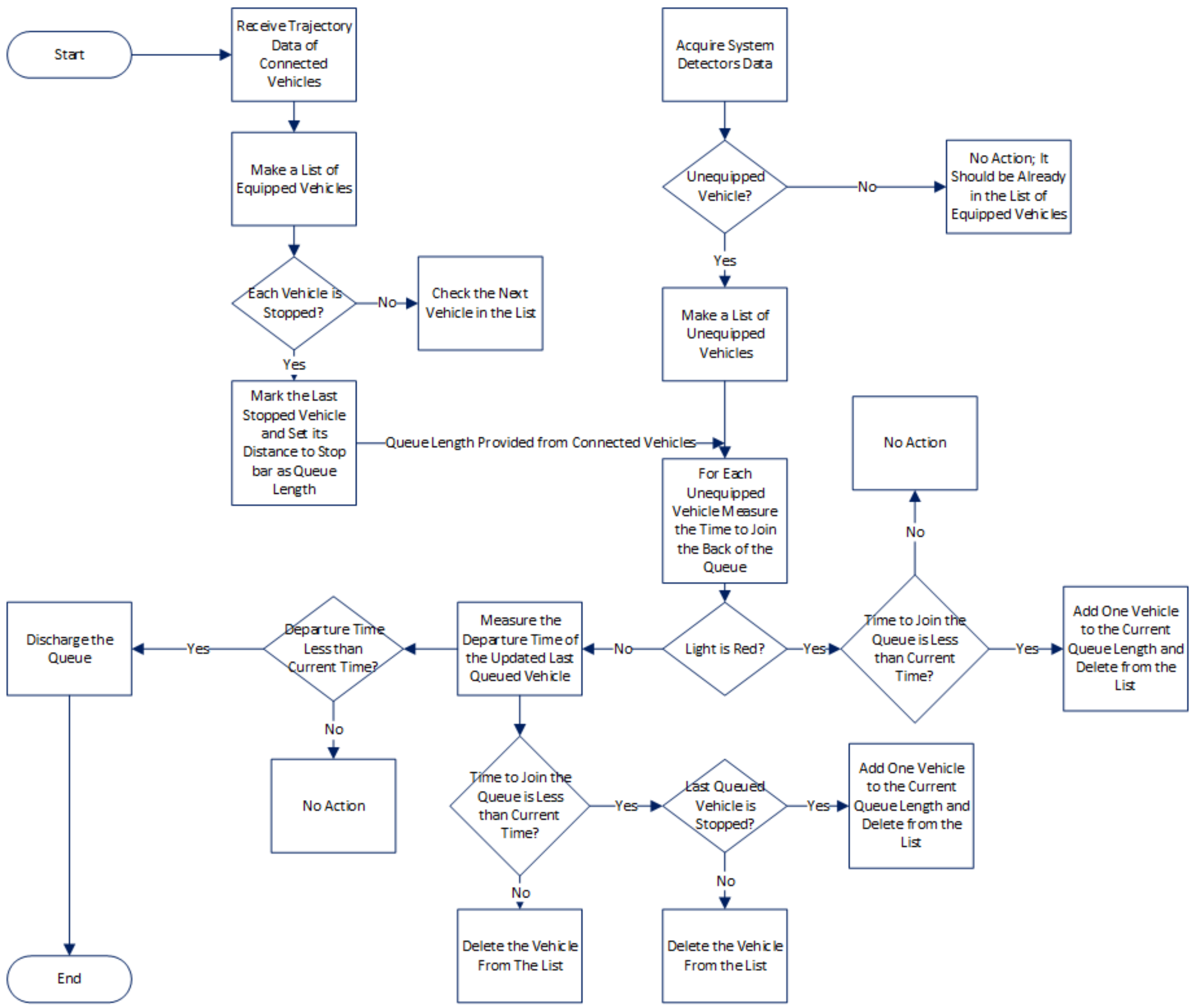


Figure 15. Queue Length Estimation Flow Chart using Detectors and BSMs Data

The additional performance metrics that represent the connected vehicle operations include the number of equipped vehicles traversing a section and Market Penetration Rate (MPR). It is measured based on the data from System Detectors and received BSMs from OBEs.

By having an accurate vehicle counts from system detectors and information about the number of connected vehicles at the intersection during a predefined time period, the market penetration rate is computed as follows:

$$Market\ Penetration\ Rate\ (\%) = \frac{Number\ of\ Connected\ Vehicles}{Total\ number\ of\ vehicles} \times 100$$

This metric is calculated for each approach, and then by computing a volume weighted sum, the penetration rate for each intersection is estimated. Given the market penetration rate of each individual approach, the average MPR value is calculated based on the following equation where "n" is the total number of approaches available at intersection:

$$Average\ Intersection\ MPR\ (\%) = \sum_{k=1}^{n} \frac{(MPR)_k}{n}$$

4.6 Design and Implementation of California MMITSS Prototype

This California MMITSS implementation followed the high level system design described in Section 4.3.2 and made adaptations to fit the existing conditions in the California Test Bed.

4.6.1 Architecture of California MMITSS

The California test bed chose the "thin" RSE deployment model, where the MRP that hosts the core MMITSS intersection-level applications is a Linux-based computer. In the vison of future expansion and integration with CV applications beyond MMITSS, the California MMITSS is centered with the MRP, with a data manager to interface with the traffic signal controller, RSE, Nomadic Traveler Server, and the MMITSS Central System, and the Nomadic Traveler Server.

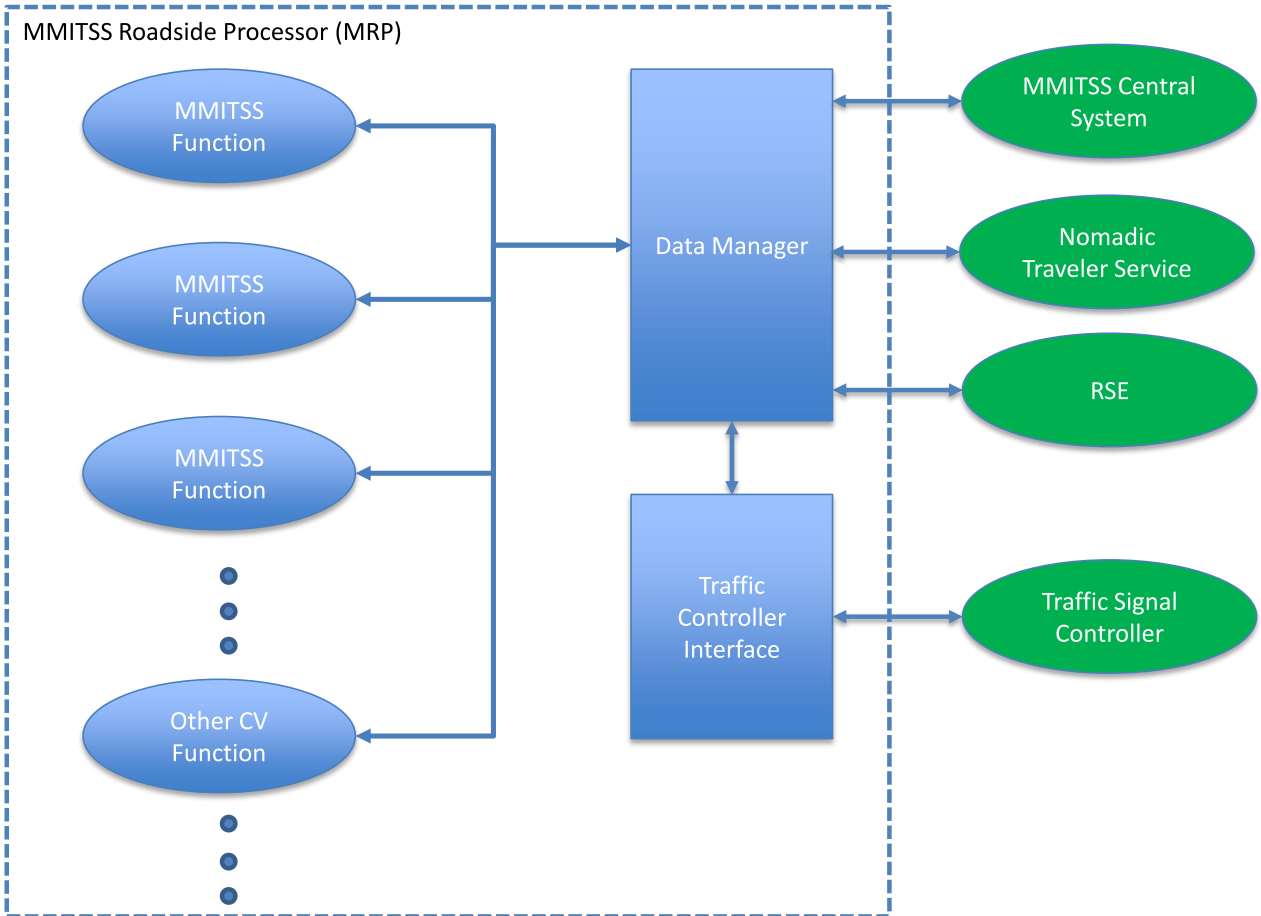


Figure 16. California MMITSS Implementation Architecture

Figure 16 illustrates the architecture of California MMITSS. The MRP data manager serves as the single sink of data from various sources and the single source for sharing the data among MRP components such that data from different sources can be used synchronously. With this architecture, integrating with other CV applications only requires the applications to register with the data manager for sharing and routing the application data.

The MRP receives multiple messages from different physical components, such as over-the-air DSRC messages from the OBEs via the RSE, pedestrian SRMs from the nomadic devices via the Nomadic Traveler Server, and configuration, signal status and detection data from the traffic controller. These messages arrive asynchronously and are processed by different MRP components, e.g. BSMs for vehicle tracking, SRMs and controller data for priority control, etc. The decision of MMITSS traffic/priority control is based on the combined vehicular, pedestrian and controller data.

The use of a data manager for routing synchronized data to the different software modules would be preferable to independently connecting these modules. Table 8 compares some of the more important inter-process tasks that need to be performed by the software system.

Table 8. PATH Comparison of Inter-process Tasks with and without Data Manager

Task	Without Data Manager	With Data Manager
Debugging	Chain of data production and use difficult to track	Easy to track data flow and identify mistakes in logic
Current Data Repository	Data must be acquired separately from each source	Single point for current data
Synchronization	Data can be timestamped, but a datum used in a particular module may have been acquired at a different time from data used in a related module	Single timestamp for current data, along with timestamp for data acquisition, so logic can be traced
Complexity	Many separate (and possibly redundant) connections among software modules	One connection = one use of data
Data Archiving for Performance Measurements	Related to synchronization and current data repository tasks; difficult to track data usage in a post-mortem	Dual timestamps ("data acquired" and "data used") allow tracking of logic and assumptions made to decide how good the system is
Fault detection	If a piece of hardware does not provide timely data, the subsequent effects on the system are hard to trace	Easy to trace stale data and detect a problem in either software logic or hardware
Future expansion and integration	May need to modify more than one module to accommodate connections to a new module	Adding a new module only requires adding a UDP/TCP socket to the data manager, along with the data structure

The data manager would connect to other software modules with a socket pair. Data sources, such as the Traffic Controller Interface, would push timestamped data into the data manager. The data manager would push synchronized data into data sinks, such as the Performance Observer. Modules that perform calculations, such as MMITSS traffic/priority control decisions, would be based on synchronized data.

A datum is timestamped when received by the data manager; it is again timestamped by a module when that datum is acted upon. These two timestamps, together with the datum itself and an identifier of the module using the datum, constitute one data point. A module using data from more than one source would process all the data and timestamp them once, so that the data logger could write it all to a data file for post processing.

4.6.1.1 MMITSS Roadside Processor

The MRP selected for the California testbed is described by the following specification:

```
Jetway NF9E-Q77 Mini-ITX Motherboard, Q77 Express vPro iAMT, LGA1155, Ivy
   Bridge
Enclosure, power supply: Super Case MI-100BK Mini-ITX Case
Digital I/O board: PCIe-IIRO-8
CPU chipset, fan: Intel Core i5-3570 Ivy Bridge 3.4GHz (3.8GHz Turbo
   Boost) LGA 1155 77W Quad-Core Desktop Processor Intel HD Graphics 2500
   BX80637i53570
Disk drive: Seagate Constellation ES ST500NM0011 500GB 7200 RPM 64MB
   Cache SATA 6.0 Gbps 3.5" Enterprise Internal Hard Drive -Bare Drive
System memory: Transcend 8GB (2 x 4GB) 240-Pin DDR3 SDRAM DDR3 1333
   Desktop Memory Model JM1333KLN-8GK
Operating System: Ubuntu Server 12.04.2 LTS
```

The MRP has two RS-232 serial ports for communicating with the traffic signal controller, and one Ethernet port for communicating with RSE and the backhaul.

4.6.1.2 RSE Radio

Both the Arizona and the California testbeds use the Savari StreetWave version 3.1 RSE units. The units support two 5.9 GHz DSRC radios and an Ethernet interface for communications. The product spec sheets are available in Appendix 11.2.

4.6.1.3 OBE Devices

The California testbed used Savari MobiWave units and Arada LocoMate Mini 2 units. These units are general purpose and provide a powerful and flexible platform for development and testing. The product data sheets for the OBEs are Appendix 11.2.

4.6.2 Traffic Signal Controller and Communication Protocol

The California testbed uses Model 2070 traffic signal controllers running Caltrans Traffic Signal Control Program (TSCP) to control the traffic signals. The controllers operate in coordinated-actuated mode between 6 AM and 9 PM with three time-of-day coordination plans, and runs free between 9 PM and 6 AM. The traffic controllers are inter-connected and communicate with a field master controller over multi-drop serial communications such that each connected local controller operates the same coordination plan.

Caltrans TSCP has deployed signal priority control logic to provide early green and green extension treatments for transit vehicles traveling on the main streets. The parameters for signal priority control (e.g., speed-up rate for early green treatment and maximum green extension interval) are configurable through AB3418 messaging over RS-232 communications.

AB3418 messages are variable length and have the following format:

```
    0x7e                0x13   0xC0                                    0x7e
+------------+----------+---------+-------+-----------------+---------+--------+----------+
| Start Flag | Address | Control |  IPI  | Message Type |  Data   |  FCS  | End Flag |
+------------+----------+---------+-------+-----------------+---------+--------+----------+
    1 Byte     1 Byte    1 Byte   1 Byte     1 Byte      0-n Bytes  2 Bytes  1 Byte
```

Existing AB3418 messages allow to read and set TSCP memory blocks for remotely downloading and modifying controller’s phase timing, plan timing, and priority control parameters. TSCP control logic responds to direct field sensor inputs, such as loop detectors, pedestrian push-buttons, and priority phase selector. Caltrans Division of Traffic Operations (TrafficOps) modified TSCP to support the MMITSS functions. The modifications include

- Push out *GetStatus8Response* message, which contains detector presence data, at a rate of 1Hz on port C20S;
- Push out *GetStatus8eResponse* message, which contains system detector count and occupancy data, every 60 seconds or at the end of a signal cycle on port C20S;

- Push out *signal_status* message, which contains information for constructing J2735 SPaT messages, at a rate of 10Hz on port C21S; and
- Accept *SoftCall* message to call/extend vehicular phases, call pedestrian phases, and call signal priority on port C20S.

TSCP control logic that responds to *SoftCall* messages remains the same as the logic that responds to direct field sensors. *SoftCall* is implemented as an alternative detection means via AB3418 protocol beyond the field sensor detection. The effect of *SoftCall* is as follows:

- *Pedestrian call* will place a pedestrian service request on the phases called;
- *Vehicle call* will place a service request on the calling phase when the phase is in yellow or red; and place an actuation (passage) when the phase is in green. Continuous calls will be required to hold the request (at a rate of 50 milliseconds); and
- *TSP call* will place a priority request on the calling phase and the call should be continuously sent at a rate of 50 milliseconds. The traffic controller will execute the build-in priority control logic until the *TSP call* is dropped.

4.6.3 Wireless Backhaul

A 4G/LTE mobile backhaul is provided at each intersection in the California testbed. The physical devices installed inside the traffic signal cabinet include an MRP, a 4G/LTE modem, and a router.

4.6.4 Summary of California MMITSS Design

This section summarizes the primary software components of California MMITSS. More detailed and completed descriptions of the software components can be found in the document of "MMITSS System Design – California Portion".

4.6.4.1 California MMITSS Software Components

The primary software components of California MMITSS implementation is shown in Figure 17.

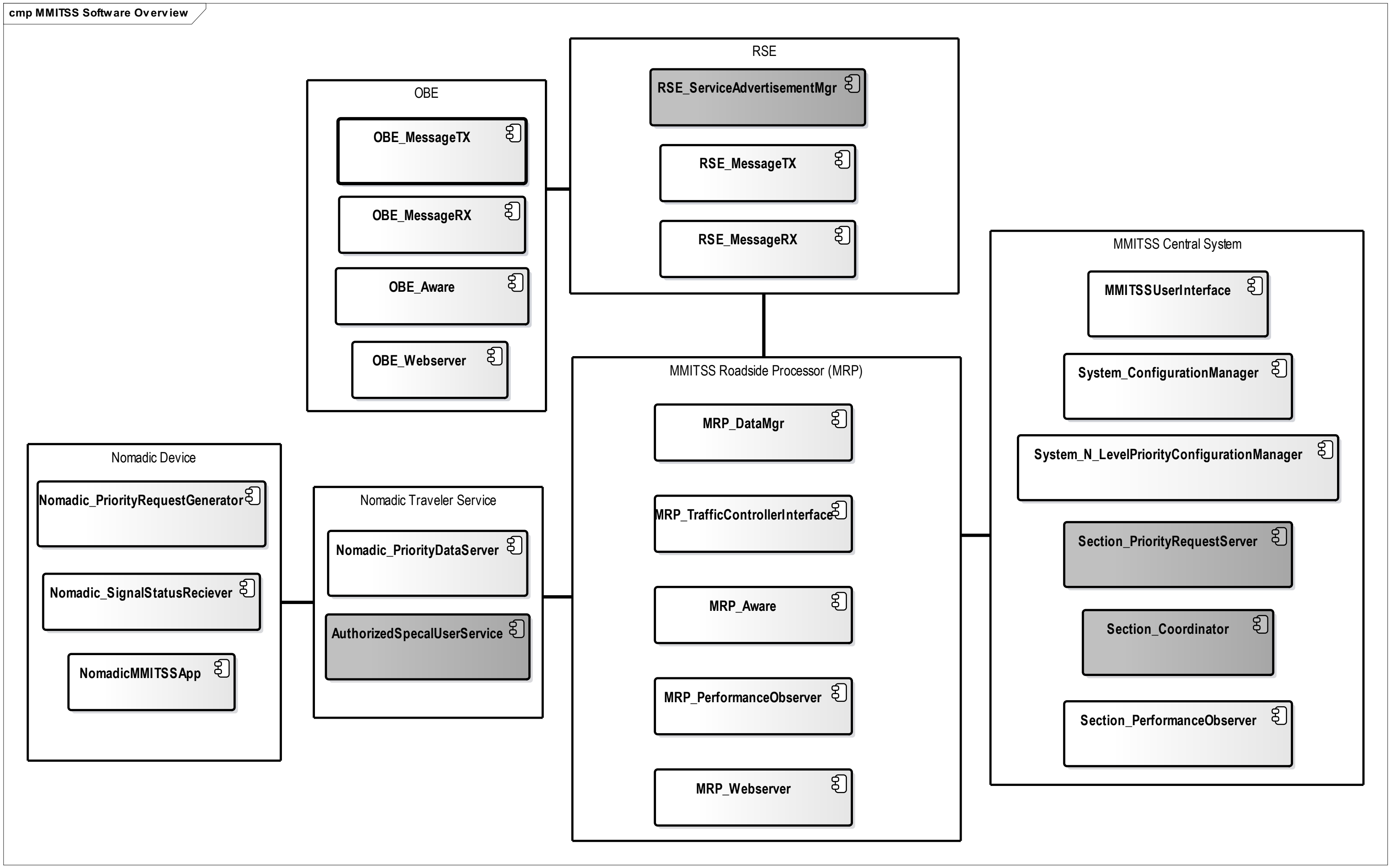


Figure 17. California MMITSS Software Components

Comparing with the software components identified in the MMITSS high-level system design (Section 4.3.2), the following adaptions are made for the California MMITSS:

- The functionalities of *OBE_MAP_SPaT_Receiver*, *OBE_BSMDataTransmitter* and *OBE_PriorityRequestGenerator* are grouped into one component (e.g. *OBE_Aware*). Regardless of whether the OBE is eligible for priority or not, the functionalities of processing MAP and SPaT messages and tracking the vehicle on the MAP are desirable features. With a single component that provides these functionalities, it could be used for other vehicle-centered applications.
- The *MRP_MAP_SPaT_Broadcast* and *MRP_SignalStatus_Nomadic* are combined and implemented as part of the *MRP_DataMgr* component. The MAP and SPaT messages broadcast over the air are the same messages provide to the pedestrian data server with different destination IP address and port. One of the functionalities of the *MRP_DataMgr* is to receive the *signal_status* message from the traffic signal controller and to generate the SPaT message.
- The functionalities of *MRP_PriorityRequestServer* and *MRP_TrafficControl* are combined as one component (e.g. *MRP_Aware*) as the *SoftCall* messages sending to the traffic controller need to be at a rate of 20 Hz to avoid dropping a control command.

4.6.5 MessageRX and MessageTX

The same *MessageRX* and *MessageTX* components are deployed at RSE and OBE. The *MessageRX* component is responsible for connecting with the WME stack, receiving over-the-air WSMs/WSMPs, and forwarding the WSMs/WSMPs to the registered software components for further processing. The *MessageTX* component is responsible for connecting with the WME stack, receiving payloads from the registered software components, and transmitting WSMs/WSMPs over the air.

Registration of software components for *MessageRX* and *MessageTX* are included in a configuration file in terms of message type, associated action (e.g. *RX*/*TX*), and register's IP addresses and ports. A sample configuration file for RSE is shown below. It indicates that the RSE is receiving BSM and SRM, and forwarding to *sendSocket*, and is receiving MAP, SPaT and SSM payload from *listenSocket*, and broadcasting over-the-air.

```
#######################################
# WME stake configuration file for RSE
#######################################
# Format of application configuration:
#    enable/disable : messageSetName : wmeRegisterType : TxDirection : PSC
# enable/disable = 0 or 1
# wmeRegisterType = USER or PROVIDER
# TxDirection is between the application and the WSMP stack:
#   TX    = routing application message to over the air message
#   RX    = routing over the air messages to the application
#   TX_RX = application sending and also receiving the same message set broadcast
#            from other parties, such as other intersection’s SPaT, etc.
#
application   1:BSM:USER:RX:MMITSS
application   1:SRM:USER:RX:MMITSS
application   1:MAP:USER:TX:MMITSS
application   1:SPaT:USER:TX:MMITSS
application   1:SSM:USER:TX:MMITSS
#
# Socket configuration:
#    protocol : hostname : port
#
listenSocket  UDP:192.168.0.150:15001 # for MessageTX
sendSocket    UDP:192.168.0.166:15000 # for MessageRX
```

4.6.6 OBE_Aware

The *OBE_Aware* is the core MMITSS OBE component. It is responsible for:

- Acquiring vehicle geolocation (latitude, longitude and elevation) and movement state (speed and heading) data from the on-board GPS receiver;
- Forming BSM messages and sending the BSMs to the *OBE_MessageTX* component;
- Processing the received MAP messages, tracking the vehicle on the MAP and estimating the expected time-to-arrive (ETA) at the intersection stop-line;
- Processing the received SPaT messages to obtain the state and the remaining time of the signal that controls the movement of the vehicle; and
- When it is eligible for priority, forming SRM messages and sending the SRMs to the *OBE_MessageTX* component.

Figure 18 presents an activity diagram of the *OBE_Aware* component.

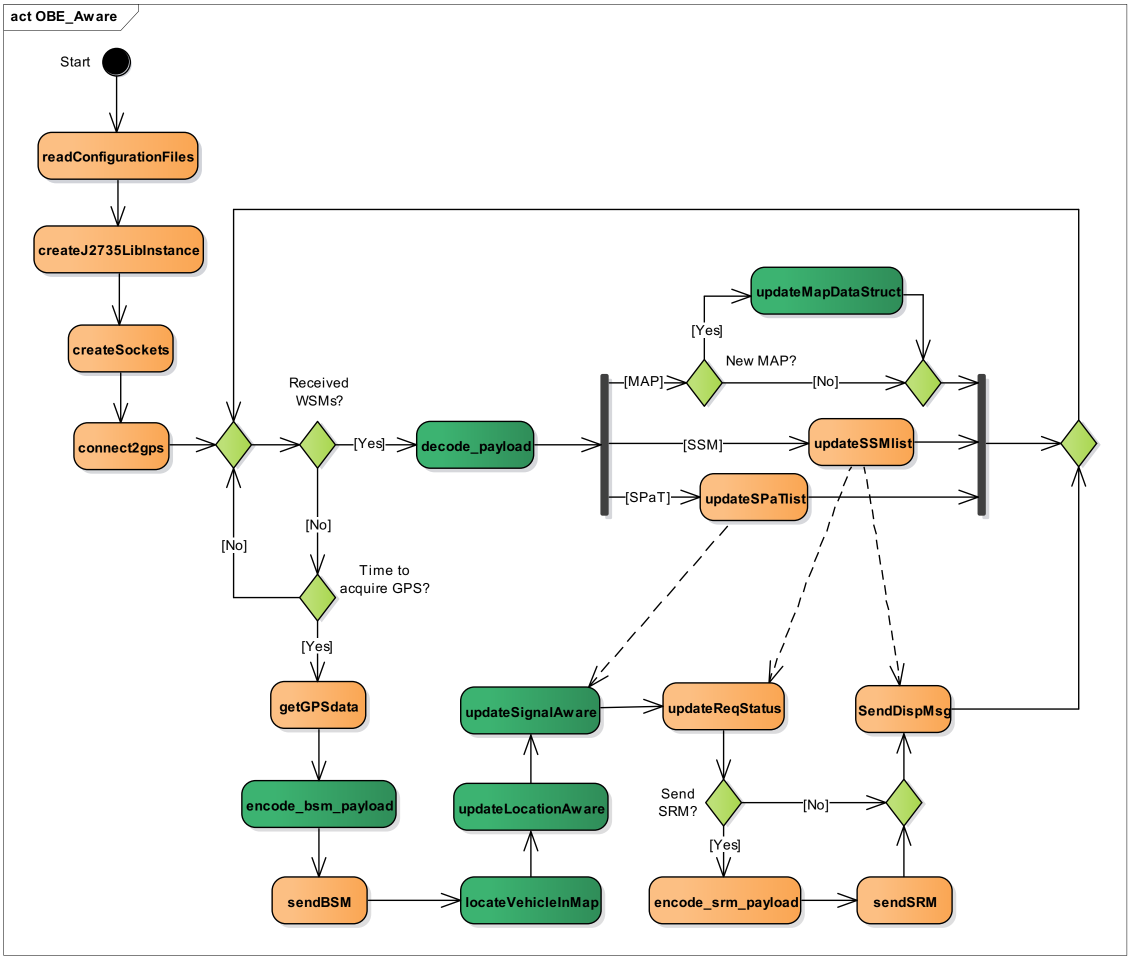


Figure 18. Activity Diagram of OBE_Aware

The *OBE_Aware* component maintains lists of received MAP, SPaT and SSM messages, one (latest) record per intersection. This component acquires GPS data at a 10Hz rate, forms a BSM, and sends to the *OBE_MessageTX* component which broadcasts the message over the air. It uses the GPS data to locate the vehicle on the MAP (determining *intersectionID*, laneID and the projection point on the lane), and calculates the distance to the intersection stop line and the expected arrival time. It finds the intersection SPaT from the SPAT list and gets the state and the remaining time of the signal that controls the vehicle's (lane) movement, and finds the intersection SSM from the SSM list and checks whether it needs to send/update the priority request. When needed, it forms an SRM and sends to the *OBE_MessageTX* component. When detected the vehicle has passed the stop line, this component sends a cancel priority SRM.

4.6.7 MRP_DataMgr

The *MRP_DataMgr* component is responsible for bridging messages between the *MRP_TrafficControllerInterface* component and other MMITSS components. The functionalities of this component include

- Receive and store data from the linked components;

- Encode SPaT message and send it to the *MRP_MessageTX*, *MRP_*AWARE and the Nomadic_*PriorityDataServer* (The *MRP_TrafficControllerInterface* component sends the unencoded SPaT data to the *MRP_DataMgr*);
- Send encoded MAP messages to the *MRP_MessageTX* component and the *Nomadic_PriorityDataServer* component;
- Forward or send poll response to the linked components; and
- Timestamp all data for use in synchronizing actions and logging data for post-processing.

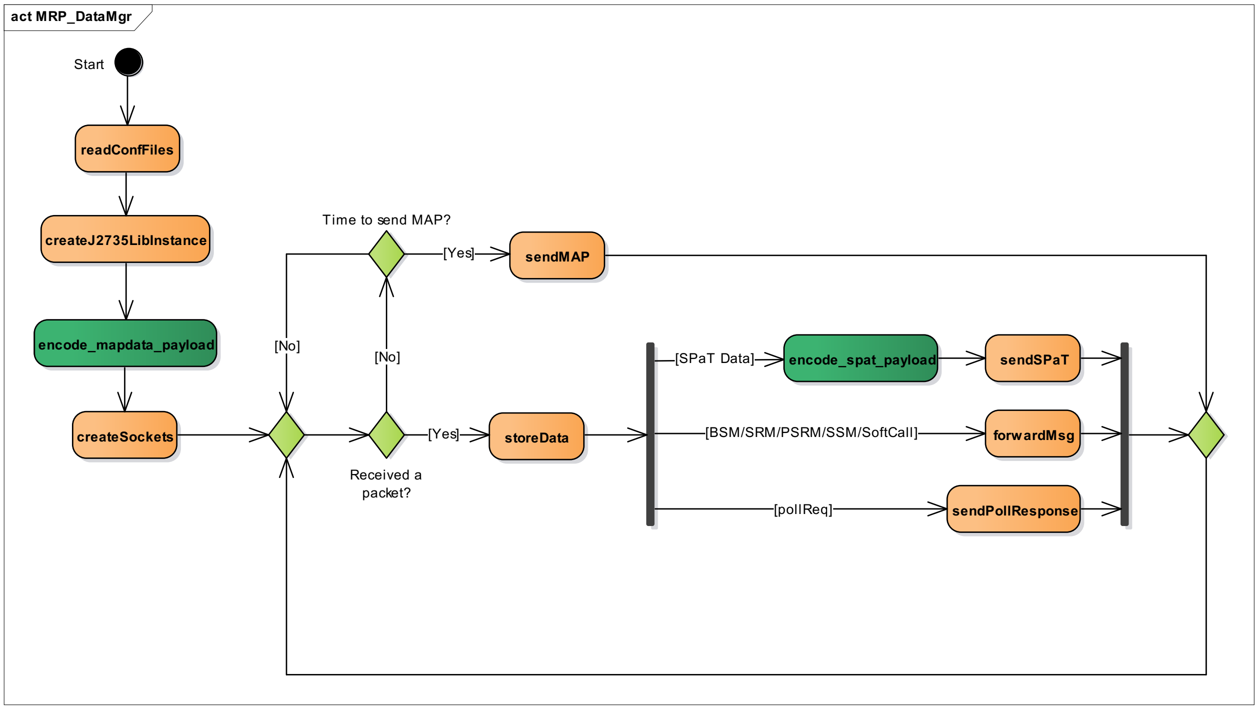


Figure 19. Activity Diagram of MRP_DataMgr

Figure 19 presents an activity diagram for the *MRP_DataMgr*. The message type and associated action of this component is summarized in Table 9.

Table 9. MRP_DataMgr Actions on Messages

Message Type	Message ID	From	Action
BSM	0x40	RSE_MessageRX	Forward to MRP_Aware
SRM	0x41	RSE_MessageRX	Forward to MRP_Aware
PSRM	0x42	Nomadic_PriorityDataServer	Forward to MRP_Aware
SSM	0x43	MRP_Aware	Forward to RSE_MEssageTX and Nomadic_PriorityDataServer
Unencoded SPaT	0x44	MRP_TrafficControllerInterface	Encode SPaT, send it to RSE_MEssageTX, Nomadic_PriorityDataServer and MRP_AWARE
MAP	0x45	MRP_DataMgr	Send to RSE_MEssageTX and Nomadic_PriorityDataServer
SoftCall	0x46	MRP_Aware	Forward to MRP_TrafficControllerInterface
VehTraj	0x47	MRP_Aware	Store in memory, send to the requested when polled
PhaseFlags	0x48	MRP_TrafficControllerInterface	Store in memory, send to the requested when polled
PhaseTiming	0x49	MRP_TrafficControllerInterface	Store in memory, send to the requested when polled
CoordPlan	0x50	MRP_TrafficControllerInterface	Store in memory, send to the requested when polled
FreePlan	0x51	MRP_TrafficControllerInterface	Store in memory, send to the requested when polled
DetPhaseAsgmt	0x52	MRP_TrafficControllerInterface	Store in memory, send to the requested when polled
DetectorCount	0x53	MRP_TrafficControllerInterface	Store in memory, send to the requested when polled
DetectorPresence	0x54	MRP_TrafficControllerInterface	Store in memory, send to the requested when polled
PollReq	0x80	Other components such as MRP_PerformanceObserver	Send back the requested message

4.6.8 MRP_TrafficControllerInterface

The *MRP_TrafficControllerInterface* component is responsible for providing the protocol specific interface to the 2070 traffic signal controller with AB3418 protocol. The functionalities of this component include:

- Collect traffic controller's configuration data (e.g. phase timing, coordination plans, free plan, detector phase assignment, etc.);
- Collect signal status data (e.g. active phases and intervals, etc.) from the traffic controller and populate SPaT data for constructing J2735 SPaT;
- Collect vehicle detection data (e.g. count and occupancy) from the traffic controller; and
- Command MMITSS traffic and priority control (*SoftCall*) to the traffic controller.

Figure 20 presents an activity diagram of the *MRP_TrafficControllerInterface*.

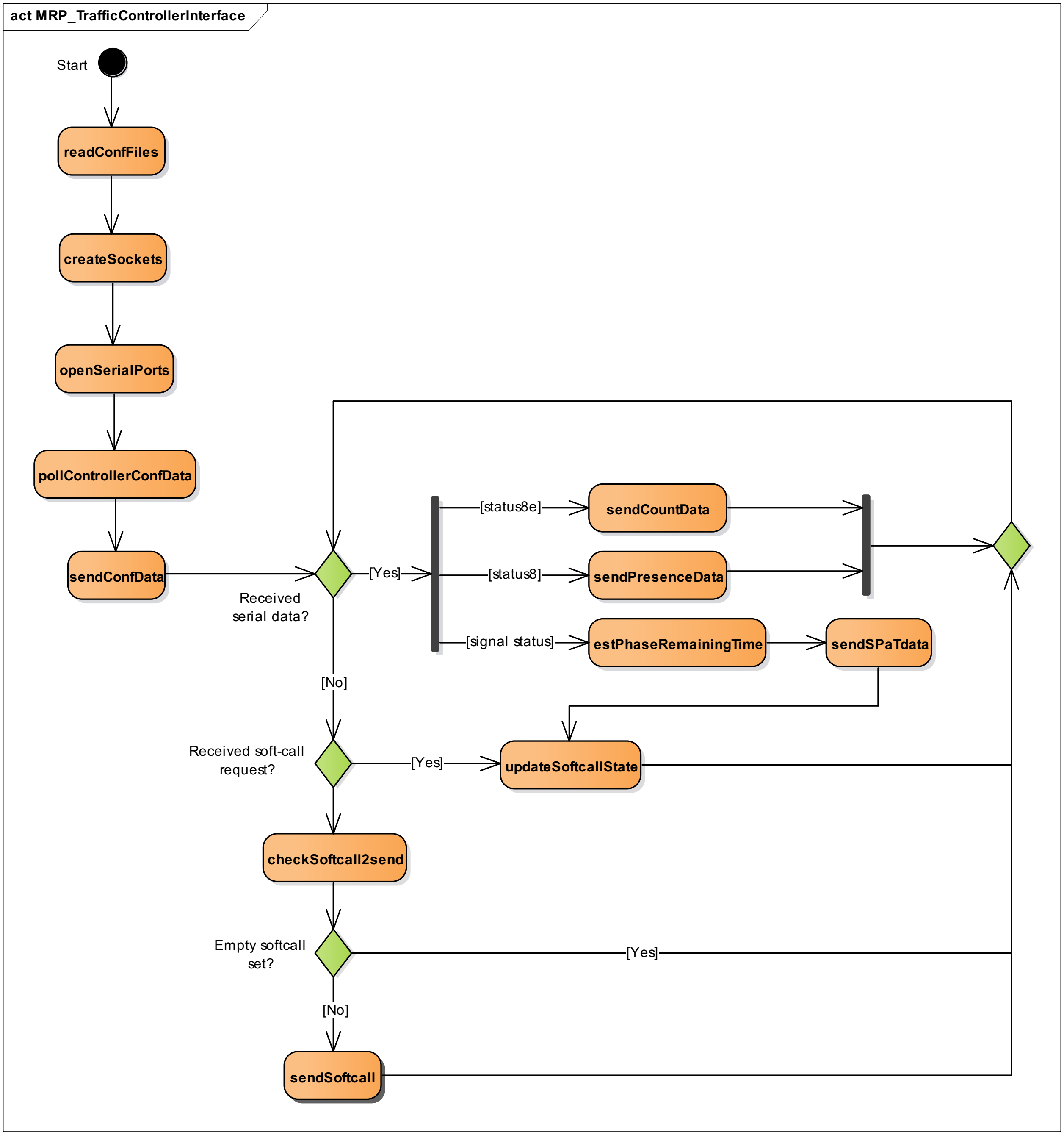


Figure 20. Activity Diagram of MRP_TrafficControllerInterface

This component collects controller configuration data, vehicle count data, and detector presence data, and sends the data to the *MRP_DataMgr*. When received *signal status* data from the traffic controller, this component estimates the remaining time for all permitted phases, and sends the SPaT data to the *MRP_DataMgr*. The estimation is a combination of *signal status* data, phase timing data, and coordination/free plan parameter, and follows NEMA dual-ring control logic.

This component also updates a list of vehicle, pedestrian and priority calls when it received soft-call requests (including cancel request) from the *MRP_Aware* via the *MRP_DataMgr*. Pedestrian calls are sent to the traffic controller right away and removed from the list. Vehicle phase calls and early green priority calls are only sent when the calling phase is in red or yellow. These calls are dropped and removed from the list when either the calling phase has turned to green or this component has received a cancel soft-call request. Vehicle green extension and priority green extension calls are only sent when the calling phase is in green, and are removed from the list when either the phase has turned yellow or the request has been cancelled.

4.6.9 MRP_Aware

The *MRP_Aware* component is the core component for MMITSS traffic and priority control. It is responsible for processing BSM, SRM and PSRM messages, managing requests for priority, and integrating equipped vehicles into the traffic control logic through phase calls and phase extensions for each of the different modes (and dynamics) of vehicles. This component is also responsible for providing SSMs to equipped vehicles and travelers.

Figure 21 presents an activity diagram of the *MRP_Aware* component. This component maintains two lists, a list of active requests for priority and a list of vehicle trajectories. When received a BSM, depending on whether the vehicle ID is on the vehicle list or not, either a new track is initiated or an existing track get updated. When received a SRM or a PSRM, it updates the request list.

In the decision of making soft-call request, this component checks every vehicle on the priority request list as well as every vehicle on the vehicle list. Rules for selecting soft-call target are as follows:

- Pedestrian phase call will be sent to the *MRP_TrafficControllerInterface* component via the *MRP_DataMgr* right away;
- Vehicle phase call will be sent if the requested phase is not in green;
- Request for priority has higher priority than vehicle phase extension calls; and
- Green extension priority request has higher priority than early green request, as priority treatments are provided to the main streets, green extension always reduce the total delay more than the early green treatment.

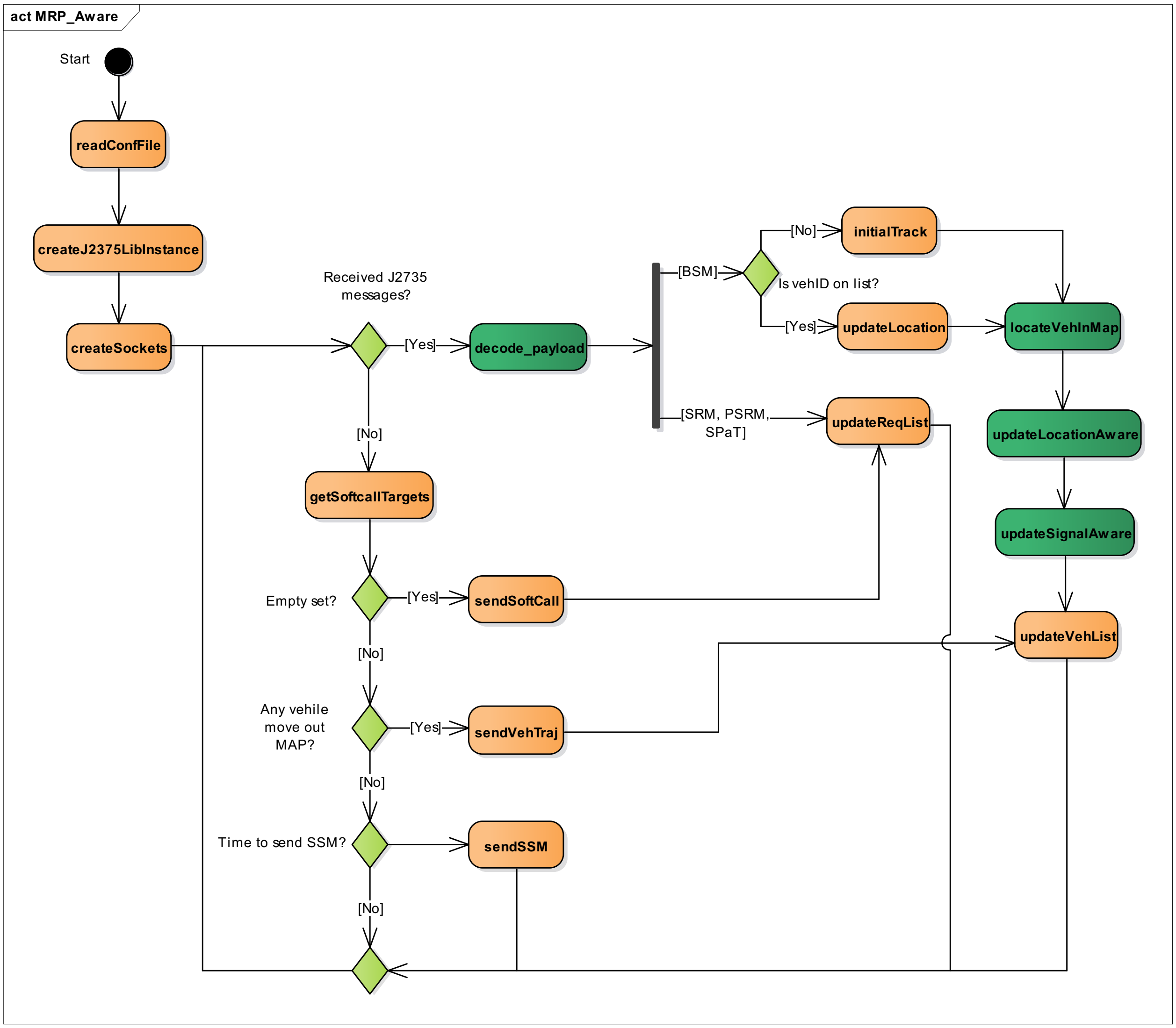


Figure 21. Activity Diagram of MRP_Aware

4.6.10 MRP_PerformanceObserver

The *MRP_PerformanceObserver* component is responsible for periodically acquiring CV trajectory data and traffic detector data from the *MRP_DataMgr* component, estimating intersection level performance measures, and sending the estimated measures to the *MRP_DataMgr* such that the measures become available to other MMITSS components.

The intersection level performance measures are estimated by this component and the data resource regarding each measure are summarized in Table 10.

Table 10. Performance Observations and Associated Data Resources

Performance Metric	Abbreviation	Unit	Data Source
Travel Time by Movement and Mode	TT	Second	BSM, SRM → Vehicle trajectory (MRP_Aware)
Travel Time Variability by Movement and Mode	TTV	Second	BSM, SRM → Vehicle trajectory (MRP_Aware)
Delay by Movement and Mode	DT	Second	BSM, SRM → Vehicle trajectory (MRP_Aware)
Delay Variability by Movement and Mode	DTV	Second	BSM, SRM → Vehicle trajectory (MRP_Aware)
Vehicle Throughput by Movement	VT	Vehicles per Hour (vph)	Detector count →Movement-based throughput (MRP_TrafficControllerInterface)

4.6.10.1 Implementation of California MMITSS

4.6.10.1.1 J2735 Library and Locating Vehicle on MAP

The data elements, data frames, and messages in SAE J2735 message sets over DSRC are defined in terms of a formal language called Abstract Syntax Notation One (ASN.1). The J2735 dictionary standard also calls for use of the Distinguished Encoding Rules (DER) to translate the ASN.1 into a concrete data stream. The WSM payload transmitted over-the-air is always encoded in the ASN DER style. Savari RSE and OBE do provide a J2735 library for encoding and decoding J2735 messages. However, applications running on the MRP cannot directly use this library and this library is not portable to the MRP. The California implementation therefore developed its own library for encoding and decoding J2735 messages, including BSM, SRM, SPaT, MAP and SSM.

Both the MRP and the OBE components require a common functionality of locating a vehicle on a MAP. The OBE component requires this functionality to associate SPaT messages with MAP messages (an OBE could simultaneously receive SPaT and MAP messages from multiple intersections), and when it is eligible for signal priority, to generate signal request messages (SRMs). The MRP component requires this functionality for MMITSS traffic/priority control and performance measures. The California implementation chose to implement this function as part of the J2735 library so the same J2735 message encoding/decoding and vehicle tracking functions can be used on both the MRP and OBE.

Encoding/decoding J2735 messages was developed based on the J2735 ASN.1 Specification J2375_200911 available at the time of this development. The ASN.1 Specification has been updated to a newer version J2375_201601. The developed J2735 library does need to update to be consistent with the latest SAE J2375 standard.

The flow diagram of *locateVehicleInMap* function is illustrated in Figure 22.

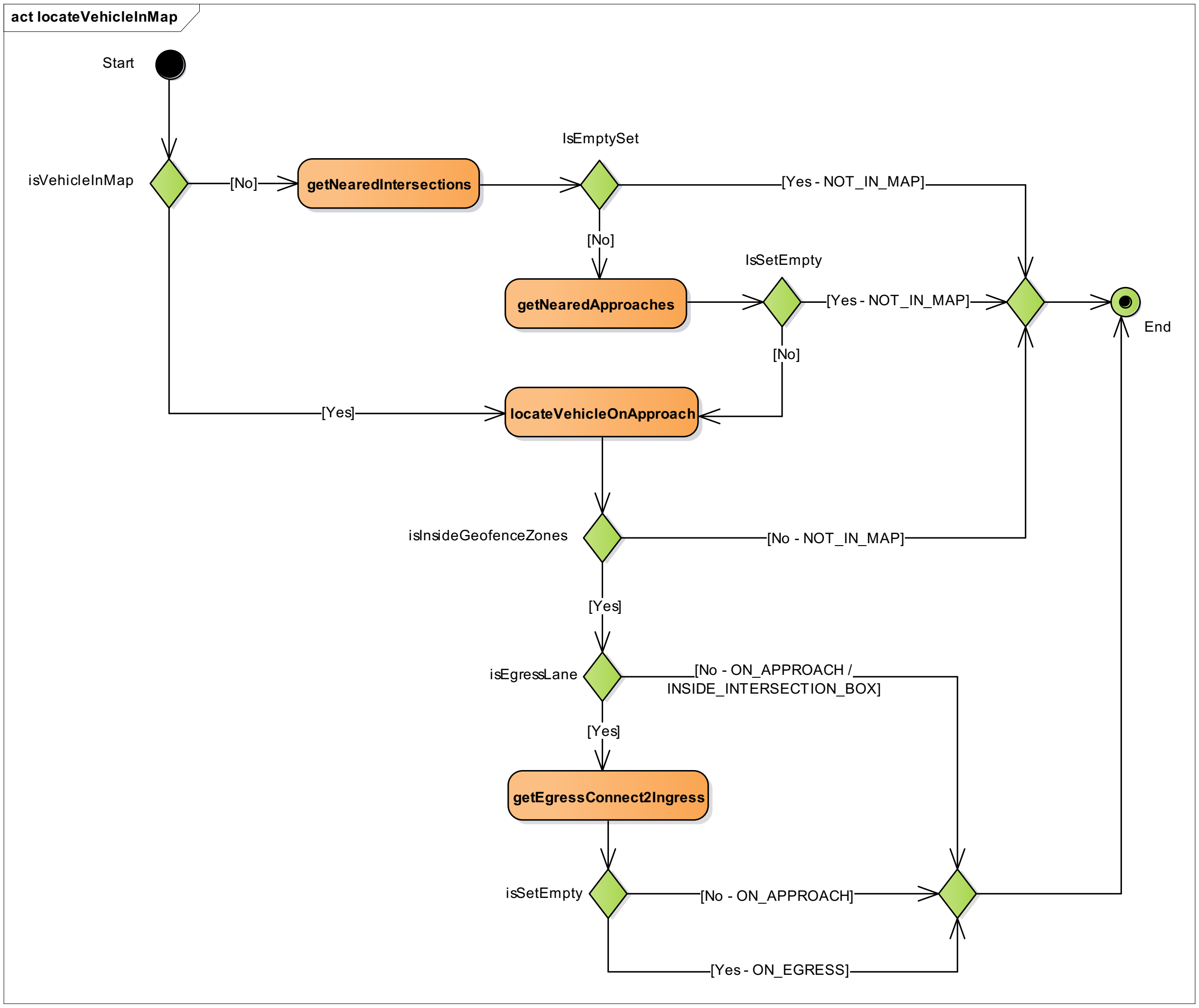


Figure 22. Flow Diagram of Locating on MAP Algorithm

Steps for initiating a track are described below:

- I1. Find the set of intersections that the vehicle is inside the intersection MAP, by checking the distance from the vehicle geolocation to the intersection reference point against the intersection MAP *radius*;
- I2. For each of the intersections that the vehicle is inside, find the convex hulls that the vehicle is inside;
- I3. For each of the convex hulls that the vehicle is inside, project vehicle geolocation onto the lanes, and get the *laneID* and the projected point that has the minimum distance from the vehicle geolocation to the lane among all lanes on the approach. Vehicle speed and heading are used for the point-to-line projection to ensure vehicle heading is along the lane node heading; and
- I4. In the case that the vehicle geolocation is projected onto two MAPs (when there is an overlap between the egress lane of an upstream intersection and the ingress lane of the downstream intersection), return the projection result on the ingress lane.

Steps for maintaining an established track are as follows:

M1. Start with the last projection result (e.g. *intersectionID*, *laneID* and the projected point along the lane) and move the projection forward along the lane. The same point-to-line method as step I3 above is used. As a vehicle could change lanes, the projection lane is determined as the one that has the minimum distance from the vehicle geolocation to the lane. Moving forward of a projected point follows the transition of *vehicleIntersectionStatus* state as illustrated in Figure 23; and

M2. When the projection lane is an egress lane, repeat steps I1 to I3 above to determine whether the vehicle is also on an ingress lane of another intersection. If the vehicle is detected also on an ingress lane of the downstream intersection, return the projection result on the ingress lane, otherwise return the project result on the egress lane of the current intersection. At closely spaced intersections, this step ensures a quick detection of state transit from ON_EGRESS at an upstream intersection to ON_APPROACH at the downstream intersection.

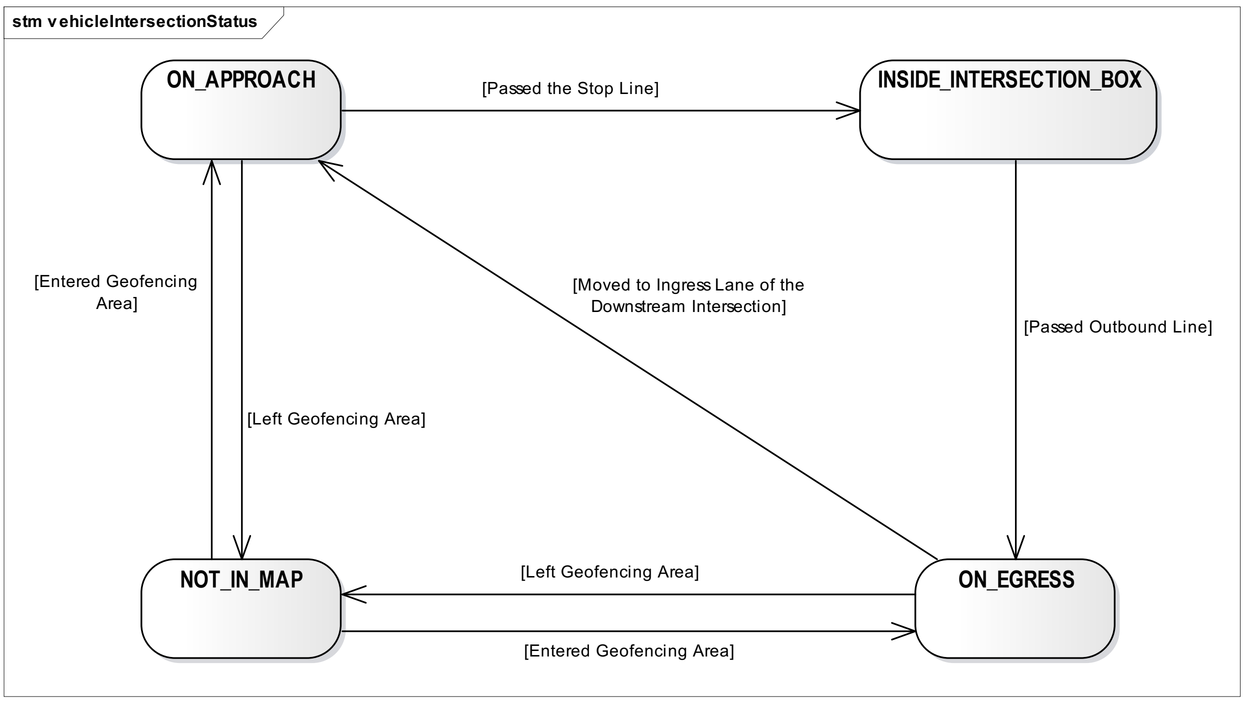


Figure 23. Vehicle Tracking State Transition Diagram

4.6.10.2 MAP Description File

The MMITSS team defined the format of the intersection MAP description file that contains all the information needed for constructing J2735 MAP messages, as follows:

```
MAP_Name        Intersection_Name.nmap
RSU_ID          Intersection_Name
MAP_Version     xxx
IntersectionID  xxxx
Intersection_attributes xxxxxxxx  /*Attributes of the intersection, 8 bits */
Reference_point xx.xxxxxxx  xxx.xxxxxxx xxxx   /*lat, long, evelation(decimeter)*/
No_Approach     xx  /*number of approaches*/
Approach        1
Approach_type   x  /*1: approach, 2: engress, 3: barrier, 4: crosswalk*/
No_lane         x
Lane            1.1 x /*x – the signal phase that controls the lane movement*/
Lane_ID         1
Lane_type       x     /* Veh Lane, Computed Lane, Ped Lane, Special Lane, Barrier*/
Lane_attributes xxxxxxxxxxxxxxxx  /*Attributes of the lane, 16 bits*/
Lane_width      xxx   /*in centimeters */
No_nodes        x     /*number of lane nodes*/
1.1.1     xx.xxxxxxx   xxx.xxxxxxx  /*latitude longitude*/
1.1.2     xx.xxxxxxx   xxx.xxxxxxx
……
No_Conn_lane    x     /*connection lanes*/
x.x x
x.x x
end_lane
Lane            1.2 x /*x – the signal phase that controls the lane movement*/
……
end_lane
end_approach
Approach        2
……
end_approach
end_map
```

A MAP description file was created for each of the intersections in the California testbed using Google Earth tools to collect the coordinates of the intersection reference point (e.g. the center of the intersection) and lane nodes (a sequence of way-points along the center of each traffic lanes and pedestrian crosswalks).

4.6.10.3 UNIX Socket API

The data transmission between each component is implemented by UNIX sockets. The California implementation wraps the basic socket functions (e.g. creating, connecting, sending and receiving) into the *socketUtils* class (see Figure 24) to reduce the redundant socket coding on each component. It supports both UDP and TCP sockets for IPv4 and IPv6.

The *setSocketAddress* function converts the configuration string into the *socketAddress_t* object for creating a socket. For receiving data, the *server* function creates a socket to get a socket descriptor, binds the socket to the local socket address, and starts listening on the port for incoming TCP connections. For sending date, the *client* function creates a socket and connects the socket to the host socket address. The argument *isNonblocking* is to set the socket in non-blocking (*true*) or blocking (*false*) mode. In blocking mode, the socket receive commands block the process until a packet arrives. All the MMITSS components require non-blocking sockets so the component does not wait for the packets and only process the packet when it arrives. Socket send commands might not send all the bytes that are asked to. The *sendall* function ensures that all the bytes are sent out so the component that uses the Socket API does not need to handle sending partial messages.

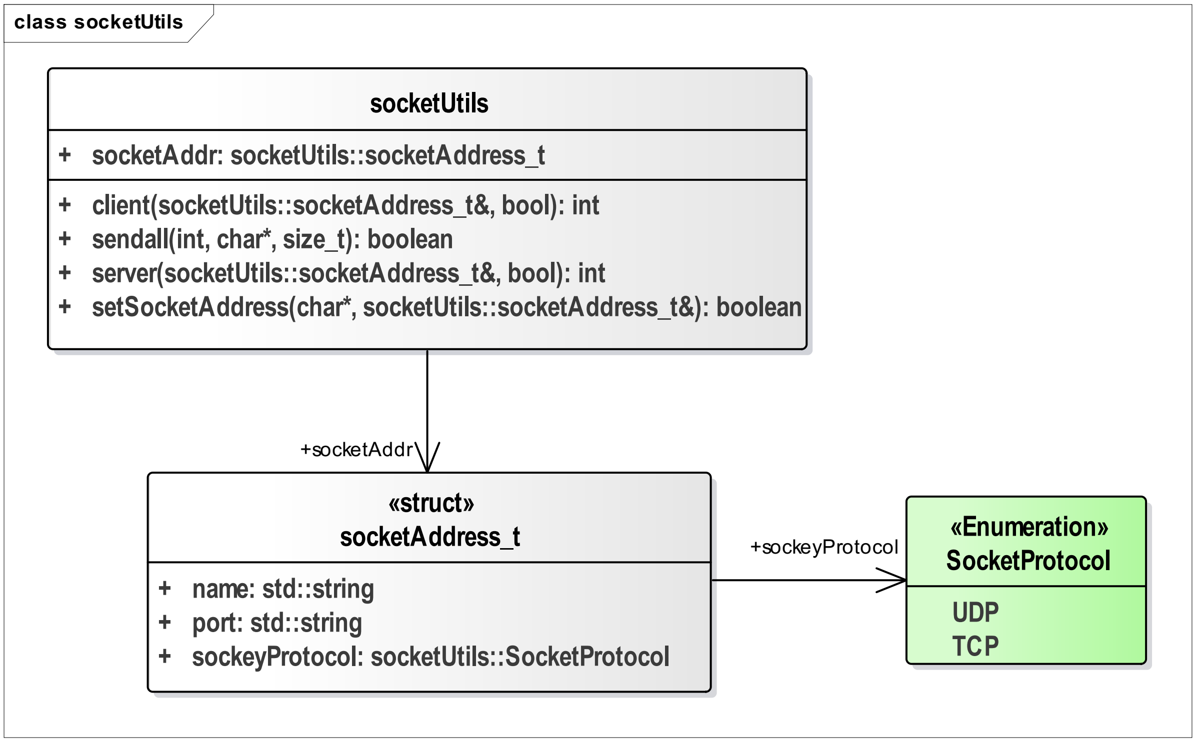


Figure 24. Class Diagram of UNIX Socket API

4.6.10.4 MMITSS Message Header

A single component may receive different types of messages over socket. For example, the *RSE_MessageTX* receives SPaT, MAP and SSM messages to be transmitted over the air. One option to handle multiple types of messages is to use multiple receiving sockets with each socket corresponding to one type of message. Another option is to include a message header that specifies the type of message so only one receiving socket would be required. California MMITSS followed the second option. The format of message header is illustrated in Figure 25. The *MessageClass* field classifies that this is an MMITSS internal message (selected as 0xFFFF), the *MessageID* field identifies the type of message (e.g. *Data*) and the *Length* field specifies the length (in bytes) of *Data*.

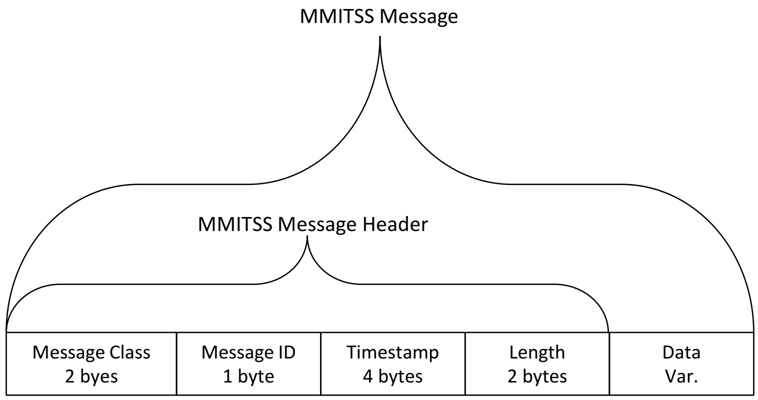


Figure 25. MMITSS Internal Message Format

Table 11. MMITSS Messages

Message Type	Message ID	From	Via	To
BSM	0x40	RSE_MessageRX	MRP_DataMgr	MRP_Aware
SRM	0x41	RSE_MessageRX	MRP_DataMgr	MRP_Aware
PSRM	0x42	Nomadic_PriorityDataServer	MRP_DataMgr	MRP_Aware
SSM	0x43	MRP_Aware	MRP_DataMgr	RSE_MEssageTX Nomadic_PriorityDataServer
SPaT	0x44	MRP_TrafficControllerInterface	MRP_DataMgr	RSE_MEssageTX Nomadic_PriorityDataServer MRP_Aware
MAP	0x45	MRP_DataMgr	-	RSE_MEssageTX Nomadic_PriorityDataServer
SoftCallReq	0x46	MRP_Aware	MRP_DataMgr	MRP_TrafficControllerInterface
VehTraj	0x47	MRP_Aware	MRP_DataMgr	Message requester via poll
PhaseFlags	0x48	MRP_TrafficControllerInterface	MRP_DataMgr	Message requester via poll
PhaseTiming	0x49	MRP_TrafficControllerInterface	MRP_DataMgr	Message requester via poll
CoordPlan	0x50	MRP_TrafficControllerInterface	MRP_DataMgr	Message requester via poll
FreePlan	0x51	MRP_TrafficControllerInterface	MRP_DataMgr	Message requester via poll
DetPhaseAsgmt	0x52	MRP_TrafficControllerInterface	MRP_DataMgr	Message requester via poll
DetectorCount	0x53	MRP_TrafficControllerInterface	MRP_DataMgr	Message requester via poll
DetectorPresence	0x54	MRP_TrafficControllerInterface	MRP_DataMgr	Message requester via poll
PollReq	0x80	Other components such as MRP_PerformanceObserver	-	MRP_DataMgr

MMITSS message ID and message flow are summarized in Table 11. SRM and PSRM (pedestrian SRM) have the same message format and are sent from different sources.

4.6.10.5 OBE_GUI

For the purpose of testing, debugging and demonstration of MMITSS functions, an *OBE_GUI* was developed to visualize the status of the vehicle, the communication system, traffic signals, and priority control services.

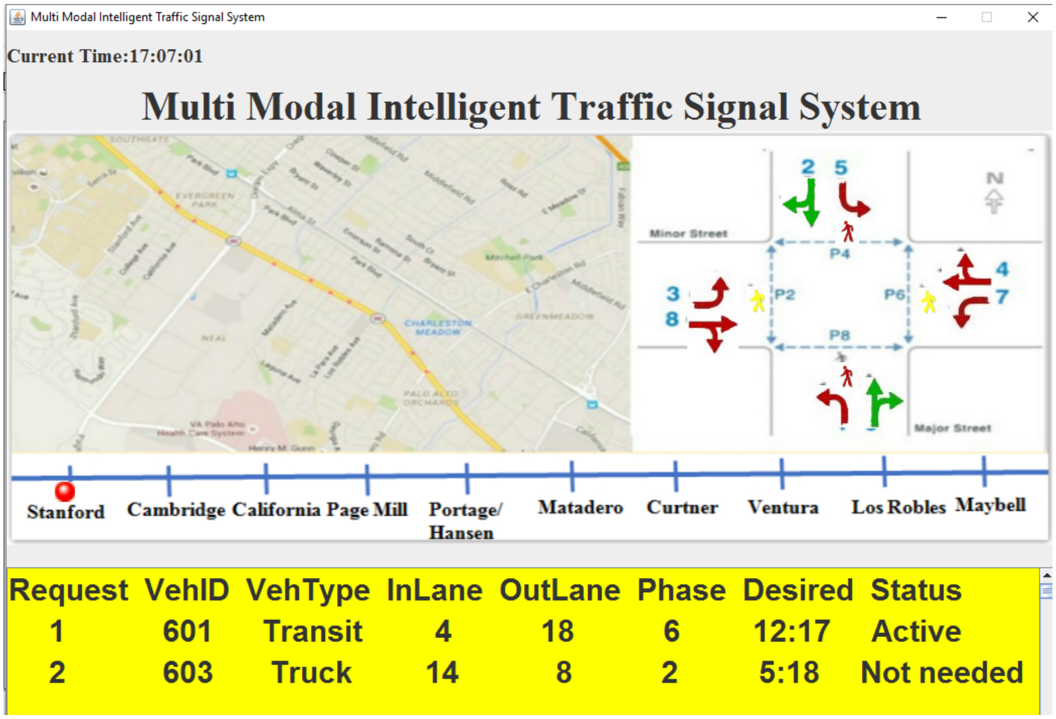


Figure 26. Example of OBE_GUI Interface

The *OBE_GUI* interface is shown in Figure 26. It consists of several frames. OBE's current location is displayed on the top-left Google Map. SPaT information of the approaching intersection is displayed on the top-right frame with colored movements and pedestrian signs. The middle frame lists all the MMITSS intersections in the California testbed with a red dot indicating the approaching intersection. The bottom frame lists the "Current Active Request Table", which shows the status of all priority requests that are at the approaching intersection. In the example shown in Figure 26, the OBE represents a transit vehicle (ID 601). It is traveling northbound on El Camino Real and heading to the intersection at Stanford Ave. Another OBE (ID 603 representing a truck) is traveling on southbound of El Camino Real and also heading to the Stanford Ave. intersection. Both OBEs have requested signal priority. The Stanford intersection is serving phase 2 and 6 (along El Camino Real) with the Flashing Don't Walk pedestrian interval. Green extension priority is granted to the bus (ID 601) as the remaining pedestrian interval (7 seconds) is shorter than the service time (12 seconds) required by the bus.

4.7 Savari SmartCross System

Both the California and Arizona MMITSS implementations utilized the Savari SmartCross system for pedestrian service. The SmartCross system consists of a SmartCross application running on Android phones and a Nomadic Traveler Server. It uses cellular technology for communications between a nomadic device (SmartCross app) and the Nomadic Traveler Server, and uses backhaul for communications between the Nomadic Traveler Server and the intersection MRP as illustrated in Figure 27.

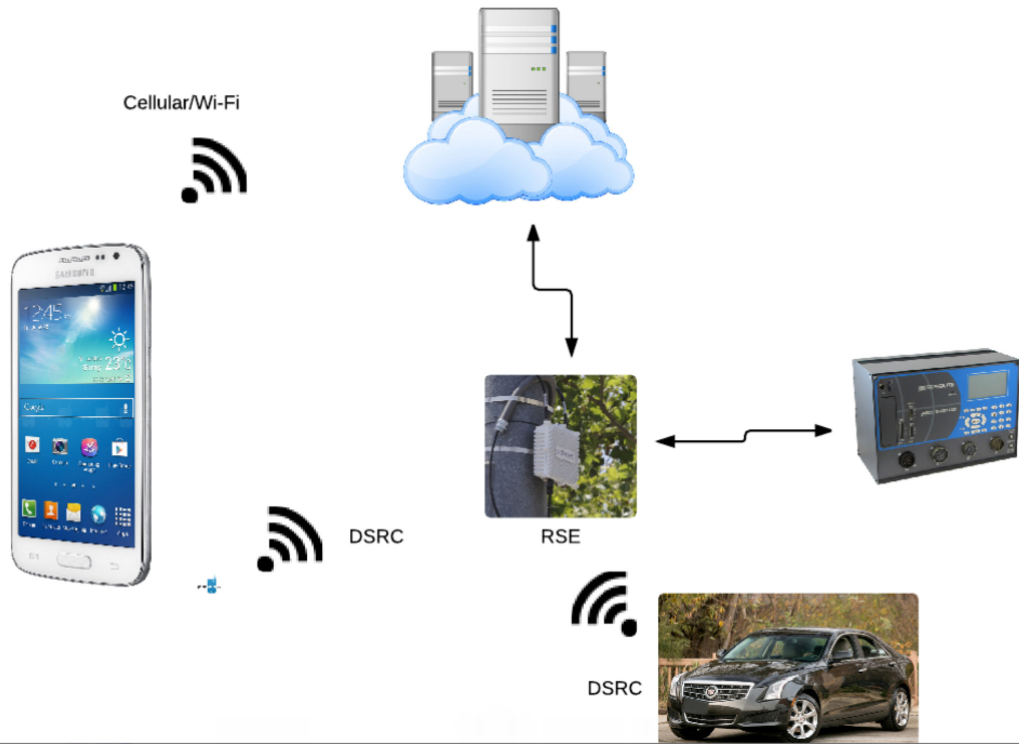


Figure 27. Savari SmartCross System Architecture

4.7.1 Savari SmartCross App

The main purposes of the SmartCross app include

- Providing signal timing information in the form of audio, visual and haptic alerts to the user based on his/her location and phone orientation, and
- Enabling pedestrian service request.

The application receives the SPaT and MAP information from the Nomadic Traveler Server, and sends the pedestrian SRM to the associated MRP via the Nomadic Traveler Server.

Figure 28 illustrates the UI of the SmartCross app. Each button shown on the UI corresponds to one operation mode of the app. If the phone is used in accessibility mode, each button will have a Talkback mode that provides spoken feedback based on where the screen is touched.

Figure 28. SmartCross App User Interface

Figure 29 shows the UI corresponding to different modes of operation. The left mode in Figure 29 is for general pedestrians. The interface provides the time remaining in seconds if the pedestrian phase interval is active, and a blue “Press to Cross” button for the user to request a Walk phase. When the button is pressed and the pointing direction of the phone aligns with the crosswalk, the app sends out a Signal Request Message. If the button is pressed while not facing a crosswalk, an audio alert (“You are not facing a crosswalk”) is spoken back to the user indicating that he/she needs to adjust the pointing direction of the phone to align with the crosswalk.

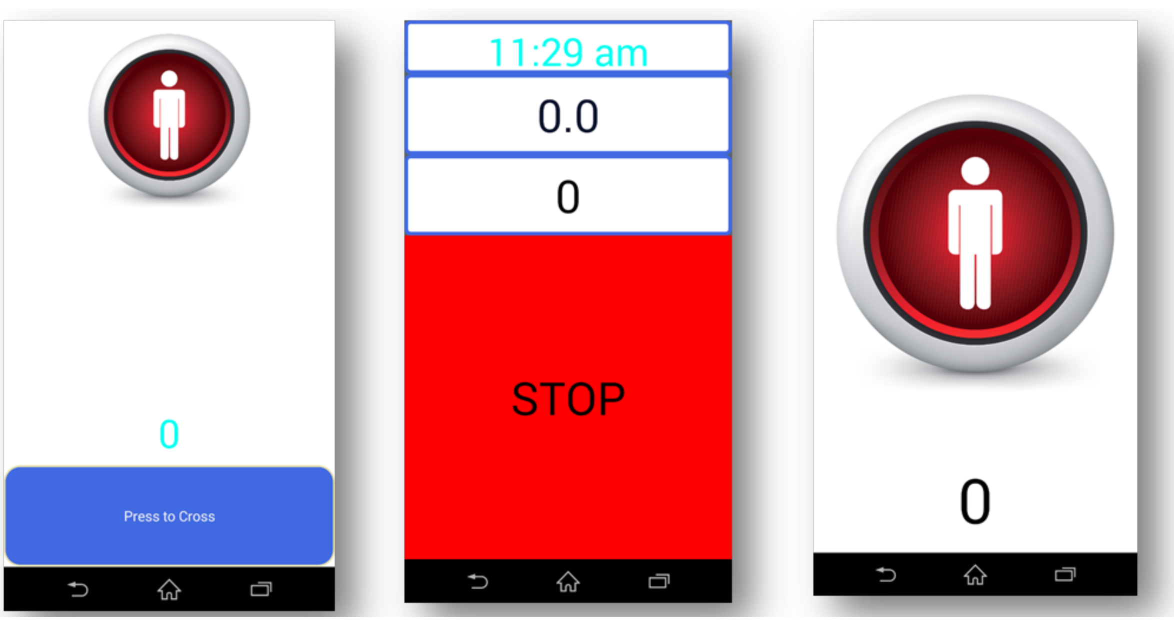


Figure 29. Ped Mode, Bicycle Mode, and Disabled Traveler Mode

For the visually impaired and mobility impaired mode, the underlying operation is the same as general pedestrians. The interface is modified to eliminate the cross button. Instead, if the user wants to request a crosswalk phase, he/she can do so by pressing and holding any part of the screen for 3 seconds or until an audio notification is heard.

In addition to the above operation, multiple scenarios are addressed by the application to enhance safety.

1. If the user enters a crosswalk when the crosswalk is not active, an audio alert ("Do not walk") and a strong haptic alert are triggered.
2. When the user is in an active crosswalk, the remaining time along with the pedestrian signal state and audio beeps accompanied by haptic feedback. The audio beep frequency and haptic feedback strength are different for WALK and Flashing Don't Walk (FDW).
3. When the system is unreliable, a notification communicating that the system is unreliable and should not be used.

4.7.1.1 Nomadic_PriorityDataServer

This component is responsible for receiving pedestrian location messages (PLMs) from the nomadic devices, receiving SPaT and MAP messages from RSEs, and relaying SPaT, MAP and pedestrian SRM messages between a nomadic device and the corresponding intersection MRP.

This component receives PLMs from all application users at all connected intersections Based on their location and the MAP data, this component matches user locations with the intersections, and transmits the corresponding intersection SPaT and MAP to the appropriate users. When receiving an SRM from the SmartCross app, this component relays the SRM to the appropriate MRP. The SSM received from the MRP is also relayed back to the SmartCross app user.

4.8 MMITSS Functions Not Implemented

Several of the originally identified use cases (See Table 1) were not implemented in the development and field testing effort as originally planned. These include:

- Use Case 11.1.2: Coordinated Section of Signals
- Use Case 11.2.3: Enhanced TSP
- Use Case 11.4.2: Coordinated Freight Signal Priority along an Arterial

The Coordinated Section of Signals use case is defined in terms of identification of platoons that are tracked through a section of 5 to 7 traffic signals. The progression quality, such as early arrival on green or arrival at the back of a queue, is monitored by the section or system level Coordination Control and the parameters of the coordinated plan are adjusted to better accommodate the movement of platoons. The CA MMITSS implementation did use coordination and did adjust the offset and phase splits, but didn't utilize information about platoons. The AZ MMITSS implementation did not utilize coordination, which is considered as a special form of priority request that can be adapted per the movement of platoons.

The Enhanced Transit Signal Priority use case provided a route based progression for a priority eligible and active transit vehicle using either peer-to-peer communications or through a section or system level route based priority component.

The Coordinated Freight Signal Priority along an Arterial use case is similar to, and should be implemented through the same mechanism as, the Enhanced TSP use case. Both of these use cases could be address through a section or system level component, or through a distributed peer-to-peer process.

In addition, the WAVE Service Advertisement (WSA) and Security Services were not implemented. The WSA (as described in Section 4.9.2) requires a PSID

4.9 DSRC Standards

4.9.1 Overview of WAVE Communications

WAVE is a means of communicating between vehicles (OBEs) and between roadside systems and vehicles. WAVE messaging uses the DSRC protocol, originally standardized as IEEE 802.11p-2010, and is now incorporated in IEEE 802.11-2012. When within the communication zones, the applications of MMITSS running on the MRP and OBE exchange information with each other to perform the MMITSS Traffic Control service.

In the WAVE environments, the *RSE_ServiceAdvertisementMgr* broadcasts WAVE service advertisement (WSA) messages on the control channel (CCH 178) to announce the presence of MMITSS traffic control service and the service channel (SCH 182) to be used for exchanging the service data (e.g. SRM - Signal Request Message and SSM - Signal Status Message). Basic safety messages (BSMs), signal phase and timing (SPaT) messages and MAP messages are expected to be transmitted on the SCH 172. Thus, no WSA would be needed for these three types of messages.

MMITSS service users (e.g. vehicles that are eligible for signal priority) monitor the WSA. If desired, they can participate in the MMITSS traffic control service by sending signal request messages and receiving signal status messages which acknowledge the status of the received SRMs at the intersection. The *RSE_SecurityCertificateService* manages trust between OBEs and the RSEs to ensure that only the trusted, eligible OBEs/ASDs can join the MMITSS traffic control service.

Over-the-air data exchange between OBEs and RSEs is managed by the *OBE/RSE_MessageTX* and *OBE/RSE_MessagesRX*, using the WAVE Short Message Protocol (WSMP). Over-the-air messages are transmitted in the forms of WAVE Short Messages (WSMs). The WSM format consists of a WSMP header followed by a variable-length payload (WSM data) as illustrated in Figure 30. The mandatory PSID (Provider Service Identifier) identifies the service that the WSM payload (e.g., BSM, SRM, MAP, SPaT, SSM, etc.) is associated with. Each type of payload must have a unique PSID so the receiving end knows how to process it.

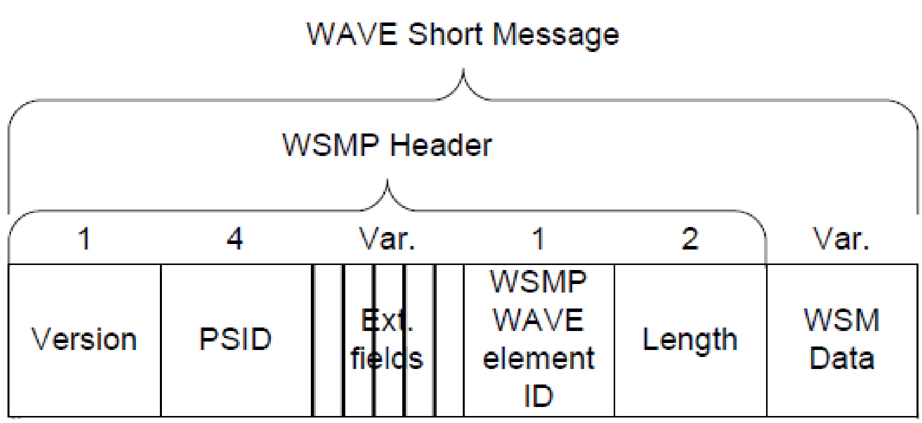


Figure 30. WAVE Short Message Format (Source: IEEE 1609.3-2010)

The message sets transmitted between the OBE and the RSE and their channel configurations are summarized in Table 12. For the MMITSS prototype, channel 182 has been selected as the operating channel for transmitting SRMs and SSMs.

Table 12. Message Sets Transmitted between the OBE and the RSE

Message	Abbr.	From	Frequency	To	Radio	Channel	PSID (Hex)
WAVE Service Advertisement	WSA	RSE	1 Hz	OBE	Ath0	178	-
MAP	MAP	RSE	1 Hz	OBE	Ath1	172	0xBFF0
Signal Phase and Timing	SPAT	RSE	10 Hz	OBE	Ath1	172	0xBFE0
Signal Status Message	SSM	RSE	1 Hz	OBE	Ath0	182	0x80
Basic Safety Message	BSM	OBE	10 Hz	RSE	Ath1	172	0x20
Signal Request Message	SRM	OBE	ASYNC	RSE	Ath0	182	0x40
Security		Both	ASYNC	Both	Ath0	174,176, 180 or 182	-

The information flow for WSM data is illustrated in Figure 31. To transmit a WSM, the *OBE/RSE_MessageTX* first registers with the WAVE Management Entity (WME) and provides the content of the WSM plus additional radio control parameters, including channel number, data rate and transmit power. To receive WSMs, the *OBE/RSE_MessageRX* registers with the WME for a list of PSIDs that it would want to receive. On receipt of a WSM, the WME compares the PSID in the WSMP header with the registered list of PSIDs and passes the matched WSM to the *OBE/RSE_MessageRX*.

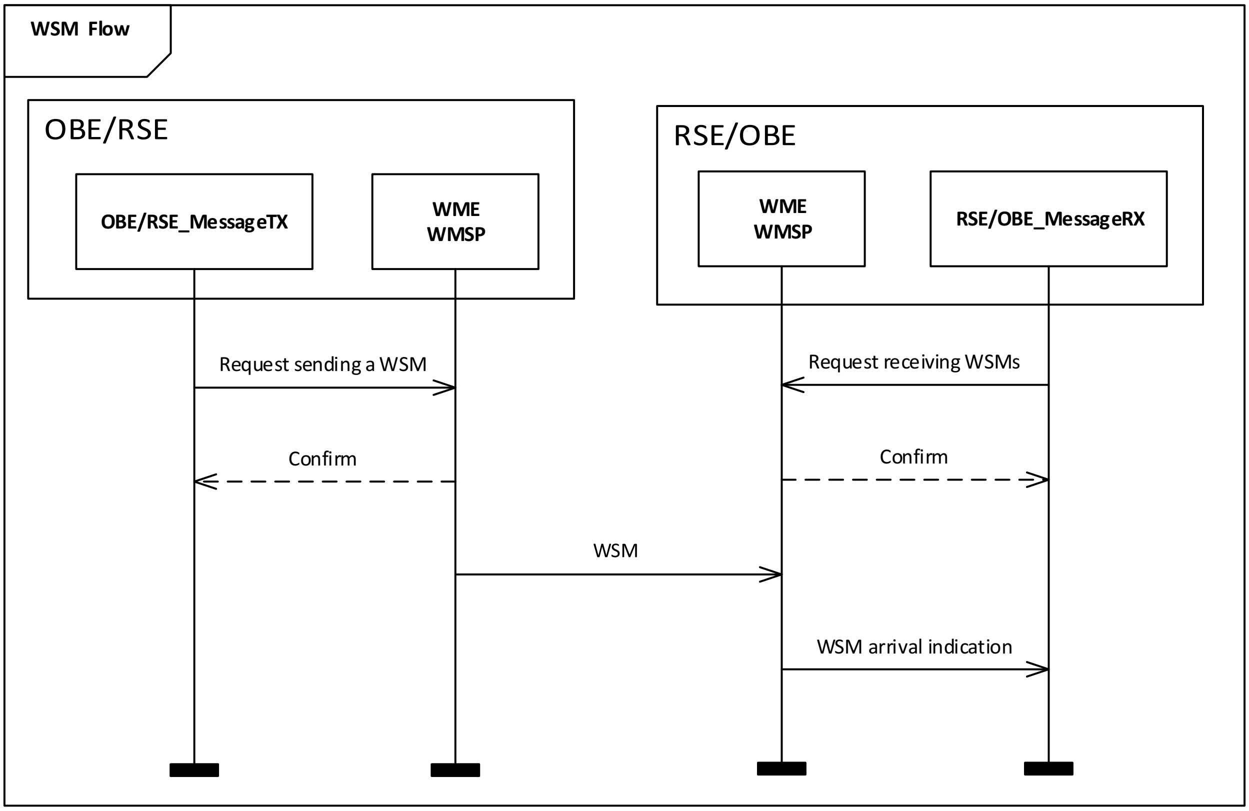


Figure 31. WSM Flow

The RSEs and OBEs have to support communication between the applications running on them and the external world. To ensure this, the WAVE messaging architecture should be capable of providing message agnostic wireless data transmission or reception services to other applications on it.

Savari StreetWave and MobiWave devices run Savari On-Board Operating System (SOBOS), which is a custom Linux distribution. A high level architecture description of SOBOS is shown in Figure 32. The SOBOS architecture consists of 4 layers. The layers, excluding the application layer, are common for all applications and based on a set of services defined in software libraries. The Kernel layer handles the hardware level device drivers for the radios and wireless devices such as DSRC, Wi-Fi and GPS. All messages on the device are sent/received via these devices. The services layer handles the messaging services such as security. Messages are processed by individual applications. These messages are handled by libraries based on the message type and the appropriate protocols. The application layer consists of applications that make use of the libraries to get the messages that they want to process further.

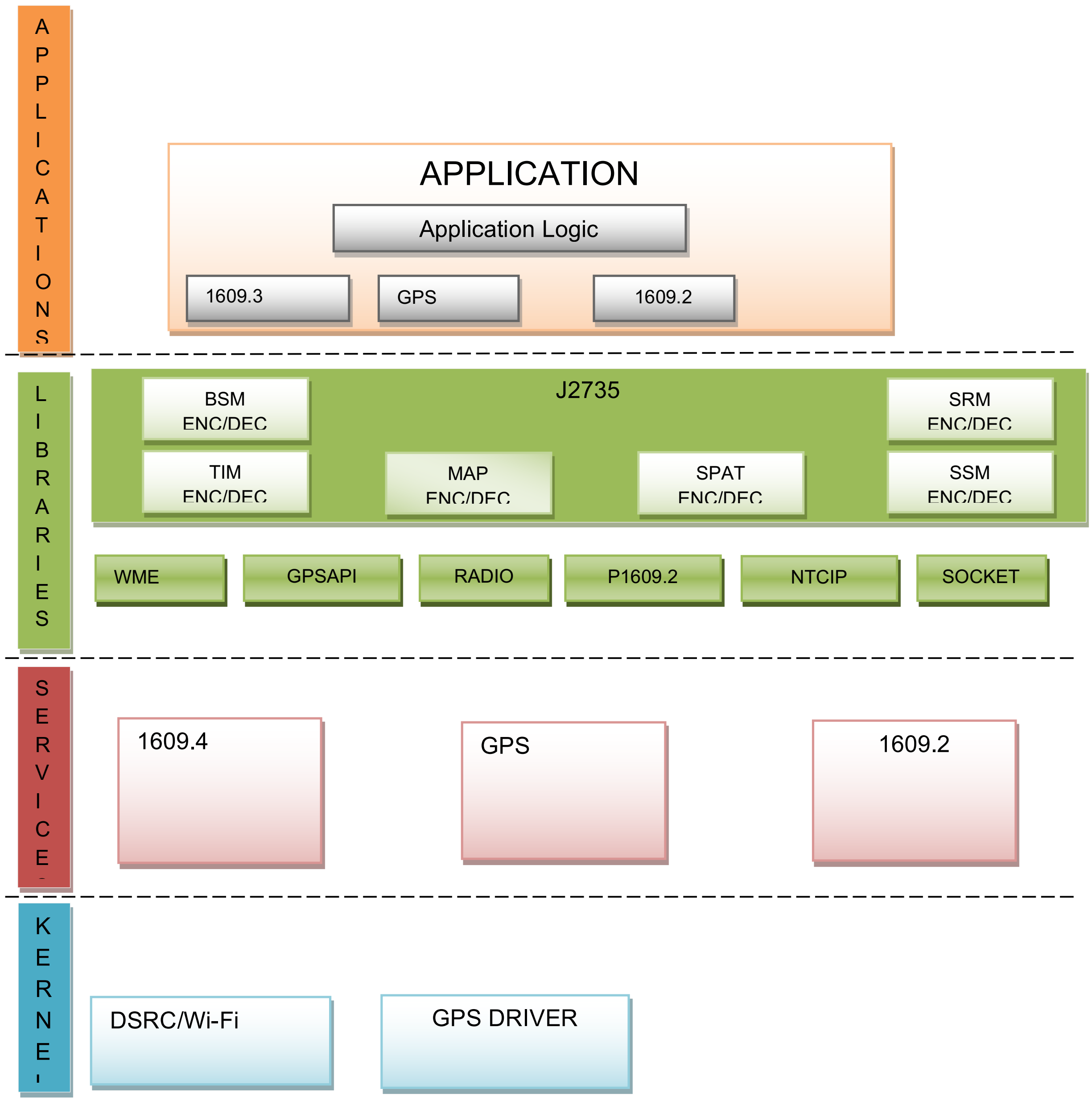


Figure 32. Savari On-Board Operation System Architecture Diagram

4.9.2 Wave Service Message

The Wave Service Advertisement (WSA) is a special class of messages that are broadcast on the DSRC control channel (Channel 178 from 5.885 GHz to 5.895 GHz) notifying users about the availability of special services, such as MMITSS, that are available at the equipped intersection, or other roadway segment. Vehicles that are eligible for priority need to be aware that a signalized intersection provides special timing services, e.g. the MMITSS application. It is envisioned that they will become aware of the MMITSS application through a WAVE Service Advertisement (WSA) that is broadcast on the DSRC Control Channel (178). The WSA will instruct eligible vehicles to switch to a specified service channel (e.g. 182) where the vehicles can send SAE J2735 specified Signal Request Messages (SRM) and receive SAE J2735 Signal Status Messages (SSM). It is assumed that the vehicles will receive SAE J2735 MAP and SPaT data on the DSRC safety of life and property channel (172) and that these vehicles will be broadcasting Basic Safety Messages on the channel 172. The MAP will be used to allow vehicles to estimate the desired time of service, identify the arrival lane (In Lane) and, optionally, the departure lane (Out Lane). The SRM contains the desired time of service, the In Lane, Out Lane (optional), and the vehicle Class (e.g. emergency vehicle, transit, truck) and Type (e.g. Fire, Police, BRT, express transit, local transit, etc.). It is assumed that fleet vehicles, e.g. transit, will have additional system logic that manages the vehicles eligibility to request priority, such as lateness and occupancy of a bus, as well as knowledge that the operating agency provides priority for their class and type of vehicle. The traffic signal control system will receive SRMs from priority eligible vehicles and will provide signal timing based on the number and class of active requests, and prevailing traffic conditions, and provide feedback through the SSM.

It is envisioned that the WSA will include a specified PSID as well as a provider service context (PSC). The PSID notifies priority eligible vehicles that MMITSS is available. The PSC can contain information about the vehicle class and types that the operating agency permits to send priority requests. In addition to the PSID and PSC from the WSA, eligible vehicle will be required to have Security Certificate SSM's that define which MMITSS services, e.g. transit priority, emergency vehicle priority, or truck priority, the vehicle is allowed to request. Management and coordination of the WSA and Security Certificate SSM's have not been implemented in the MMITSS Prototype.

The MMITSS project has requested the IEEE 1609 committee allocate a PSID for the MMITSS application. The IEEE 1609 committee and the SAE DSRC Technical Committee are still debating the allocation of PSID's for applications such as MMITSS and haven't issued an official PSID for MMITSS at the time of the preparation of this report, but have suggested that MMITSS could use the PSID for Traffic Control (PSID = 0x05). The 0x05 PSID is owned by ISO 15628 it is not clear how MMITSS would interoperate with other applications intended by the ISO committee. Also, the PSC portion of the PSID will be important in granting authorization of specific services, and the related security certificates, to the eligible vehicles per the operating agencies policies and authority there remains a need to resolve the PSID and PSC issue for MMITSS.

Since the intention of the PSID is to allow messages to be sent to the applications, there is still a need to address how multiple applications, such as safety applications and mobility applications, can receive the same data elements. For example, the basic safety message is used by both safety and mobility applications. Currently, the basic safety message has been assigned a PSID of 0x20 which is not application based. The current standards development efforts (e.g. SAE DSRC) are not addressing the multiple application issues.

4.9.3 Standard Messages

MMITSS utilizes several of the SAE J2735 messages including:

- Basic Safety Message (BSM)
- Signal Phase and Timing (SPaT)
- MAP Message (MAP)
- Signal Request Message (SRM)
- Signal Status Message (SSM)

All of these messages, except the SSM, were implemented according to the 2009 version of the SAE J2735 standard. The SSM didn't provide sufficient feedback to a requesting vehicle to allow the vehicle to know that its request had been received and considered – either accommodated or denied. The SSM was redefined in MMITSS and a new message, called the Priority Status Message (PSM), was developed and used in the prototypes. The PSM concept was presented to the SAE DSRC Technical Committee in 2014 and was used to form a revised SSM that meets the needs of MMITSS and the broader community.

The 2009 BSM message didn't' contain any information about the vehicle type or class, which was desired for the purposes of performance observation. This need was presented to the SAE DSRC Technical Committee and it was decided that vehicle class and type information would be included in Part II of the BSM and that private (passenger) vehicles would not populate this field. Other vehicles would populate the field so that they could be classified for performance, and other, purposes. This was an important need for performance measures for situations where a vehicle, say a transit bus, didn't request priority because it didn't need it, but the performance observer should be able to report the performance measures related to the transit bus.

5 Connected Vehicle Test Beds

This section provides descriptions of the Arizona Connected Vehicle Test Bed (Maricopa County) and the California Connected Vehicle Test Bed. These test beds were used to support development testing as well as the impact assessment and demonstrations.

5.1 Arizona Connected Vehicle Test Bed (Maricopa County)

The University of Arizona has partnered with the Maricopa County Department of Transportation in the development and support of the Arizona Connected Vehicle Test Bed, which is one of the seven USDOT National Affiliated Test Beds. During the MMITSS project, the test bed, shown in Figure 33. Arizona Connected Vehicle Test Bed – Maricopa County, AZ., consists of 6 signalized intersections along 1.93 miles (3,100 meters) of Daisy Mountain Drive in Anthem, AZ in the northern part of Maricopa County. The green circles shown in **Error! Reference source not found.** represent the theoretical 300 m DSRC communication range. The yellow circles represent the installations planned for 2016 and 2017. Anthem, AZ is a developed community of about 20,000 residents and includes schools (Deer Valley Unified School District), shopping, community center, golf, and a fire stations operated by the Daisy Mountain Fire Department. Traffic volumes on Daisy Mountain Drive are relatively light except in the AM and PM peak and just after school when there are busy periods of traffic. During the off-peak periods the traffic flow is very light and was an ideal environment for testing MMITSS.

Figure 33. Arizona Connected Vehicle Test Bed – Maricopa County, AZ.

In the Fall 2014, the freeway interchange at I-17 and Gavilan Peak Drive was added to the system by ADOT. In Fall 2015, the remaining 5 intersections in Anthem were added to the system so that a total of 11 signalized intersections and one freeway interchange were in the system. ADOT is planning expansion of the network to include the I-17 interchange at Anthem Way as well as the ramp meters at I-17 and Daisy Mountain and Anthem in the near future.

The 11 signalized intersections and one signalized interchanges (diamond), are each equipped with a DSRC Roadside Unit that broadcasts MAP and SPaT data as well as hosts the MMITSS system. The system is operated 24-hours/day, 7-days/week and two MCDOT REACT vehicles are equipped with vehicle on-board units operate in the network frequently during the week and when requested for testing. The MMITSS central system can be operated from the MCDOT Traffic Operations Center (and has been used to provide TOC demonstrations).

5.1.1 Field Test Procedures

Figure 34 shows the MMITSS Field Test procedures used in the Arizona Test Bed. Testing was based on each eight (8) defined software build-test cycles [originally nine (9) build-test cycles were defined, but the System Performance Observer was not included in the development since the Section Level Performacne Observer performed the same functions for the MMITSS Prototype] that included the following components:

1. Basic WAVE Message Transmission and Reception
2. OBE BSM Transmission and RSE BSM Reception; RSE MAP and SPaT Transmission and OBE MAP and SPaT Reception
3. MRP Equipped Vehicle Trajectory Aware, basic Ped App (Smartphone), OBE Priority Request Generator, and OBE GUI (webpages for visualization)
4. MRP Traffic Control Interface, Traffic Control, and Ped Signal Status Transmitter and Ped Signal Status Receiver.
5. MPR Priority Request Server, System N-Level Priority Configuration Manager
6. System Configuration Manager and User Interface, and Ped Priority Request Generator
7. MRP Performance Observer
8. Section Performance Observer

There were three stages to testing: Simulation Testing, UA Field Testing, and Arizona Connected Vehicle Test Bed (Maricopa County) Testing. Each build-test cycle is based starts using the Arizona Simulation Platform (see Section 6.1.1) where the software build is installed in the virtual RSE devices and single intersection test scenarios are run. The controller (virtual ASC/3) and MIMTSS are observed to ensure they are functioning as designed and expected. Performance data is collected for analysis. If the software build passes the single intersection test, then it is installed and tested on the six (6) intersection in the simulation platform. Again, the behavior at each intersection is observed to ensure the system performs as designed and expected. Performance data is collected and compared to the baseline case of fully actuated control. If the software build doesn't pass the intersection or network test, it is sent back to the development team for debugging.

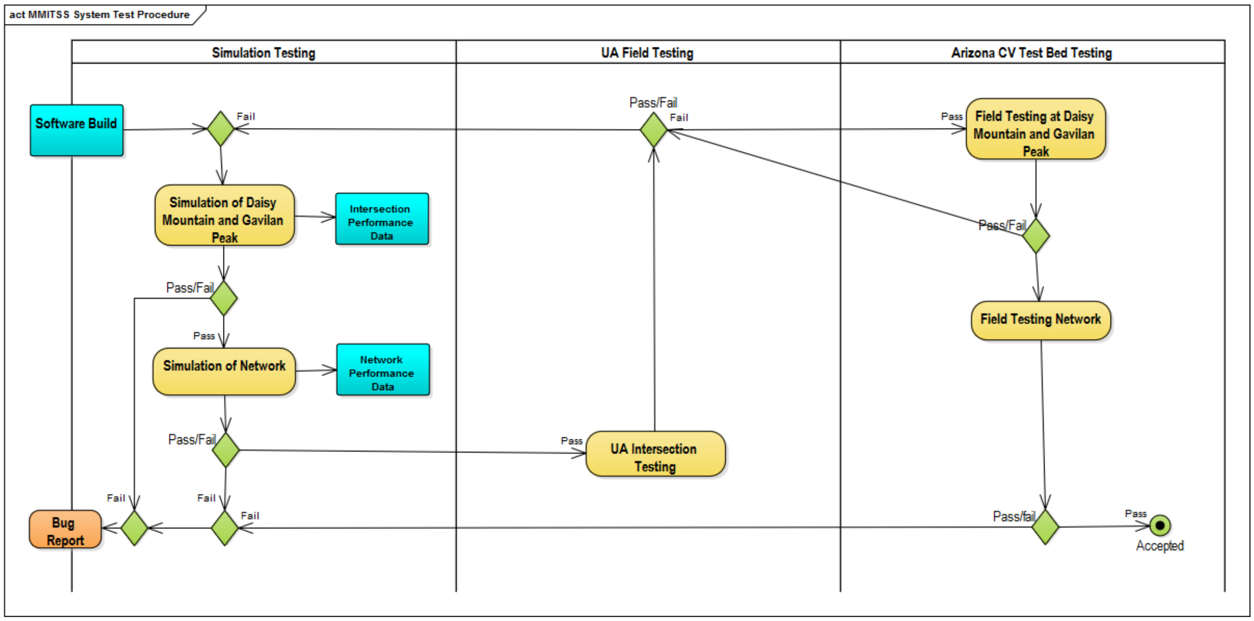


Figure 34. Arizona Field Test Procedures

Once the software build has passed simulation testing, it is tested using the intersection of Speedway Blvd and Mountain Ave in Tucson, AZ, which is a busy intersection at the north entrance to the UA campus. A RSU has been installed at the intersection, but a signal controller that is not actually controlling the intersection is used in a cabinet that has been installed by the City of Tucson for transportation research. One, or more, OBU's are installed in vehicles that are driven on Speedway and on Mountain to replicated the tests that were performed in simulation. MMITSS and the controller are observed to make sure the software build performs as designed and expected. If it passes, then the software build is installed (remotely) in the Arizona Connected Vehicle Test Bed in Anthem. If it doesn't pass, the scenario is replicated in the simulation platform to identify the problem.

Once the software build is installed in the test bed, testing is performed at the intersection of Daisy Mountain and Gavilan Peak. If the software performs as designed and expected, it is installed at the other intersections in the network. If it doesn't pass testing, the field testing is concluded and the process returns to the simulation testing where the problems are studied and replicated. If it does pass testing, the software build is installed at the other five (5) intersections and each intersection is tested. The iteration of testing ends if all intersections pass, or returns to the development team if problems are identified.

5.1.2 Field Test Results

The Field Test procedures defined above was used to ensure that MMITSS functioned as designed and expected. The Field Tests were not performance evaluations, which were conducted using simulation (see Sections 6.1.2 to 6.1.4) and by the FHWA Impact Assessment Contractor (the Impact Assessment Report is available at: http://ntl.bts.gov/lib/55000/55700/55710/MMITSS_IA_REPORT_0811_v1.4.pdf).

Table 1 summarizes the results of the Field Testing for each build-test cycle. Several components were not deployed or developed due to technical reasons or due to the change in effort required to support the two MMITSS architecture (AZ and CA).

Table 13. Field Test Results

Target Build Cycle	System Component	Field Test Result
1	RSE_SecurityCertificateService	Not Deployed
1	RSE_ServiceAdvertisementMgr	Not Deployed
1	RSE_MessageTX	Passed
1	RSE_MessageRX	Passed
2	OBE_BSMData_Transmitter	Passed
2	OBE_MAP_SPaT_Receiver	Passed
2	RSE_MAP_SPaT_Broadcast	Passed
3	MRP_EquippedVehicleTrajectoryAware	Passed
3	NomadicMMITSSApp	Passed
3	OBE_PriorityRequestGenerator	Passed
3	OBE_GUI	Passed
4	MRP_TrafficControllerInterface	Passed
4	MRP_TrafficControl	Passed
4	MRP_SignalStatus_Nomadic	Passed
4	Nomadic_SignalStatusReciever	Passed
5	System_N_LevelPriorityConfigurationManager	Passed
5	MRP_NomadicDeviceTrajectoryAware	Not Developed
5	Nomadic_PriorityDataServer	Passed
5	MRP_PriorityRequestServer	Passed
6	AuthorizedSpecalUserService	Not Developed
6	Nomadic_PriorityRequestGenerator	Passed
6	System_ConfigurationManager	Passed
6	MMITSSUserInterface	Passed
7	MRP_PerformanceObserver	Passed
7	Section_Coordinator	Not Developed
8	Section_PerformanceObserver	Passed
8	Section_PriorityRequestServer	Not Developed

5.1.3 Significant Demonstrations Conducted in the Test Bed

A significant effort was undertaken during the develop and testing cycles to provide demonstrations to key transportation stakeholders. Table 14 provides a summary of the major demonstrations with comments and observations about the features that were demonstrated.

Table 14. Summary of Field Testing using the Arizona Connected Vehicle Test Bed.

Date	Field Test Objective	Observation/Comments
October 31, 2013	Test Existing (pre-MMITSS) applications	Initial field visit with project team prior to upgrading the RSE units (to version 3.0). Testing priority control algorithms without SAE J2735 or WAVE (IEEE 1609) messaging. Preparation for the AASHTO SCOR committee visit.
November 14, 2013	Test Existing (pre-MMITSS) applications	Additional field visit with project team prior to upgrading the RSE units (to version 3.0). Testing priority control algorithms without SAE J2735 or WAVE (IEEE 1609) messaging.

		Preparation for the AASHTO SCOR committee visit.
December 3, 2013	Demonstration for AASHTO SCOR Committee	Demonstration for the AASHTO SCOR committee. The demonstration was successful and there was very good discussion about the technology, algorithms, and future research needs as well as the MMITSS project goals and approach.
March 4, 2014	Demonstration for MMITSS Pooled Fund Panel	Initial field visit for the MMITSS Pooled Fund Panel as part of the MMITSS Design Review. Field Demonstration included MAP, SPaT, SRM and SSM messages, but not using WAVE messaging.
March 20, 2014	Demonstration for APTA	APTA held their annual meeting in Tempe and took the afternoon to visit the Arizona Connected Vehicle Test Bed. The demonstration including MAP, SPaT, SRM and SSM messages (INFUSION), but WAVE was not used for the MAP and SPaT messages.
July 15, 2014	Test SRM, MAP and SPaT using WAVE Messaging. Test signal controller interface to control phase times (HOLD, FORCE OFF, OMIT).	INFUSION messaging components performed as designed. Issue with MAP message – map data to large for WAVE message limit of 1K. [MAP and SPaT messages were based on the MMITSS MRP_MAPSPaTBroadcaster (UA Implementation of the INFUSION MAP and SPaT Message)]. Successful test of signal controller interface [Signal controller interface was based on the MMMITSS MRP_SignalControlleInterface (UA implementation of the INFUSION signal controller interface)].
July 22, 2014	Testing and Demonstration of Basic Priority Control using SRM/SSM	Demonstration for Integrated Corridor Management Scanning Tour (FHWA). Successful demonstration of basic priority control including a transit vehicle and two emergency vehicles at all six intersections.
October 21, 2014	Test BSM broadcast and collection and new (smaller) MAP message	Successfully broadcast and collected BSM in the MMITSS MRP_EquippedVehicleTrajectoryAware software component (UA implementation of the INFUSION MAP Engine). New MAP message (smaller) successfully transmitted, but issues with map range (distance much less than DSRC range) and capture of roadway curvature.
November 4, 2014	Test Performance Measure estimation of Queue Length, Travel Time and Delay. Test collection of system detector data (VOS). Test Signal Control (adaptive).	Successful tests of queue length using 6 vehicles, but due to the low number of vehicle (market penetration) the full capability could not be tested. Successful capture of travel time and delay. System detector data collected for some detectors, but there were configuration issues. Adaptive signal control was tested by fixing all by two phases to be fixed time and then the six vehicles were driven back and forth for the two adaptive phases. Due to the low number of vehicles, the adaptive phases. This testing included the FHWA Impact Assessment (IA) Contractor who was on a preliminary site visit to plan the March 2-4, 2015 IA Test Plan.
November 18, 2014	Repeat tests from November 4 with bug fixes. Revised MAP tested.	Successful testing of collection of system detector data (VOS). Regression testing of priority traffic signal control, including SRM, SSM (revised version), and performance measure estimation at all six intersections.
February 2, 2015	Full system testing in preparation for March 2-4, 2015 IA Assessment.	Successful testing of all testable system features including broadcast and capture of BSM, broadcast and capture of MAP and SPaT, and use of SRM and the revised SSM for priority control.
March 2-4, 2015	FHWA Impact Assessment Testing (Contractor: Leidos)	Execution of the FHWA Impact Assessment contractors field test plan (see below). [This testing marked the development of version 1.0 of MMITSS] using some INFUSION components.
April 29, 2015	Testing of enhancement to priority signal control and regression testing of other components.	Significant enhancements to the priority control components were tested. The primary enhancements included the separation of control between the MMITSS MRP_PriorityRequestServer and the MRP_TrafficControl

		Interface. Prior to this version the traffic control commands were sent to the controller by the priority request server.
May 8, 2015	Additional regression testing of components and of a command to set the controller to the free plan so that the local controller would not operate in a coordinated fashion.	Continued testing of revisions to the traffic signal control interface. First test of the coordinated control component. At the end of this testing the team believed that there were still issues in the signal control interface. The command to put the controller into free mode caused the signal to freeze in one interval.
May 14, 2015	Additional regression testing of components	The command to set the control to the free mode was removed from the software (pending investigation of a potential problem on the controller with setting this command every 2 seconds). Regression testing of all components successful.
May 19, 2015	Demonstration for the TRB Traffic Signal System Committee Mid-year Meeting	Approximately 95 attendees participated in the demonstration of the MMITSS system which included demonstration of INFUSION components for messaging (BSM, SRM) and performance measurement. The demonstration was very successful.
December 3, 2015	Demonstration for Pool Fund Members	The Pooled Fund Members participated in a demonstration of the AZ MMIITSS system as part of the Pooled Fund Member Meeting. All capabilities were demonstrated, except for I-SIG which cannot be demonstrated due to the low market penetration. Figure 35 shows members of the Connected Vehicle Pooled Fund at the December 3, 2105 demonstration.

Figure 35. Connected Vehicle Pooled Fund Members at the December 3, 2015 demonstration in the Arizona Connected Vehicle Test Bed (Maricopa County)

5.2 California Connected Vehicle Test Bed

The California testbed and a number of test vehicles are instrumented for the MMITSS testing.

5.2.1 Field Test Corridor

The California Connected Vehicle Testbed is the nation's first DSRC test bed, which became operational in 2005 with the aim of assessing real-world implementations of Vehicle Infrastructure Integration (VII). Currently this testbed is part of the National Connected Vehicle Testbed and conforms to the latest technology standards and architecture of U.S DOT's Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) research program and is on a 4G LTE backhaul. The Testbed is supported by Caltrans and the ITS Joint Program Office (JPO) within the US DOT. It supports cutting edge research in the Connected Vehicle safety, mobility and environmental related applications, services and components. The Testbed utilizes state-of-the- art DSRC devices and provides the capability to evaluate safety, mobility, and environmental applications. The testbed spans 11 consecutive intersections along a 2-mile stretch of the highly travelled arterial of El Camino Real (SR-82), as shown in Figure 36. The location is close to many of the auto OEM research labs and the Silicon Valley.

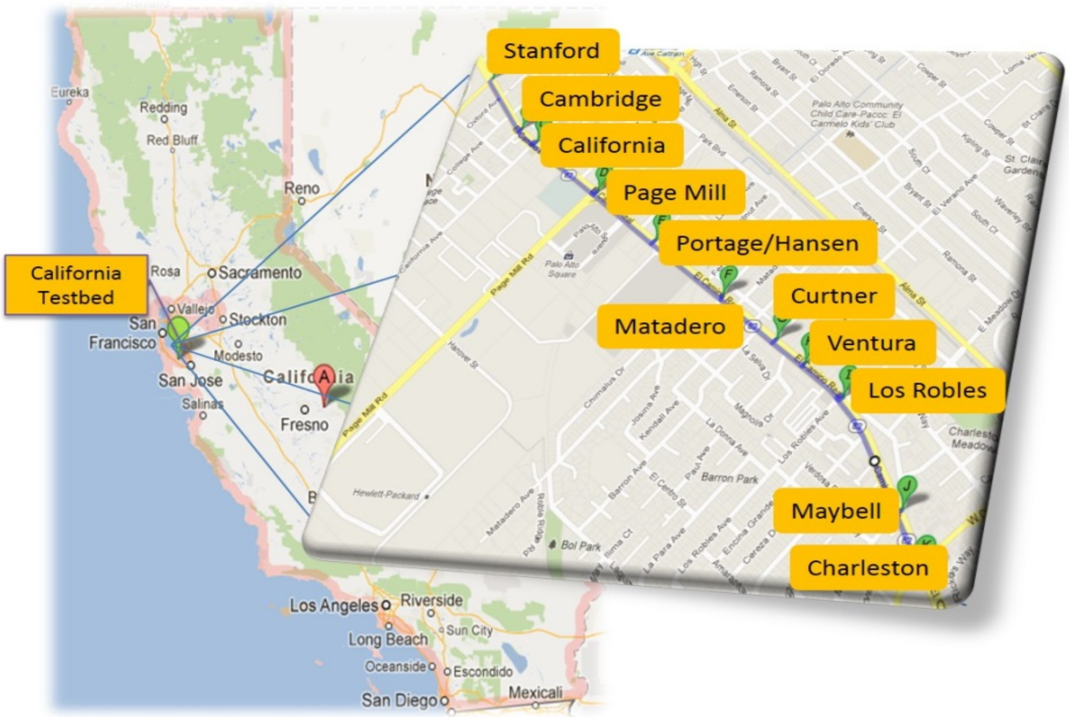


Figure 36. California Testbed at El Camino Real

Table 15 lists the 11 consecutive signalized intersections, from the northern end to the southern end, with the GPS location of the intersection (center of the intersection) and distance between consecutive intersections.

Table 15. List of intersection along California testbed (from north to south)

Intersection ID	Intersection Name	Distance to the next Intersection (m)	Latitude	Longitude
1000	Stanford Ave	320	37.42777	-122.14922
1001	Cambridge Ave	48	37.42559	-122.14678
1002	California Ave	464	37.42510	-122.14590
1003	Page Mill Rd	400	37.42305	-122.14205
1004	Portage/Hansen	384	37.42117	-122.13843
1005	Matadero Ave	368	37.41919	-122.13451
1006	Curtner Ave	125	37.41762	-122.13160
1007	Venturce Ave	211	37.41684	-122.13006
1008	Los Robles Ave	528	37.41574	-122.12815
1009	Maybell Ave	176	37.41202	-122.12464
1010	Charleston Rd	-	37.41044	-122.12331

5.2.2 Setup of Intersection Facilities

The function of intersection facilities, e.g. RSE, is to collect the signal timing and to transmit SPaT and MAP messages out with the DSRC protocol. The flow of information, shown in Figure 37, is from the signal controller, calculating at the MRP server, forwarding to the RSE, and then broadcasting to the air. The Geometric Intersection Description (GID) -- intersection map using a standard message for the applications of entry localization, tracking and navigation is created for each intersection.

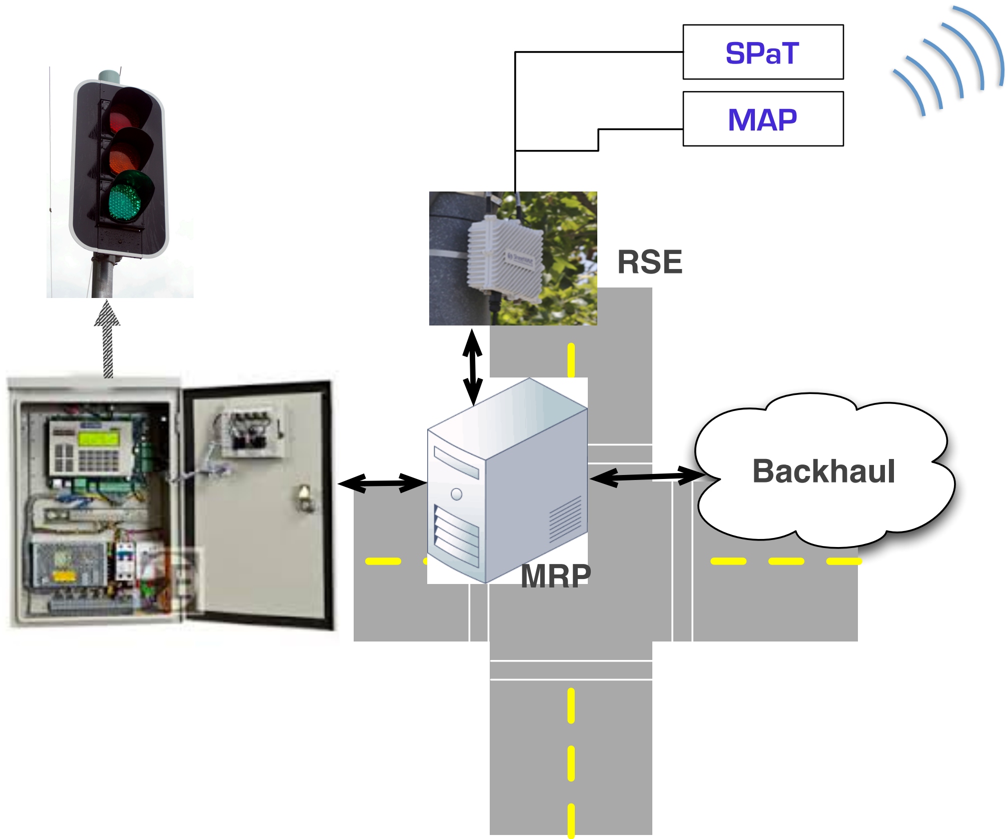


Figure 37. Intersection facilities

The consecutive intersections on the El Camino Real corridor are coordinated, which means the entire system is controlled from a master controller and are set up so lights "cascade" (progress) in sequence so platoons of vehicles can proceed through a continuous series of green lights. From the perspective of the control mode, Fully Actuated Control is usually used at an isolated and non-coordinated intersection, and Semi-Actuated Control is used for a coordinated system. Under semi-actuated (coordinated) control, non-coordinated phases (left-turn movements and side-street through movements) vary their lengths in response to vehicular and pedestrian demands, measured by embedded loops and pedestrian pushbuttons. When serviced, the length of a non-coordinated green phase varies between the minimum and imposed maximum for coordination; phase start time also varies depending on the time the call was present.

5.2.3 Overview of Intersection Conditions

A traffic study conducted by Valley Transportation Authority (VTA) in 2013 investigated a few congested corridors in the region[1]. The California CV test corridor is within the studied region.

The study shows that the average daily traffic (ADT) along the El Camino Real portion of the Project corridor ranges from approximately 36,500 to 52,600 vehicles per day, with the average vehicle trip length ranging from 5 to 6 miles. Intersection operating conditions were observed through traffic counts for the weekday morning (AM) peak period (typically between 7 AM and 9 AM) and the weekday evening (PM) peak period typically (between 4 PM to 6 PM), including intersection turning movement counts. The data were collected in May, October, and November 2013. The intersection conditions are described using level of service (LOS)[2], with LOS A representing little or no delay, to LOS F representing long delays. Based on the VTA Congestion Management Plan and general plans prepared by study area cities, the acceptable LOS are generally defined as LOS E or better for CMP (Congestion Management Program) intersections, and LOS D or better for local signalized intersections. Table 16 summarizes the conditions of the intersections within the test corridor.

Table 16. California Testbed Intersection Conditions

Intersection Location by Cross Street Name	AM Peak Hour		PM Peak Hour	
	Delay	LOS	Delay	LOS
Stanford Ave	31.2	C	31.3	C
Cambridge Ave	7.0	A	13.5	B
California Ave	15.7	B	24.8	C
Page Mill Rd/Oregon Expy	73.4	E	58.5	E
Portage Ave	30.2	C	43.2	D
Hansen Way	80.2	F	45.7	D
Matadero Ave/Margarita Ave	19.6	B	15.1	B
Curtner Ave	4.1	A	3.2	A
Ventura Ave	13	B	11.3	B
Los Robles Ave	17.9	B	14.3	B
Maybell Ave/El Camino Way	19.8	B	10.6	B
Charleston Rd/Arastadero Rd	37.5	D	38.3	D

5.2.4 Overview of Test Vehicles

Five conventional automobiles are used as test vehicles. Each of the test vehicles is temporarily instrumented with an OBE to determine the geographic coordinate of the test vehicle, then can be used to calculate its relative location to the approaching intersection. Five OBEs, including 2 Savari MobiWave units and 3 Arada LocoMate mini 2 units, were used for the field testing. Each OBE is programed to be a transit vehicle, a freight, or a general car.

[1] VTA, El Camino Real Bus Rapid Transit Project Draft Environmental Impact Report/Environmental Assessment, 2014

[2] Defined in 2000 Highway Capacity Manual by Transportation Research Board

5.2.5 Field Test Procedures

A field testing plan was developed to provide the experimental protocol for MMITSS field testing and data collection at the California Connected Vehicle Testbed. A probe vehicle technique was used for the field data collection. Test vehicles were driven over the study section in a series of runs based on the data collection plan. The drivers, including PATH researchers and Caltrans employees, were instructed to travel at a speed that is judged to be representative of the speed of general traffic at the time.

A staging area is located at parking lot of Stanford Palo Alto Playing Field at the intersection of Page Mill Rd. crossing El Camino Real. Test trips leave from the staging area southbound along El Camino Real passing Charleston and make a U-turn at Deodar St. intersection or further. The northbound trip will travel along El Camino passing Stanford Ave and make a U-turn at Leland Ave. The travel time ranged between 15 min to 25 min for a round trip, depending on the traffic.

To collect the 'before' data, 5 'silent' vehicles (without generating requests for priority) were used to circulate within corridor to collect the field data of baseline scenario. Another set of tests were conducted with same vehicle settings but with priority request function tuned on to collect 'after' data for evaluation of the benefits of MMITSS signal priority application at a single intersection and at a corridor.

Test of PED-SIG were also conducted to demonstrate Savari SmartCross app can generate pedestrian crossing request, and once executed, provides added convenience for pedestrian and potentially reduce pedestrian waiting time at a single intersection.

The test data collected from the field tests include:

- CV vehicle trajectories saved at MRP at intersections,
- Traffic data (including pedestrian) saved at MRP at intersections,
- OBE data saved on at least one test vehicle together with fuel consumption data, and
- Performance measures, including trip travel time, number of stops at intersections, cumulative stop time, and intersection delays, and priority execution frequency.

5.2.6 Test Results

This section provides a general overview of the testing result, including general information on the bus operation and the ATSP system.

5.2.6.1 Sample Size

Test vehicles emulate buses and trucks to continuously travel along the corridor southbound and then northbound for 'before' and 'after' data collection. 'Before' data were collected on Thursday November 5th, 2015, 10 AM – 6 PM, with all instrumented vehicle transmitting BSM and receiving SPAT and MAP messages but not emitting priority requests. 'After' data were collected on Thursday November 12th, 2015, 10 AM – 6 PM, with 2 'buses', 2 'trucks' and 1 car. One of the bus also has Eco-driving advisory system in active. Pedestrian call with Savari SmartCross App was tested for functionality. Table 17 summarizes the sample size of the field testing.

Table 17. Sample Size of Test Trips

Type	Device	Units	Number of “Before” Trips (without priority)		Number of “After” Trips (with priority)	
			Southbound	Northbound	Southbound	Northbound
Bus	Savari	2	14	16	18	18
Truck	Arada	2	19	20	17	18
Subtotal (Bus + Truck)			33	36	35	36
Car	Arada	1	7	9	3	4

5.2.6.2 Test Data

Figure 38 and Figure 39 depict two single vehicle trips with priority function activated. In both cases, both early green (EG) treatments and green extension (GE) treatments are granted. In the southbound trip shown Figure 38, the test vehicle first stopped at Cambridge Ave and then California Ave. An EG was submitted and was granted at California Ave. The test vehicle made a brief stop and passed the intersection. When approach Portage Ave, the signal was about to change to red. A GE request was submitted and granted. The vehicle successfully passed the intersection without a stop. The vehicle encountered a red signal when approaching the Maybell Ave and requested EG treatment. EG was successfully granted before the test vehicle had to stop. In another northbound trip shown in Figure 39, a granted GE treatment helped the vehicle to pass through the intersection without stopping. These two example trips clearly demonstrate how priority reduce number of stops and save times for test vehicle.

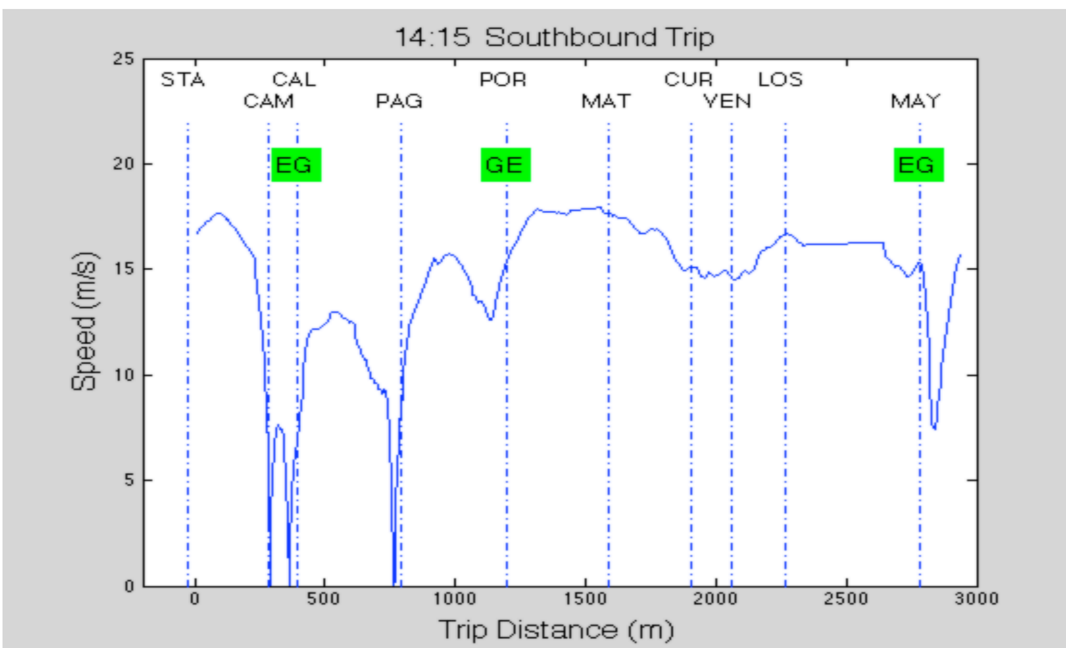


Figure 38. Example of Granted Priority Treatments (Southbound Trip)

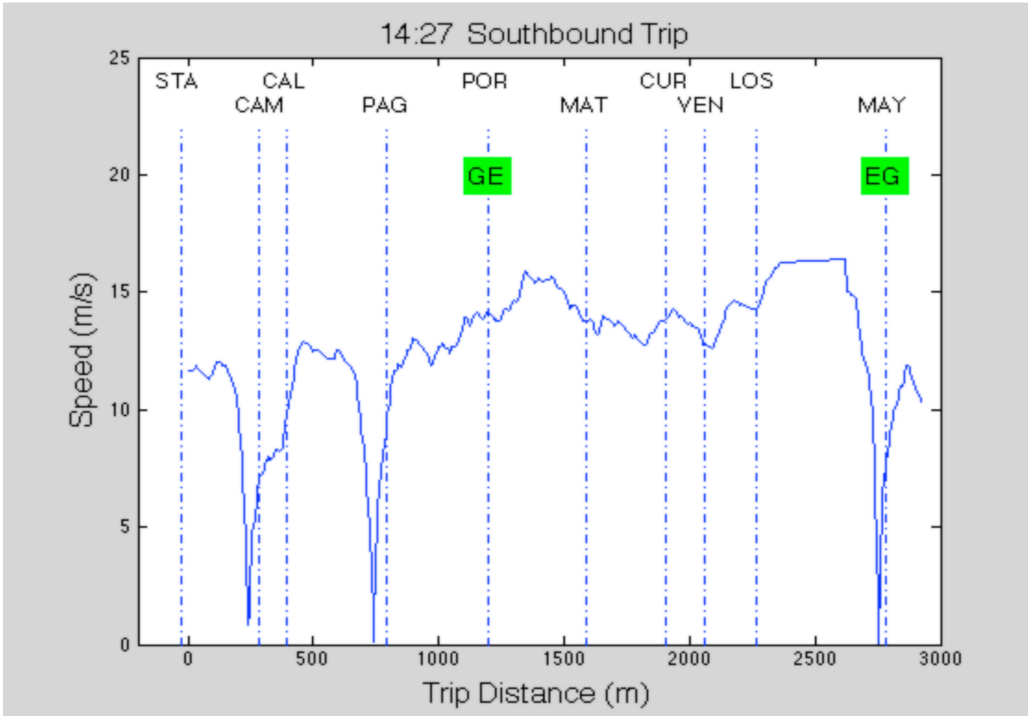


Figure 39. Example of Granted Priority Treatments (Northbound Trip)

Figure 40 summarizes the percentage of executed priority treatments (at each of the intersections along the test corridor). The frequency of executed early green (EG) treatments are shown in color blue while the executed green extension (GE) treatments are shown in color yellow. Page Mill and Ventura intersections have much lower priority treatments than the other intersections. Page Mill is operated at running free mode most of times. Ventura is close to the adjacent intersections therefore the opportunity for vehicles to stop at this intersection is low. In all cases, there are significant more executed EG treatments than executed GE cases. However, the numbers of executed GE treatments are interestingly quite different. Most EG calls occurred at four consecutive intersections California, Page Mill, Portage/Hansen and Matadero. Because Page Mill is operated in running free mode, the adjacent intersections cannot totally coordinated, causing vehicles running at lower average speeds therefore having higher chance to arrive at intersections at the end of green.

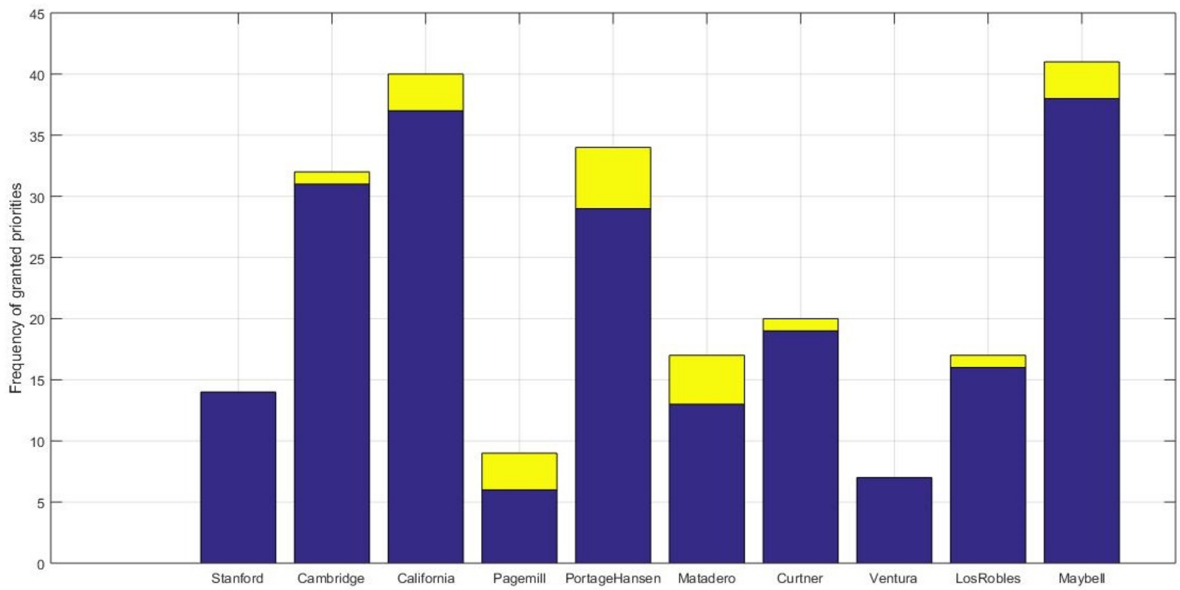


Figure 40. Priority Frequency by Intersection

Figure 41 illustrates the distribution of granted priority treatment at different time of day. Vehicles have obtained similar percentage of early green treatments across the day, while substantial higher percentage of green extension priority treatments in the afternoon peak period. This is due to a combinations of signal coordination, pedestrian demand, and most importantly the higher traffic congestion level for the southbound in the afternoon peak. As a result, the opportunity for vehicles encountering the end of green is greater. Cumulatively, the percentage of priority treatments in the morning is moderately higher than non-peak hours while the percentage of the priority treatments in the afternoon peak is much higher.

In order to evaluate the benefits of MMITSS based priority treatments, the following Measure of Effectiveness are used to analyze the effectiveness of priority treatments for test vehicles.

1) Number of Stops at red: The number of stops at intersections is a direct measurement of the effectiveness of signal priority. The reduction in number of stops at red is not only decreases vehicle waiting time at intersections and also leads to reduced fuel emission and lowered vehicle wear and tear due to braking.

2) Cumulative time of delay: The delay caused by stops for any reason within the test corridor, including stops at red signal or to react to the traffic in front.

3) Trip travel time: The total travel time for each trip from the beginning to the end of the test corridor. The reduction of the number of stops and intersection delays will reduce the total trip travel time within the concerned segment of road.

4) Cumulative intersection delay: Delays are caused by stops at red. The change in delay, more specifically the waiting time, at individual intersections is a direct measure of benefits on designed vehicle types.

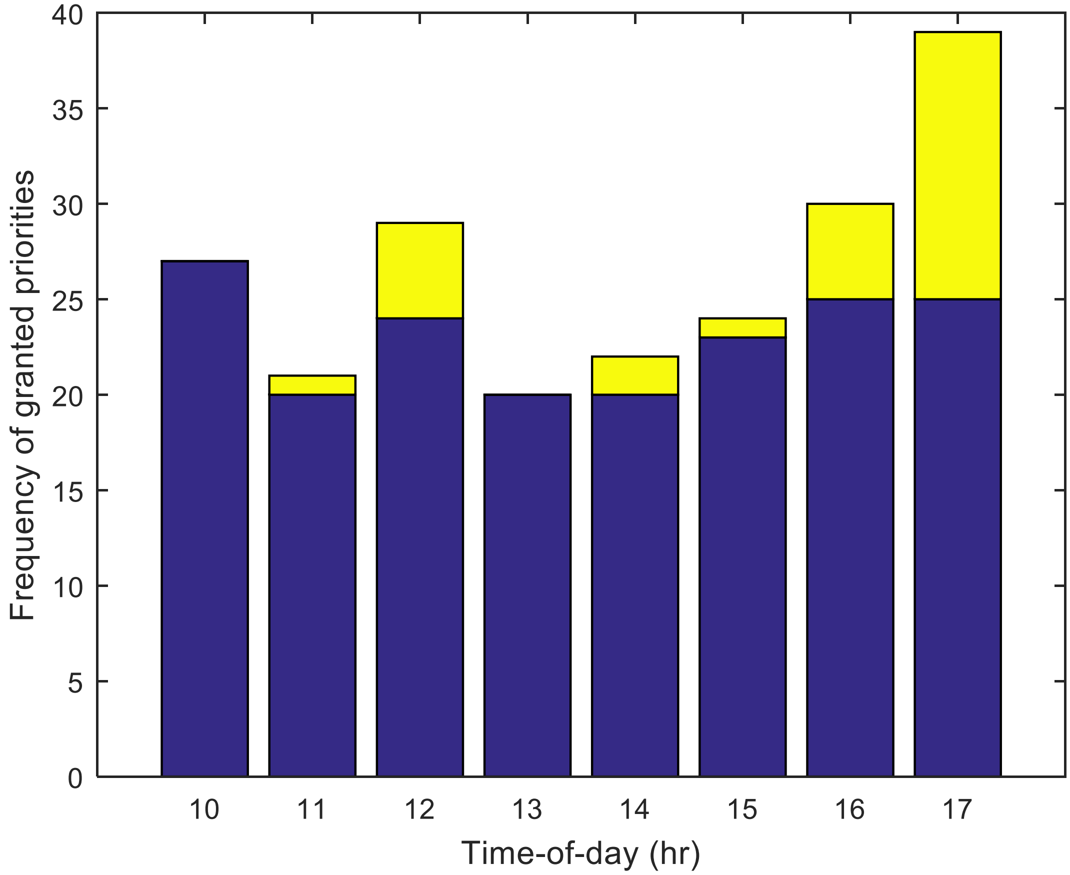


Figure 41. Priority Frequency by Intersection

Table 18 is a statistical summary of the benefits of MMITSS priority treatments with data collected on for all trips made during the MMITSS field testing. “Without Priority” refers to the test trips made on November 5th, 2015, with all 5 test vehicles not emitting signal priority requests. “With Priority” refers to the test trips made on November 12th, 2015, with 2 “buses” and 2 “trucks” emitting signal priority requests and 1 ‘car’ not emitting signal request messages. The total trips made on these two test days are slightly different due to the rotation of test drivers. The data is categorized using the MOEs presented above. The data shows that, based on limited tests conducted along the 2-mile segment of El Camino Real testbed, MMITSS priority treatment can help to reduce 7.3% stops at red and 7.4% time in trip travel time.

Table 18. Summary of the benefits of priority treatments

	Without Priority			With Priority			Improved Percentage
	SB	NB	Total	SB	NB	Total	
Number of Trips	33	36	69	35	36	71	
Number of Stops	3.9 ± 1.5			3.6 ± 1.4			7.3%
Cumulative Time of Stop	127.0s ± 65.3s			109.1 s ± 53.2s			14.1%
Trip Travel Time	386.4s ± 82.1s			358.1 s ± 67.6s			7.4%
Delay due to Stops	132.1s ± 74.2s			115.7 s ± 54.5s			12.5%

During the field testing, one testing vehicle was served as a general car for both the ‘before’ and ‘after’ data collection. The comparison of the MOEs for the car trip is summarized in Table 19. Summary of Car Trips

Table 19. Summary of Car Trips

	Trips on 11/5/15			Trips on 11/12/15		
	SB	NB	Total	SB	NB	Total
Number of Trips	7	9	16	3	4	7
Number of Stops	4.4 ± 2.1			3.4 ± 1.0		
Cumulative Time of Stop	169.4s ± 87.8s			162.3s ± 107.2s		
Trip Travel Time	452.1s ± 112.5s			415.3s ± 117.7s		
Delay due to Stops	163.8s ± 60.1s			143.5s ± 47.6s		

Figure 42 to Figure 44 show the cumulative time of stops, trip travel time and delay due to stops in relationship with number of stops. Each blue dot represents one ‘before’ trip and each red dot represents one ‘after trip. It is clear that the red dots have all moved toward the lower left indicating that the ‘after’ have less number of stops while the cumulative time of stops, trip travel time and delay due to stops are all reduced. These figures graphically display the benefits of priority treatments.

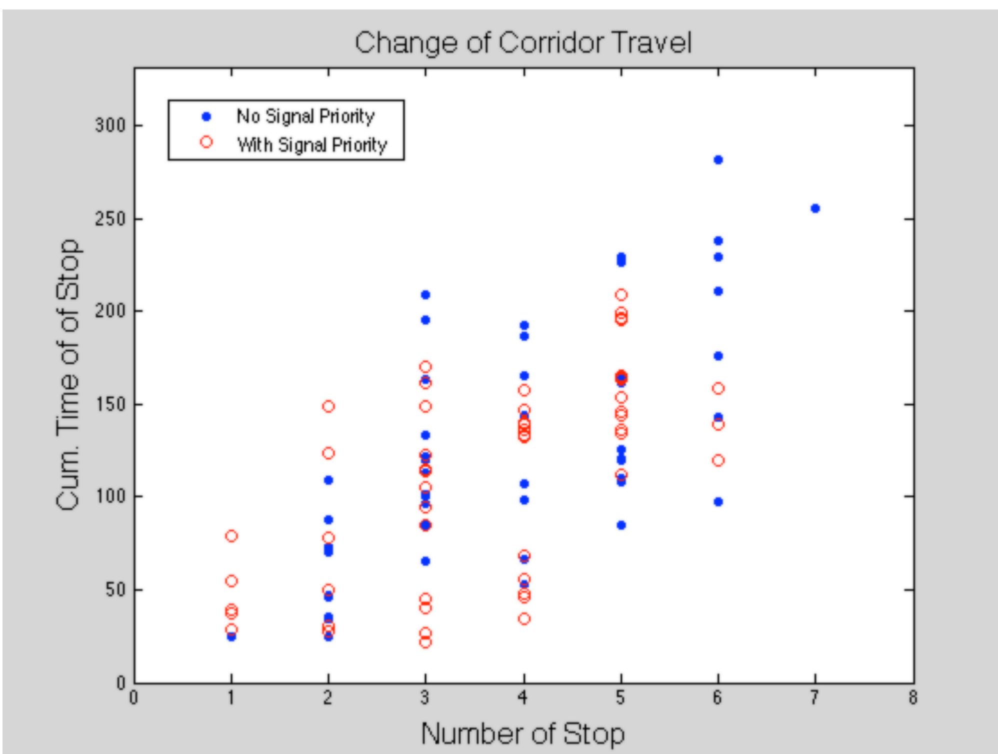


Figure 42. “Number of Stop” and “Cumulative Time-of-Stop”

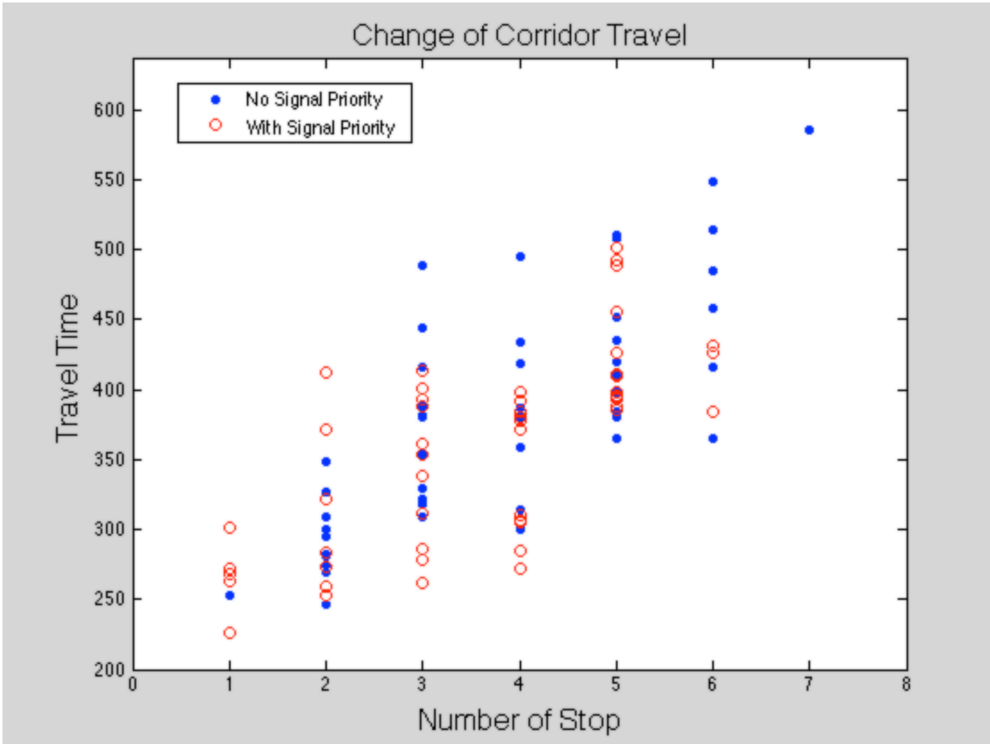


Figure 43. “Number of Stop” and “Total Travel Time”

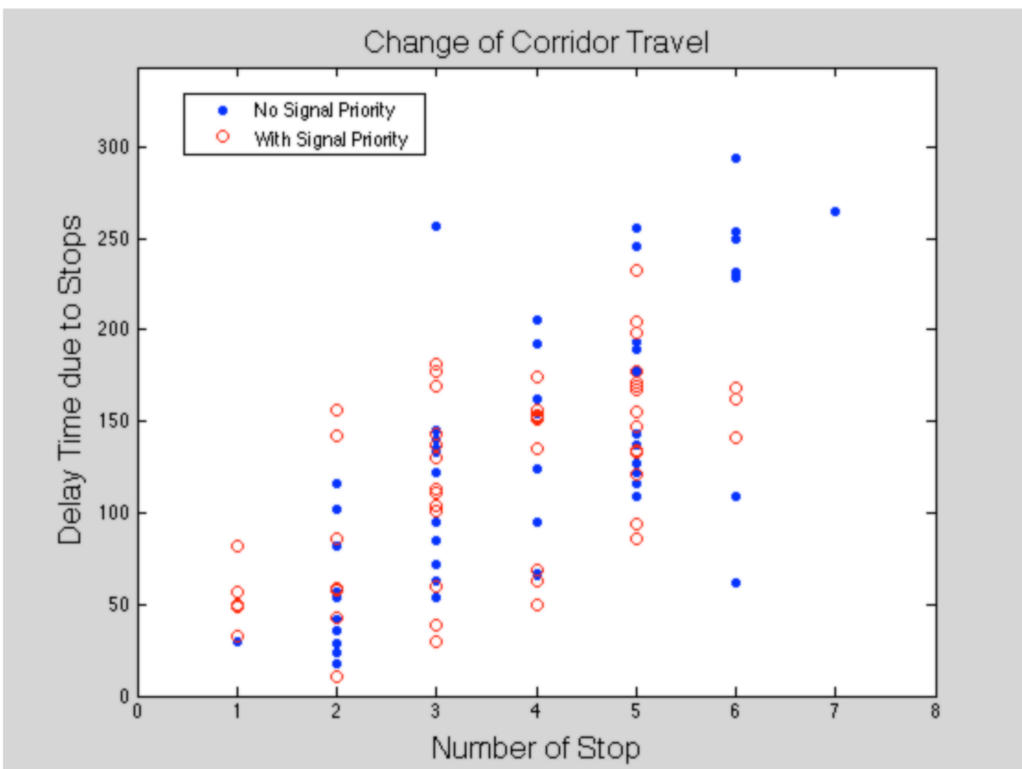


Figure 44. “Number of Stop” and “Delay Time due to Stops”

5.2.6.3 Analysis of Potential Time Saving for Trucks and Transit Buses

While our testing has shown significant time savings for testing vehicles, these results can apply to trucks better than to buses because the test trips did not include bus dwelling and other features of transit operation. Below is a trip-based analysis to evaluate the benefits of priority treatment on transit operation.

Transit vehicles spend an average of 15 percent of their trip time waiting at traffic signals (Baker et al. 2004). TSP aims to reduce bus intersection delay and thereby leading to shortened trip time and potential reduced number of stops at red and faster average running speed.

Bus trip travel time (TT) consists of running time (RT), dwell time (DT) at bus-stops and waiting time (WT) at intersections, i.e.,

$$\begin{aligned} \mathrm{TT} &= \mathrm{RT} + \mathrm{DT} + \mathrm{WT} \\ &= \mathrm{TT} \cdot (P_{\mathrm{RT}} + P_{\mathrm{DT}} + P_{\mathrm{WT}}) \end{aligned}$$

where P_{RT}, P_{DT} and P_{WT} are the percentages of running time, dwell time and waiting time in the total travel time.

With TSP off, the average total trip travel time on study section was 386.4 sec (with error of 82.1 sec). In a previous study involving field operational test of transit signal priority, statistical results were collected on operation characteristics for Samtrans 390 and 291 bus lines. The data show that dwell time accounts for about 15-18% of the trip travel time and intersection waiting time accounted for 13-16 % of trip travel time for buses operated along El Camino Real[3]. The reduction in waiting time was 12.5% from the collected under this study. Applying the above equation, the potential reduction in travel time along the study section is 6.2%.

5.2.7 Field Test Results

Field operational tests have demonstrated the effectiveness of MMITSS priority treatment, as summarized below:

- Number of stops for red was reduced on an average by 7.3%
- Bus waiting time at intersections was reduced by 12.5%
- Average vehicle trip time was increased by 7.4%

This evaluation study also demonstrated that MMITSS priority treatment might provide higher percentage of reduction in trip time for trucks than for buses due to the nature of transit operation for frequent stops. However, transit signal priority can achieve significant benefits for headway-based bus service as well as when bus stops and schedules are designed to take advantage of TSP.

6 Simulation Experimentation and Analysis

Although real-world tests of small fleets of equipped vehicles can demonstrate some of the potential benefits of connected vehicle applications, they are not able to show the potential outcomes of these applications in large-scale deployments. Due to the complex nature of traffic control systems it is desirable to utilize traffic simulation in order to test and evaluate the effectiveness and safety of new technologies and systems. The complexity of cooperative intelligent transportation systems presents

[3] PATH, Field Operational Test of Adaptive Transit Signal Priority System, PATH report for TO6400. Oct, 2008

significant challenges for modeling and simulation before efficient and reliable applications can be deployed in the real world. A methodology is required to test and evaluate applications during the design, development, and testing phases of a project.

Current advances in computational technology make it possible to simulate a large-scale traffic network in a short time. Among the different levels of traffic simulation models, microscopic traffic simulation is the most suitable for connected vehicle applications because individual vehicles are modeled in detail. It has become one of the traffic researchers' most useful tools for analyzing various traffic related concepts such as driver behavior modeling and traffic signal timing plans. A microscopic traffic simulation tool represents the most adequate and viable alternative for evaluating the benefits of connected vehicle applications compared to field tests. In addition, the cost of a simulation is much less than the cost of field tests.

Simulation of any real-time system not only depends on the numerical computation, but also depends on the timeliness with which the simulation model interacts with external control equipment (Bullock, Johnson, Wells, Kyte, & Li, 2004). Microscopic hardware-in-the-loop (HILS) and software-in-the-loop (SILS) simulations are techniques that have been used in traffic signal research to test new concepts. There are several reports of successful use of HILS. Bullock, et al. reviews the application of hardware-in-the-loop simulation and recommends procedures that permit rigorous and systematic evaluation of alternative control algorithms using a common microscopic traffic model. Byrne et al. (Byrne, Koonce, Bertini, Pangilinan, & Lasky, 2005)), proposed a HILS to simulate one intersection in Portland to examine the effects of TSP signal control strategies on transit performance. Compared to the traditional HILS and SILS, that primarily includes traffic signal controllers, simulation of connected vehicle environments require roadside equipment (RSEs) and onboard equipment (OBEs) as well (Ding, He, Head, Saleem, & Wu, 2013).

There have been several studies that developed simulation tools for vehicular mobility and wireless communications. Wu (Wu et al., 2005) proposed a separated wireless and traffic simulation tool based on CORSIM and QualNet. They investigated the needs to test the advanced communication networks along the I-75 freeway in the Atlanta, Georgia during peak and off peak traffic periods. Piorkowski (Piórkowski et al., 2008) proposed TraNS which is a combination of two independent open source traffic and wireless simulators, namely SUMO and ns-2. SUMO (Simulation of Urban Mobility) is an open source, highly portable, microscopic and continuous road traffic simulation package designed to model large road networks ("DLR - Institute of Transportation Systems - SUMO – Simulation of Urban MObility,"). Vehicular Network Integrated Simulation (OVNIS)(Pigne, Danoy, & Bouvry, 2010)) is an online platform that presents a combination of SUMO and ns-3. In this platform, ns-3 is extended to be a traffic aware network manager that can simulate wireless transmission between vehicles based on their positions (retrieved by SUMO) and can control the whole simulation process to have relative interactions between the connected blocks. Rondinone et al. presented an open source platform called "iTETRIS" that can study and optimize cooperative connected vehicle applications in large scale scenarios (Rondinone et al., 2013).

The platform developed to support testing MMITSS functionality provides opportunities for demonstrating intelligent traffic control methodologies and algorithms with the involvement of hardware Onboard Equipment (OBE) and Roadside Equipment (RSE) devices. In this platform, Savari StreetWave RSE units and OBE MobilWave devices are used in both field tests as well as in HILS. The simulation platform is created based on the VISSIM simulation tool. The simulation tool developed uses the external driver model .dll interface (DriverModel.dll) of VISSIM to get the information from connected vehicles, by encoding critical vehicle information into the SAE J2735 Basic Safety Messages (BSM) and/or Signal Request Messages (SRM), and transmitting these messages to the associated OBE through socket.

6.1 Description of the Connected Vehicle Simulation Platform

6.1.1 The Arizona Simulation Platform

Figure 45 illustrates a scaled view of the traffic signal control components implemented in the connected vehicle simulation platform. The simulation platform is capable of including all of the equipped travelers (except pedestrians and bicycles) from different modes as connected vehicles that can communicate with RSU radios by using an OBU. The OBU is a hardware device deployed on the vehicle and contains a suite of technology and applications and utilizes wireless communication standards to establish a connection to RSU. The RSU Radio is the hardware device that is responsible for managing all of the 5.9GHz DSRC communications between the vehicles and the infrastructure. The RSU receives messages such as the Basic Safety Message (BSM) and Signal Request Message (SRM) from the OBU and forwards them to the Traffic Control Processor where all intersection level traffic control applications reside. The RSU radio broadcasts the intersection geometry (MAP) and signal phase and timing (SPaT) messages to nearby connected vehicles. The Traffic Control Processor also utilizes data from the traditional field sensor/detector system and combines them with connect vehicles data to make traffic control decisions. The Traffic Management System hosts section level or system level software components such as user interface, section level priority server, and signal coordination.

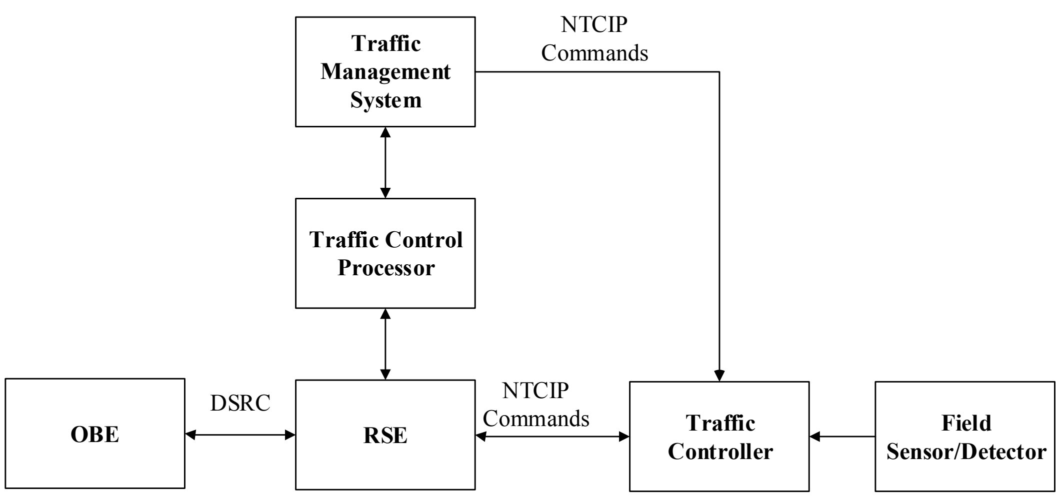
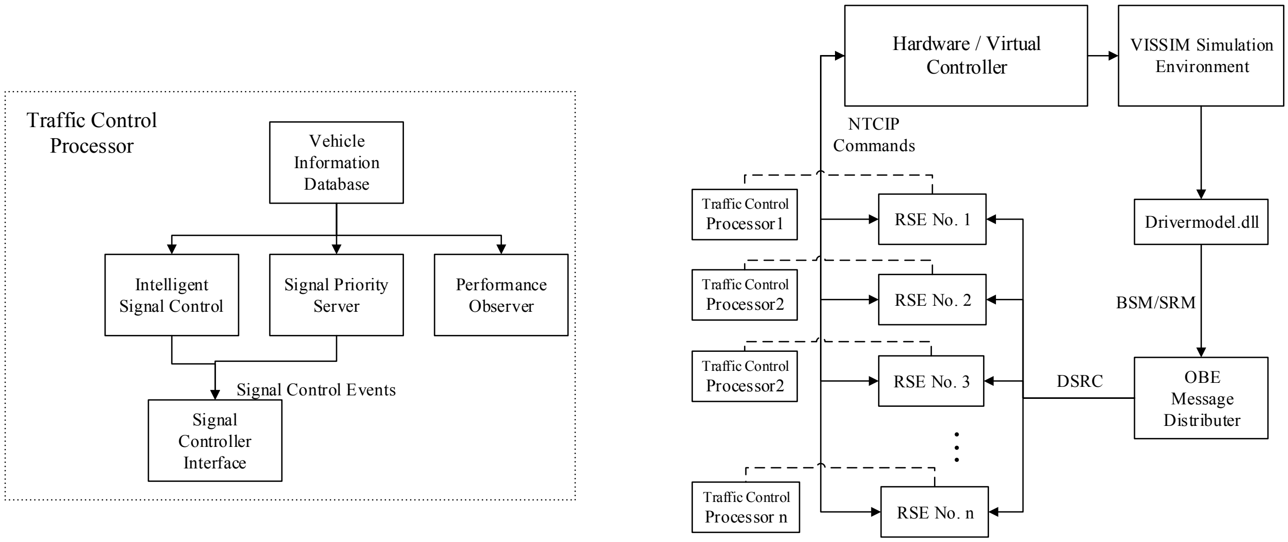


Figure 45. The Traffic Signal Control System Under Connected Vehicle Simulation Platform

The simulation platform uses a combination of simulation software along with one OBE and several RSEs to replicate the real-world system mentioned above. Figure 46 illustrates the architecture of the simulation platform which includes multiple OBE and RSEs. Figure 46(b) shows the HILS structure and Figure 46 (a) shows the components reside inside of the Traffic Control Processor.



(a) Traffic Control Processor components

(b) HILS structure

Figure 46. The Architecture of the Proposed Simulation Platform

The platform uses the dynamic-link library DriverModel.dll (DLL) API in VISSIM to get the information from a connected vehicle. This API provides the option to replace the internal driving behavior by a fully user-defined behavior for some or all vehicles in a simulation model. Each vehicle type in VISSIM can have a different DLL. During a simulation run, VISSIM calls the DLL for each specified vehicle type during each simulation time step to determine the behavior of the vehicle. VISSIM passes the current state of the vehicle and its surroundings to the DLL and the DLL computes the vehicle information including position, speed acceleration, steering angle, and other vehicle attributes (14).

A generic open source ASN1 encoder/ decoder tool (15) is used to process the ASN1 specified SAE J2735 message and generates an encoded message based on a data structure in the C/C++ programming language. The tool encodes the BSM and sends it to the OBE Message Distributor through a UDP socket with a specific IP address and port. If the connected vehicle is a priority eligible vehicle, it also encodes an SRM and sends it as well.

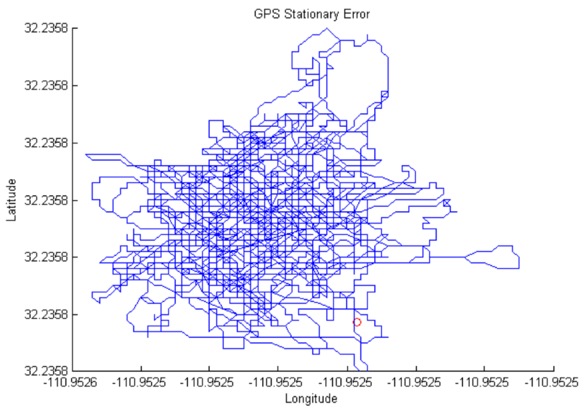
The OBE Message Distributor is responsible for receiving messages from all vehicles that have been defined to be connected vehicles and distributing them to the correct RSE according to the location of the connected vehicle in the network. The OBE Message Distributor is one OBE device that serves as the OBE for each simulated vehicle. The communication between OBE and RSEs is through DSRC. The BSM is sent at 5.86 GHz frequency (e.g. DSRC Channel 172) and the SRM is sent at 5.91 GHz frequency (DSRC Channel 182 – which has been selected as the Service Channel for MMITSS priority control). Each RSE also communicates to one virtual controller in VISSIM through NTCIP commands through the controllers Ethernet interface. The Econolite ASC/3 controller firmware is used on both the hardware and in the software (virtual) controller.

Each RSE is connected to a Traffic Control Processor. There are five major components in the TC Processor: vehicle information database, intelligent signal control, signal priority server, performance observer, and signal control interface. The vehicle information database receives the BSM/SRM from the OBE message distributer and forwards vehicle information to different components as required. The intelligent signal control component is responsible for generating optimal signal plans for non-priority vehicle while the signal priority server is responsible for serving priority eligible vehicles. The performance observer monitors the intersection performance in real-time using the BSM data. The signal control interface is responsible for managing communications between the traffic control components and the traffic signal controller using NTCIP(16) commands. A summary description of each of the components is provided later in this paper.

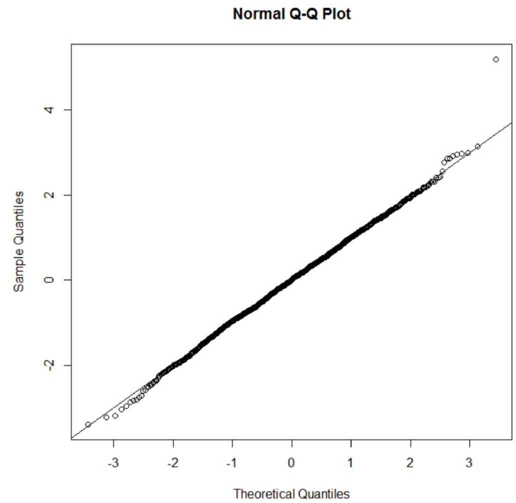
6.1.1.1 Calibration of The Model

Calibration of simulation models is a critical part of the simulation process. In addition to the recommended processes of calibrating model parameters such as roadway geometry, traffic volumes, transit routes and headways, traffic composition (e.g. percentage of trucks or other vehicle classes), turning ratios (or OD matrices), startup lost time, saturation flow rates, and traffic signal timing parameters(17), simulation of a connected vehicle environment requires calibration of DSRC range, packet loss probabilities, market penetration (by class and type), and GPS location and errors. The DSRC range can be measured for each intersection approach in the field network if DSRC radios are installed. Packet loss can be estimated from BSM data collected and logged assuming that equipped vehicles produce BSM every 0.1 seconds. The packet loss function can depend on the distance from the DSRC radio. Market penetration is essentially zero at this time, unless a fleet of equipped vehicles is used as part of a field test. If there is a fleet available, the traffic volume (measured from system detectors) and the number of OBEs can be used to estimate market penetration.

In order to replicate the real-world situation in the simulation environment, the GPS errors must also be modeled. In the field test, the vehicle location data collected from GPS is subject to errors. The errors come from a variety of sources including signal arrival, ionospheric effects, ephemeris errors, and multipath distortion etc. which will affect the accuracy of locating the vehicle in the map algorithm. However, these errors don't exist in the simulation environment. Two hours of 1Hz GPS data were collected from the GPS unit located in an RSE device as shown Figure 47(a). The Device is stationary at the intersection of Mountain and Speedway in Tucson, AZ. The blue lines show the trajectory of the GPS points and the red dot shows the origin. It can be seen from Figure 47(a) that the GPS errors are not independent over time so a time-series ARIMA model is used to model the error. The latitude and longitude are treated independently so two univariate ARIMA models are applied separately to the latitude and longitude respectively. Normality test and the correlation function of the residuals show a good fit of the data. Figure 47(b) shows the Q-Q plot of the standardized residual of the latitude data with p-value 0.618 after fitting to a ARIMA(2,0,2) model. The fitted ARIMA model is then used to create errors in the simulation platform.



(a) Stationary GPS Error



(b Residual Normality Test

Figure 47. GPS Data Error Modeling

6.1.1.2 Simulation of the Arizona Connected Vehicle Test Bed (Maricopa County)

The simulation platform is used to model the network of 6 intersections in Anthem, AZ. The test bed consists of six signalized traffic intersections along a 2.3-mile stretch of Daisy Mountain Drive. The traffic volume during off peak hours is low. This feature makes Anthem an excellent test bed for testing traffic signal applications. Figure 48 shows the geometric layout of the test bed. Each intersection is equipped with a Savari StreetWave RSE. The dashed lines show the DSRC range (in meters) of the RSEs based on measurements taken in the field. Gavilan Peak and Dedication Trail are two closely spaced intersections and their DSRC ranges have significant overlap. This overlap is captured in the simulation platform.

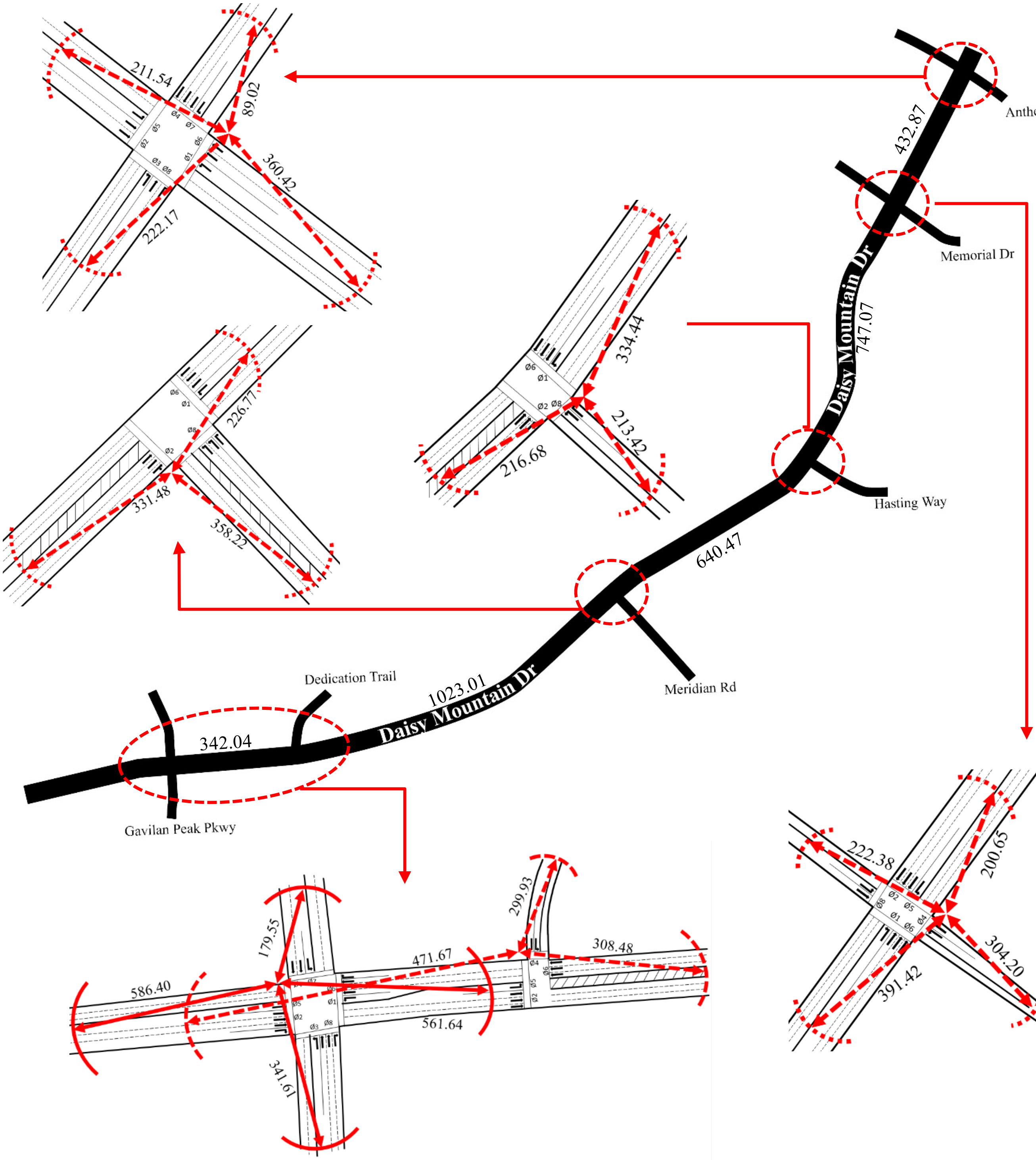


Figure 48. The Geometric Layout of Test Bed (Distances are in Meters)

Table 20 presents the DSRC range of each intersection in the north, south, east and west directions. The data are based on four on-road experimental observations for each intersection. Table 20 shows that there is a difference in the DSRC range for each intersection based on a vehicle leaving or approaching the intersection. There may be several reasons that cause these differences. Factors such as separation distance, signal propagation environment, relative vehicle velocity and effective vehicle velocity, DSRC radio transmission power and modulation rate are among controllable factors that can affect the range of DSRC *(Bai, Stancil, & Krishnan, 2010)*. DSRC range follows a statistical distribution that depends on a packet delivery ratio. Because it is not possible to obtain a statistical inference out of small number of observations, the reported DSRC range is the greatest distance from the RSE from where 95% of the packet were received.

Table 20. Intersection DSRC Range (meters)

	Gavilan		**Dedication**		**Meridian**		**Hastings**		**Memorial**		**Anthem**	
Move-ment	*App.*	*Leav.*	*App.*	*Leav.*	*App.*	*Leav.*	*App.*	*Leav.*	*App.*	*Leav.*	*App.*	*Leav.*
NB	318.2	175.6	-	329.2	368.4	-	204.6	-	496.7	484.1	382.5	88.3
EB	636.6	596.1	459.7	317.9	315.3	245.5	220.4	310.1	231.3	308.0	204.1	387.1
SB	183.6	365.0	270.7	-	-	348.1	-	222.2	474.6	286.1	91.0	361.8
WB	527.2	536.2	299.1	483.6	208.0	347.7	358.7	213.0	300.4	213.5	333.8	219.0

A microscopic simulation platform should have a proper calibration and validation process. Without the calibration, the results could be misleading and lose the trust of the decision makers. In the proposed simulations platform, the DSRC range of each RSE radio is calibrated using data from Table 20. Since traffic volume during the on-road testing hours (off-peak hours) is very low, the average vehicle input in VISSIM model is 200 (vphpl). Once the simulation model is calibrated, the vehicle volume can be increased to consider more congested traffic flow.

In order to show the capability of the simulation platform, an experimental study was designed. Two bus lines are considered in the simulation model. One bus route is from Anthem Way to Gavilan Peak southbound and the other one is from Gavilan Peak to Anthem Way northbound. The headway for each lines is assumed to be 10 minutes. The travel time collection section is set to be from Anthem Way to Gavilan Peak in the simulation platform. Travel times are collected based on two scenarios: with and without implementing the priority signal control application. The scenario is simulated for a duration of 1.5 hours and the average bus travel time is obtained from the second 45 minutes of simulation.

Field travel times for this section were collected for 4 trips from Anthem to Gavilan and vice versa to compare with the results from the simulation platform. In the first 2 trips, the multi-modal priority server application ran on each of the RSEs in the corridor and give priority to the bus. In the last 2 trips, no priority was provided for the bus. The average travel time was calculated for each scenario. Figure 49 shows the average bus travel time with and without priority in the field test and simulated platform. The result shows that the difference between simulated and field average travel time without applying priority control is less than 2% which indicates the simulation platform is well calibrated. In the field, the priority server application reduces the average travel time of the bus to 21% while this reduction is 16% in the simulation platform. These results are not statistically significant due to the limited number of trips taken in each scenario, but they do indicate that the simulation and field tests produce similar and logical results.

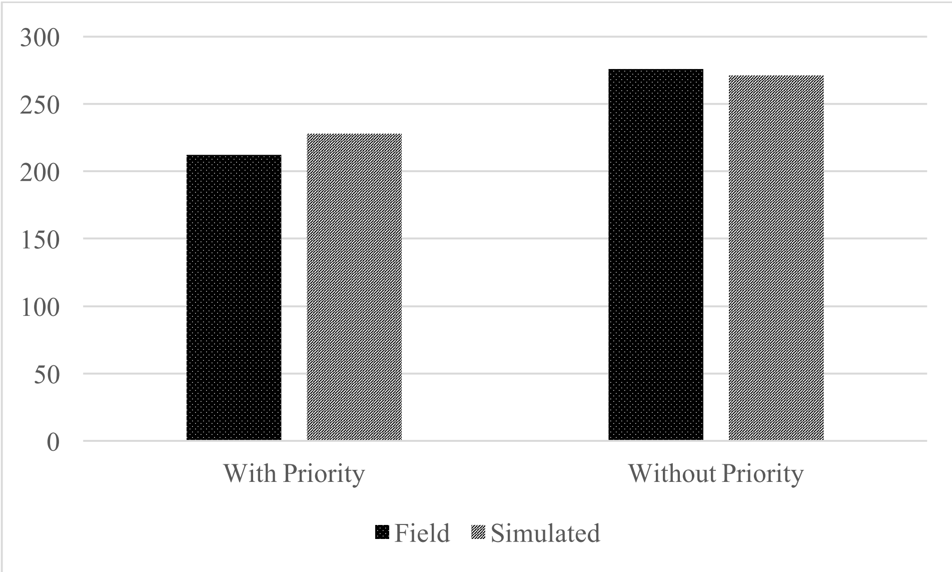


Figure 49. Comparing Simulation Platform Output with Field Data

6.1.2 Intelligent Signal Control Experiments and Analysis

The simulation environment was used to explore the effectiveness of the MMITSS Intelligent Signal Control (ISIG) application at one intersection in the Arizona Connected Vehicle Test Bed – the intersection of Gavilan Peak and Daisy Mountain. Several aspects of ISIG were evaluated including the performance of the estimation of vehicle location and speed (ELVS) algorithm and the performance of the phase allocation algorithm that utilizes the ELVS estimates.

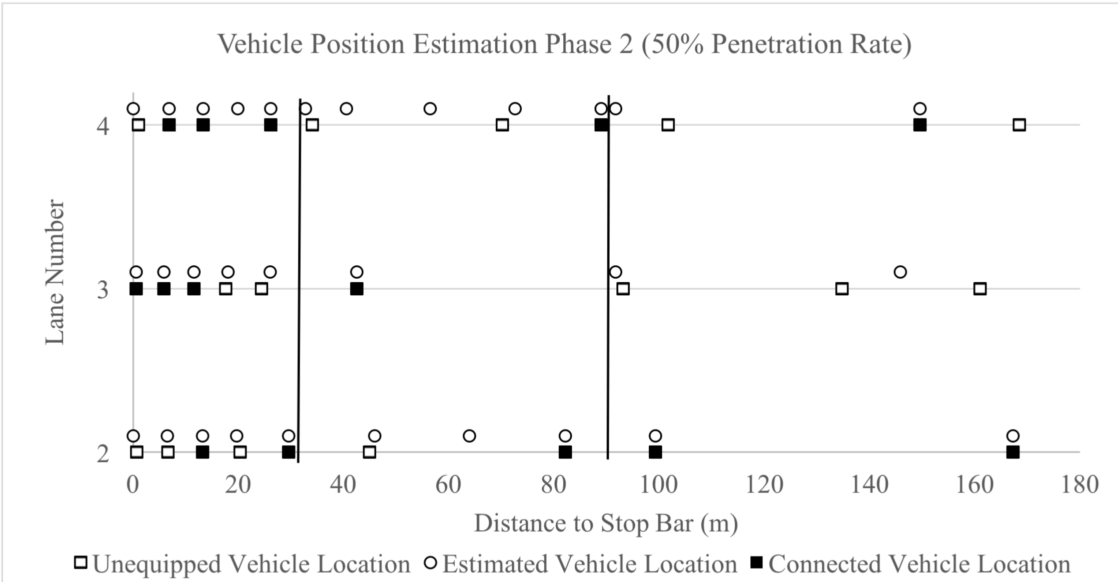
An example of the results of the EVLS algorithm are shown in Figure 50. This figure shows the estimated vehicle locations versus actual vehicle locations of both connected vehicles and unequipped vehicles for phase 2 (eastbound through movement) with different penetration rates (50% and 25%) under medium demand level. The vertical lines are the estimated boundaries for the queuing region, slow-down region, and free-flow region (from left to right).

The estimation of vehicles in the queuing region has the best performance which indicates the queue propagation speed and the average vehicle length is estimated accurately. The estimation in the slow-down region tends to insert more vehicles in some cases such as lane 4 with 50% penetration rate because the car-following distance is larger than two times the maximum following distance, causing the insertion to be triggered. In this case, two more vehicles are inserted based on Wiedemann's car following model. However, in reality, different driving behaviors may result in longer car following distances, especially on the approach to a queue at a red traffic signal. On the other hand, in lane four of the 25% penetration rate case, there is no connected vehicle in the slow-down region, hence no vehicle insertion is triggered. For the free flow region, since the insertion location is randomly selected, the location of the inserted vehicles are not very accurate. However, as discussed before, errors in free flow region have much less impact to the phase allocation algorithm than errors in queuing region. Overall, the performance of the algorithm shows that the estimated vehicle locations are similar to the true vehicle locations.

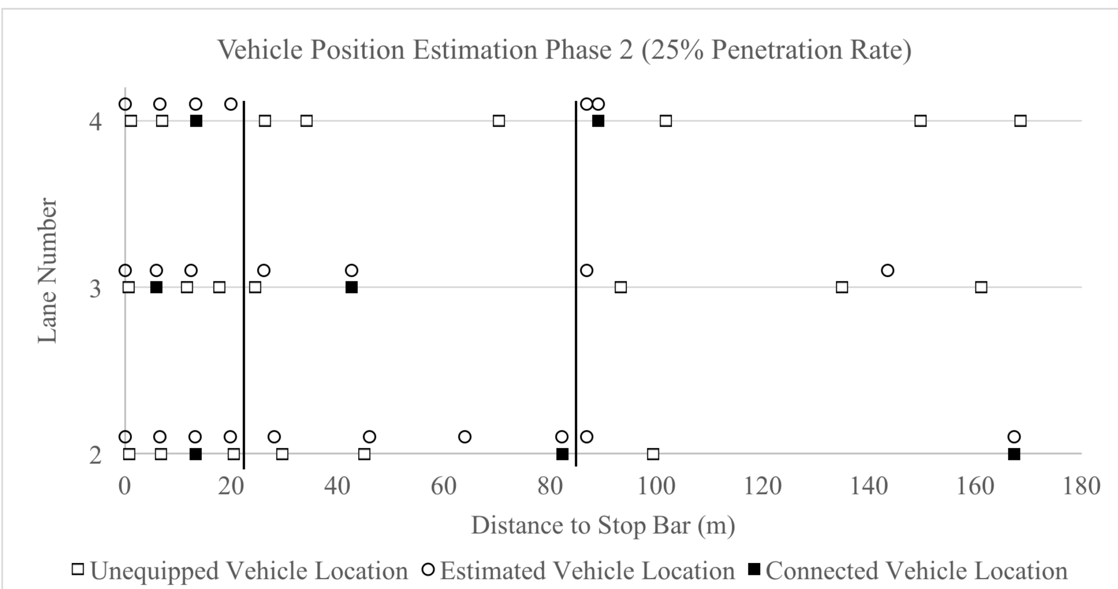
Based on the results of the EVLS algorithm, the complete arrival table is constructed and used as the input for the phase allocation algorithm. Different scenarios with two different demand levels and four penetration rates were tested. The two demands levels are 500 and 667 veh/h/lane eastbound and westbound and 375 and 500 veh/h/lane northbound and southbound. Four sets of penetration rates are considered: 100%, 75%, 50%, and 25%. A total of 1125 seconds are simulated for each scenario with 125s of warm-up period and 1000s of data collection time.

The algorithms are run with different objective functions: minimizing total vehicle delay and minimizing queue length. The results of the algorithms are also compared to well-tuned fully actuated control. The unit extension time of the fully actuated control is set to be 1.4s which is estimated by the recommendation from FHWA (138). Table 21 and Table 22 show the performance of the algorithms with different objective functions.

The results show an improvement of the proposed algorithm compared to actuated control when the penetration rate is equal to or greater than 50% in almost all cases. Under the 100% penetration situation, the total delay is decreased by 10.04% and 14.67% under two demand levels when minimizing total vehicle delay, respectively, and 6.37% and 16.33% when minimizing queue length. As the penetration rate decreases, the total delay tends to increase. Under the 500/375 demand level scenario, the performance of the algorithms under the 25% penetration rate assumption have higher total delay than actuated control for both objective functions. However, under the 667/500 demand level, both objective functions outperform actuated control. With the higher demand, the total number of connected vehicle is increased and the estimation errors in the EVLS algorithm are less.



(a) Result of EVLS algorithm with 50% penetration rate



(b) Result of EVLS algorithm with 25% penetration rate

Figure 50. Estimated Locations of EVLS Algorithm with Different Penetration Rates

Table 21. Comparison of Average (s/veh) Vehicle Delay for Each Phase and Total Vehicle Delay (s)

Delay (s) Under Different Scenarios (Minimization of Total Vehicle Delay)									
500/375 Demand Level	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	Total
Actuation	36.22	30.65	34.28	35.39	36.82	27.46	38.19	31.64	39546.26 (0%)
100% PR	43.98	22.85	44.32	26.89	42.12	22.32	37.74	26.90	35576.17 (-10.04%)
75% PR	43.75	23.26	44.36	27.15	36.02	25.50	32.27	34.32	35412.19 (-10.45%)
50% PR	40.22	24.48	40.75	31.25	36.80	23.45	39.20	26.79	35806.88 (-9.46%)
25% PR	40.39	28.37	41.32	30.89	34.95	29.32	42.35	33.62	39874.87 (+0.83%)
667/500 Demand Level	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	Total
Actuation	50.08	38.50	55.96	38.96	57.15	35.13	51.72	37.76	64892.80 (0%)
100% PR	63.22	28.70	64.13	32.74	47.16	27.59	50.20	30.74	55371.99 (-14.67%)
75% PR	61.71	30.89	57.01	34.33	61.01	30.76	43.32	31.96	59004.30 (-9.07%)
50% PR	56.97	31.24	71.61	35.64	45.44	26.80	42.43	33.67	56945.22 (-12.25%)
25% PR	47.98	31.21	55.19	34.47	42.12	32.32	47.85	32.93	57094.73 (-12.02%)

Table 22. Comparison of Average (s/veh) Vehicle Delay for Each Phase and Total Vehicle Delay (s) Under Different Scenarios (Minimization of Queue Length)

Delay (s) Under Different Scenarios (Minimization of Queue Length)									
500/375 Demand Level	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	Total
Actuation	36.22	30.65	34.28	35.39	36.82	27.46	38.19	31.64	39546.26 (0%)
100% PR	33.49	30.37	38.70	29.91	36.47	25.27	31.40	27.63	37027.40 (-6.37%)
75% PR	34.24	28.07	38.89	34.66	33.00	28.97	34.66	30.72	38443.46 (-2.79%)
50% PR	28.22	31.90	33.62	28.01	34.27	25.90	35.82	28.03	36538.61 (-7.61%)
25% PR	35.98	39.87	33.92	33.79	31.50	30.06	33.97	29.77	41675.62 (+5.38%)
667/500 Demand Level	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	Total
Actuation	50.08	38.50	55.96	38.96	57.15	35.13	51.72	37.76	64892.80 (0%)
100% PR	39.44	34.75	41.85	33.51	34.84	28.93	36.26	33.43	54297.80 (-16.33%)
75% PR	54.25	37.51	53.22	35.36	34.87	34.94	43.07	41.21	62140.09 (-4.24%)
50% PR	44.75	34.30	53.36	37.07	37.99	32.95	41.61	34.26	58856.05 (-9.30%)
25% PR	41.60	38.49	52.56	36.97	37.70	42.46	39.47	38.02	64365.98 (-0.81%)

The average vehicle delay of each phase between two objective functions as well as actuated control are compared in a radar diagram as shown in Figure 51. The radar diagram allows visualization of the performance measures by movement (phase) at the intersection. In a connected vehicle environment, it is possible to directly observe the performance of connected vehicles by movement and by mode. The radar diagram provides an interesting visualization of the performance by phase. For example, consider the 100% penetration rate case with traffic demand of 667/500. The average vehicle delay for the left turning phases (odd-numbered phases) is significantly higher than the delay for the through phases (even numbers) since the number of vehicles served by the left turn phase is less per unit green that for the through movements, hence the algorithm favors the through movements when minimizing total delay. However, this difference isn't observed when the objective is to minimize queue length or when actuated control is used.

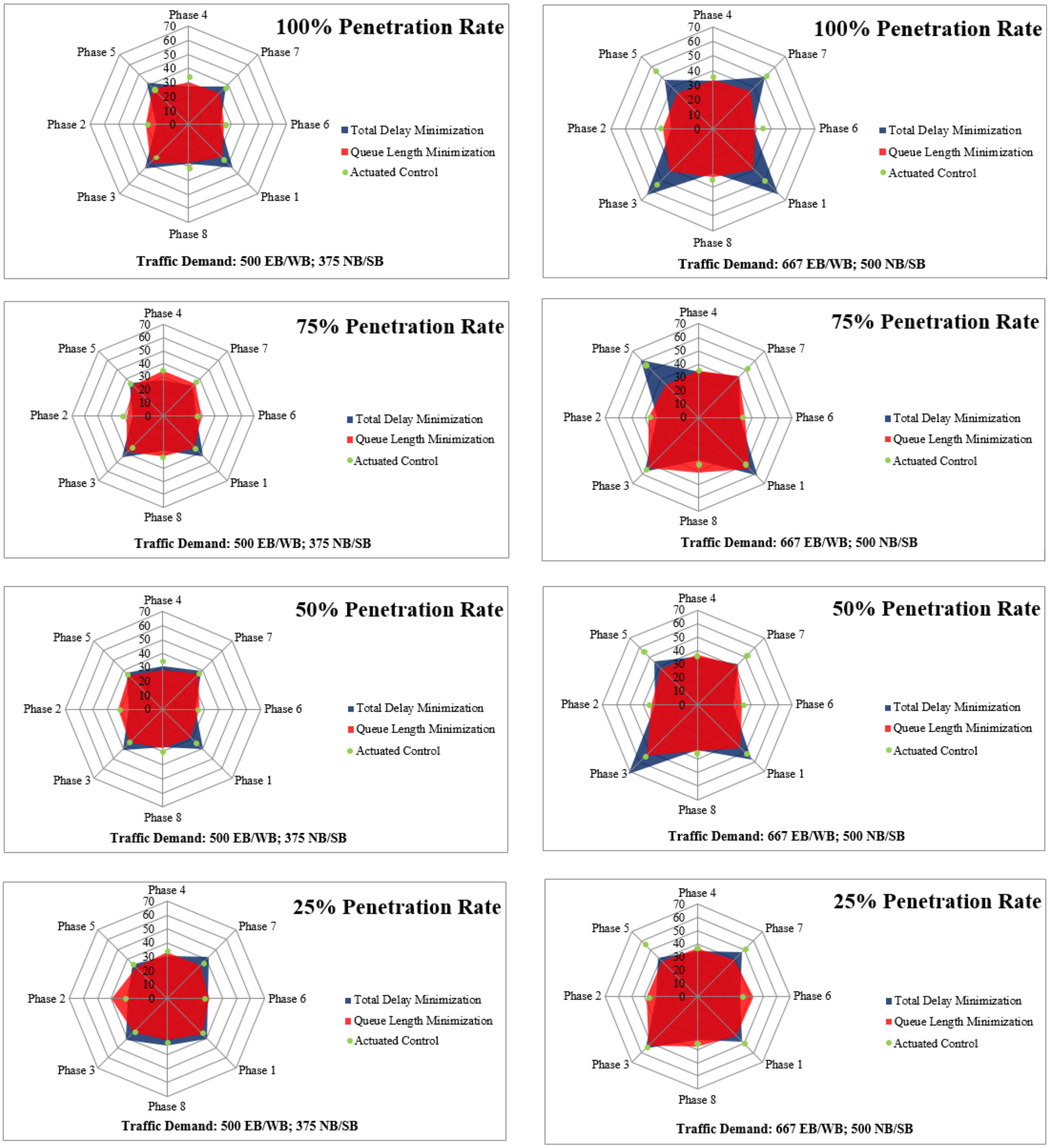


Figure 51. Comparison of Average Vehicle Delay of Each Phase Under Different Objective Functions

In general, minimization of total vehicle delay generates lower total vehicle delay compared to minimization of queue length. However, the variance of vehicle delay of each phase is much higher when minimizing total delay as shown in Table 23. The average vehicle delays of left turn phases are much higher than the delays of those through phases. Because the phase allocation algorithm assigns the green duration based on vehicle delay, phases with higher volumes (number of vehicles) receive longer duration and priority which means the phase will be served first within a barrier group. In addition, the algorithm may terminate a left turn phase when the queue has not been fully discharged because the residual queue is not able to introduce enough delay compared to through phases. When the demand is higher, the difference is more significant. On the other hand, when minimizing the queue length, each phase is served more equally, but usually results in higher total vehicle delay. It depends on the policy makers to decide which strategy to implement when operating the signal. For example, if the traffic demand is high, the policy makers can use minimization of total vehicle delay as the objective function to reduce congestion. If the demand is low, minimization of queue length can be used to balance the service of each phase.

Table 23. Variance of Average Delay over All Phases

Penetration Rate	100%	75%	50%	25%
500/375 Demand Level	79.21 / 17.24	59.23 / 10.93	46.29 / 11.56	27.23 / 9.61
667/600 Demand Level	202.75 / 13.59	169.78 / 55.29	195.24 / 41.21	71.91 / 22.59

Note: The values are: Variance of minimizing total vehicle delay / Variance of minimizing queue length

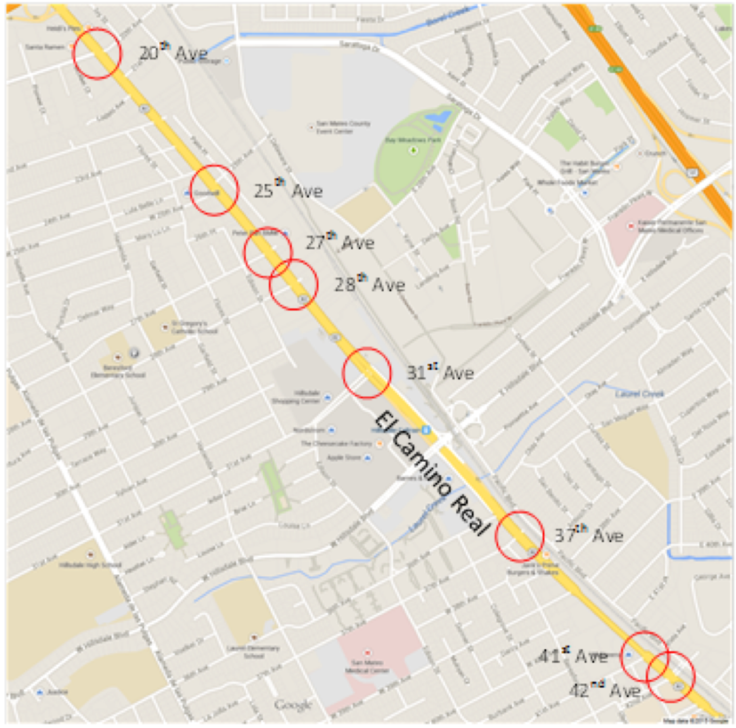
6.1.3 Transit and Freight Signal Priority Experiments and Analysis

The AZ MMITSS priority control framework was implemented on two sample network models based on a San Mateo, CA network and the Anthem, AZ network (Note: The simulation analysis in the San Mateo network was conducted as part of another study under an IDIQ Task through Kittleson and Associates, but is included in this report to show the AZ MMITSS priority algorithm performance in different networks). VISSIM models were created and calibrated to match the field sites so that GPS data could be created for connected vehicle operations. In the first scenario in both networks, transit vehicles are considered as priority vehicles. In the second scenario in AZ network, transit vehicles and trucks are considered as priority vehicles.

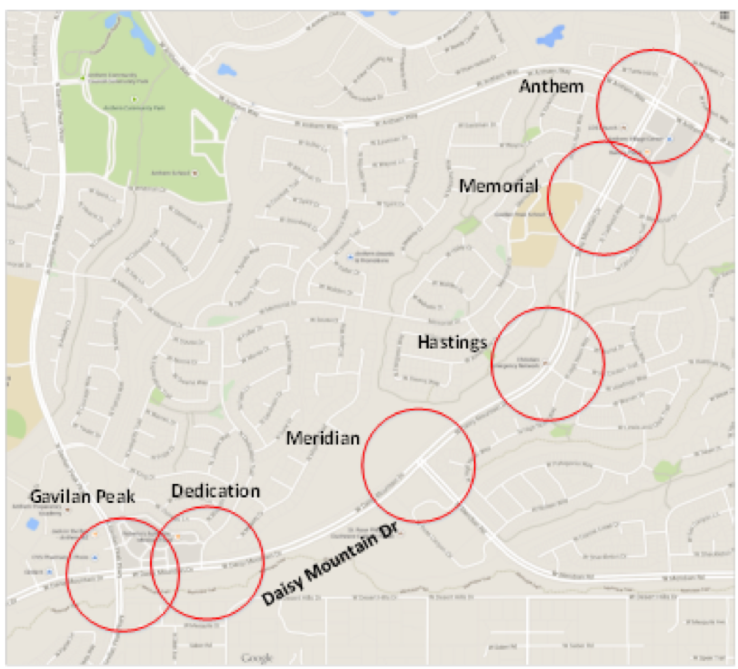
6.1.3.1 San Mateo, CA Network

The simulation network is a segment along El Camino Real that consists of 8 intersections from 20th Ave. to 42nd Ave. (Figure 52(a)). The main street (El Camino Real) has two to three lanes each direction and the side streets have one to two lanes. Some intersections are closely spaced where the distance between the intersections is approximately 100m. The total simulation period is 5 hours with 30 minutes of warm up time (0s-1800s) and 4 hours and 30 minutes of data collection period (1800s-18000s). There are five different traffic demand levels over the time period. Each level lasts for one hour except for the last demand level that is only for half an hour. The demand levels and corresponding time are defined as follows:

- Level 1: 1800s-5400s (Low)
- Level 2: 5400s-9000s (Medium)
- Level 3: 9000s-12600s (High)
- Level 4: 12600s-16200s (Medium)
- Level 5: 16200s-18000s (Low)



(a) San Mateo, CA



(b) Anthem, AZ

Figure 52. Network Overview (source: google map)

The low, medium, and high demands are 1000 (veh/h), 1200 (veh/h), and 1500 (veh/h) on the main street, respectively. The side streets demand is increased by the same ratio as the main street. The traffic signal control simulation uses the Econolite ASC/3 SIL which supports the NTICP interface. The vehicle composition is 98% passenger vehicles and 2% trucks. There are 14 bus stops in this arterial, 7 bus stops in the northbound direction and 7 bus stops in the southbound direction. The headway is 10 minutes in both directions. The dwell time distribution of transit vehicles at each bus stop is distributed as a normal random variable with a mean of 20 seconds and variance of 2 seconds (N(20, 2)). The results presented in this section are from the base case test scenario in which each intersection is operated independently without coordination. Priority is provided for transit vehicles only. Fully actuated control is implemented with transit signal priority.

Average bus travel time and average bus delay in the southbound and northbound directions for 5 different simulation replications are presented in Table 24. The performance of the priority control model with respect to these measurements is compared to fully actuated control.

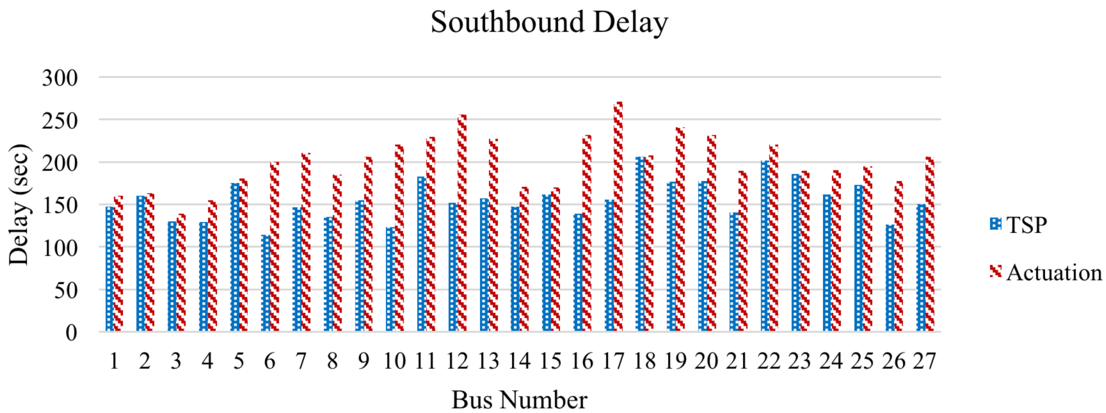
Table 24. Fully actuated control (baseline)

	Southbound	Northbound
Average Transit Vehicles Delay (sec)	200.92	224.33
Average Transit Vehicles TT (sec)	586.49	607.42
Transit Vehicles TT Standard Deviation	42.90	45.71
Average Weighted Delay of Regular Vehicles		18.23

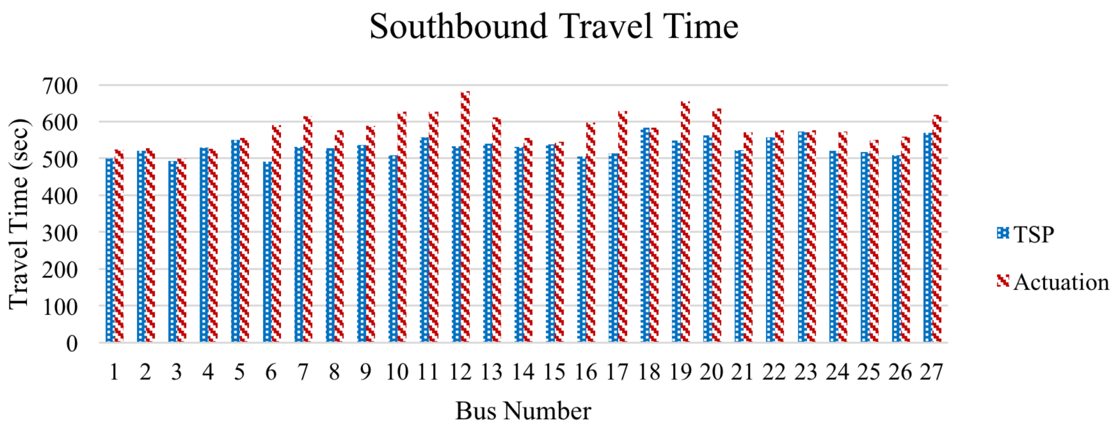
Table 25 summarizes the benefits of transit priority compared to the baseline case. The average transit travel time is decreased by 9.85% in the northbound direction and by 8.02% in the southbound direction. The delay is decreased by 24.04% in northbound and 23.20% in southbound directions. Comparison of the average travel time standard deviation shows that the model increased the reliability of travel time by 42.8% in southbound direction and 41.91% in northbound direction. Figure 53(a) and (b) illustrate bus-by-bus comparison of each bus under fully actuated control and transit priority control for the northbound direction considering bus delay and bus travel time, respectively. Figure 54(a) and (b) show the southbound comparison. These figures show that transit priority is beneficial for almost every individual transit vehicle.

Table 25. Transit priority

	Southbound	%	Northbound	%
Average Transit Vehicles Delay (sec)	154.27	**-23.20**	170.42	**-24.04**
Average Transit Vehicles TT (sec)	539.45	**-8.02**	547.55	**-9.85**
Transit Vehicles TT Standard Deviation	24.05	**-42.8**	26.54	**-41.91**
Average Weighted Delay of Regular Vehicles			18.94	**3.9**

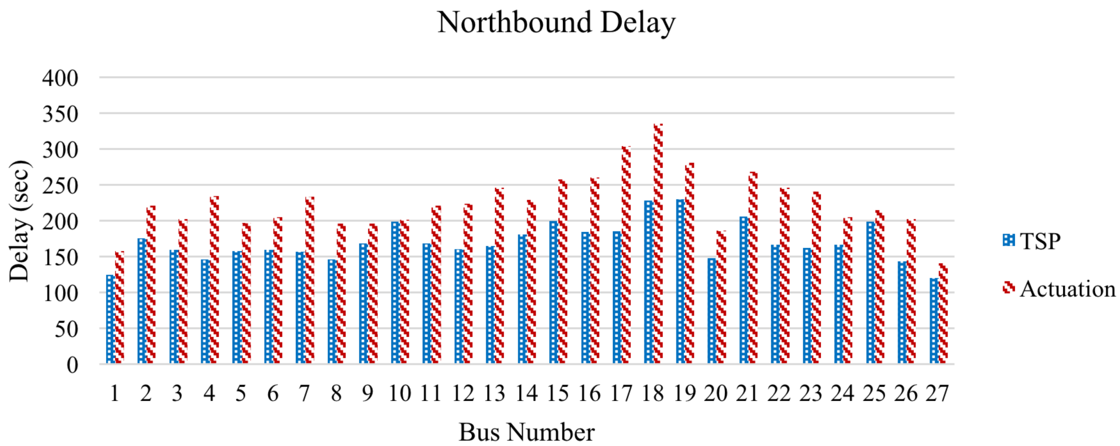


(a)

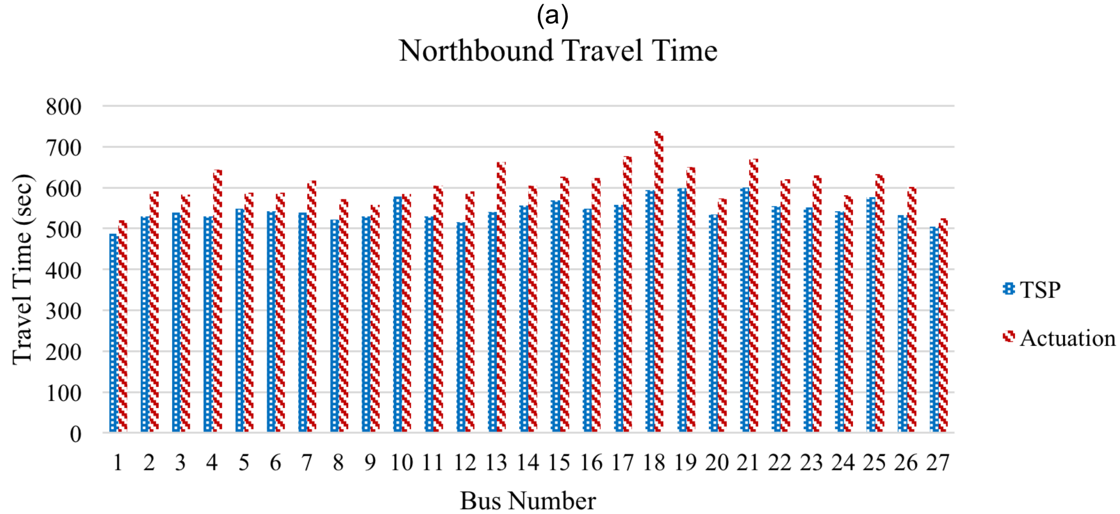


(b)

Figure 53. Southbound Delay and Travel Time in California Network



(a)



(b)

Figure 54. Northbound Delay and Travel Time in California Network

6.1.3.2 Arizona Network (Maricopa County)

The Arizona simulation network is a segment along Daisy Mountain Drive that consists of 6 intersections from Gavilan Peak Pkwy to Anthem Way (Figure 52(b)). The main street (Daisy Mountain Dr.) has three lanes in each direction and the side streets have one to two lanes.

The total simulation period is 3 hours with 30 minutes of warm up time and 2 hours and 30 minutes of data collection period. During the data collection period, there are five different traffic demand levels. Each level lasts for half an hour. The demand levels, corresponding time and vehicle composition are defined similar to CA network. The performance of transit priority control is compared to fully actuated control. There are 5 bus stops in this arterial, 3 bus stops in westbound and 2 bus stops in eastbound. All of the bus stops are far side. The dwell time distribution of transit vehicle at each bus stop is a normal random variable with a mean of 20 seconds and variance of 2 seconds (N(20, 2)). Average bus travel time and average bus delay in eastbound and westbound in the corridor for 5 simulation runs are presented in Table 26 for the baseline case.

Table 26. Fully actuated control (baseline)

	Eastbound	Westbound
Average Transit Vehicles Delay (sec)	143.38	144.43
Average Transit Vehicles TT (sec)	409.95	430.25
Transit Vehicles TT Standard Deviation	26.25	24.18
Average Weighted Delay of Regular Vehicles (sec)		21.17

Table 27 and Table 28 summarize the benefits of transit priority compared to the baseline case. The average travel time decreased by 15.24% in the eastbound direction and by 11.54% in the westbound direction. The delay decreased by 43.8% in eastbound and 32.07% in westbound. Comparison of the average travel time standard deviation shows that transit priority decreased the reliability of travel time by 3.02% in the eastbound direction and increased it by 24.83% in the westbound direction. Also, the average weighted delay of regular vehicles decreased by 0.06%. Average weighted delay of the regular vehicles in the network was defined as the number of served vehicles in each section times the average delay of the vehicles in that section divided by the total number of vehicles in all sections.

Table 27. Transit Priority Results

	Eastbound	%	Westbound	%
Average Transit Vehicles Delay (sec)	80.56	**-43.8**	98.01	**-32.07**
Average Transit Vehicles TT (sec)	347.44	**-15.24**	364.02	**-11.54**
Transit Vehicles TT Standard Deviation	27.8	**3.02**	22.98	**-24.83**
Average Weighted Delay of Regular Vehicles (sec)			21.04	**-0.06**

Figure 55 (a) and (b) illustrate bus-by-bus comparison of each bus under fully actuated control and transit priority control for the eastbound travel considering bus delay and bus travel time, respectively. Figure 56(a) and (b) show the westbound comparison. These figures show that on average, transit priority is beneficial for each individual bus in the 5 simulation replications.

Both the Arizona and California simulation networks consisted of regular vehicles and transit vehicles. In order to investigate the performance of the proposed framework when there are multiple conflicting priority requests, another scenario was designed to consider priority for both trucks and transit vehicles.

Trucks composed 2% of the vehicles in the AZ network and do not have specific origins or destinations. Therefore, they may have a priority conflict with each other or with transit vehicles. Transit routes and departure times of transit vehicles remained the same as the previous case study. The results of 5 simulation runs showed 9.7% and 6.7% improvement in the travel time of buses in the eastbound and westbound directions, respectively (Table 28). Average bus delay reduces by 41.3% and 24.1% in each direction. Average travel time and delay of trucks reduced by 5.1% and 9.1% correspondingly. Meanwhile, the average weighted delay of all regular vehicles increased by 9.8%.

Table 28. Transit and Truck Priority Results

	Fully Actuated	FSP + TSP	%
Eastbound Average Transit Vehicles Delay (sec)	139.3	98.3	**-29.4**
Eastbound Average Transit Vehicles TT (sec)	421.1	380.1	**-9.7**
Westbound Average Transit Vehicles Delay (sec)	116.7	88.6	**-24.1**
Westbound Average Transit Vehicles TT (sec)	417.2	389.1	**-6.7**
Average Trucks Delay (sec)	19.8	18.0	**-9.1**
Average Trucks TT (sec)	57.1	54.1	**-5.1**
Average Weighted Delay of Regular Vehicles (sec)	21.21	23.28	**9.8**

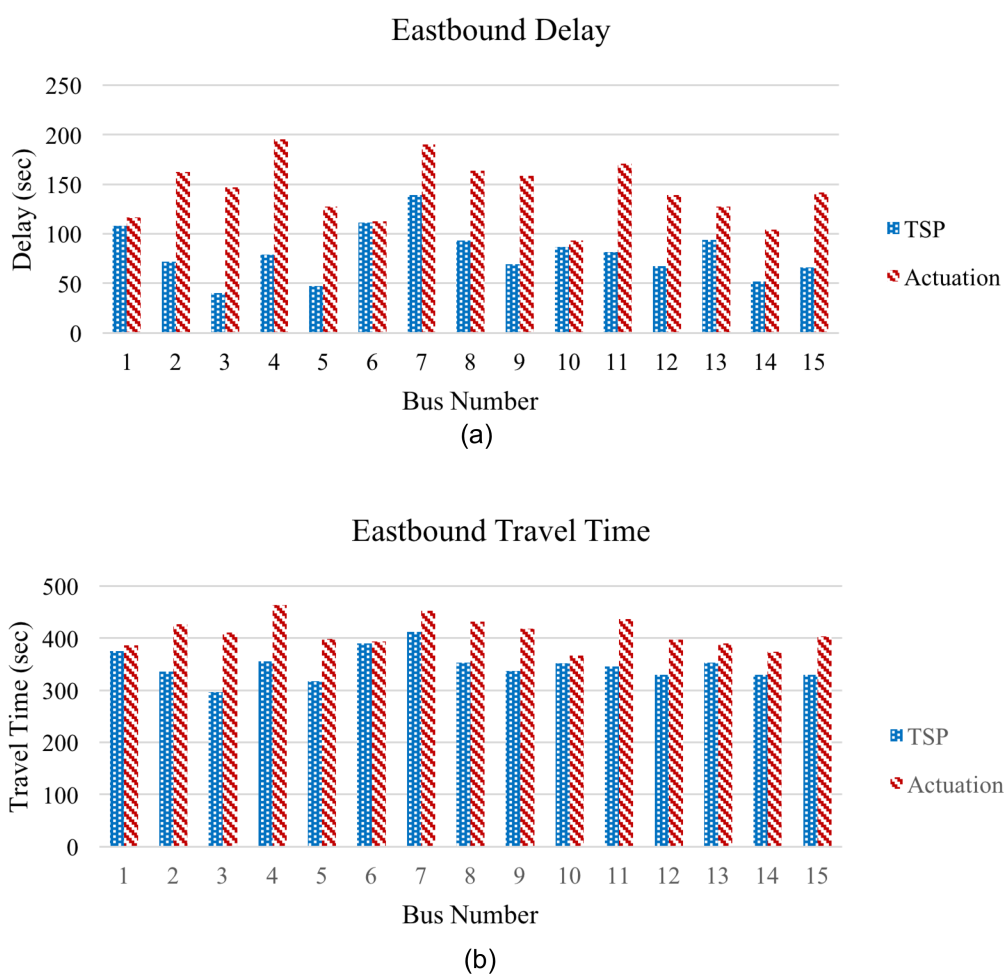
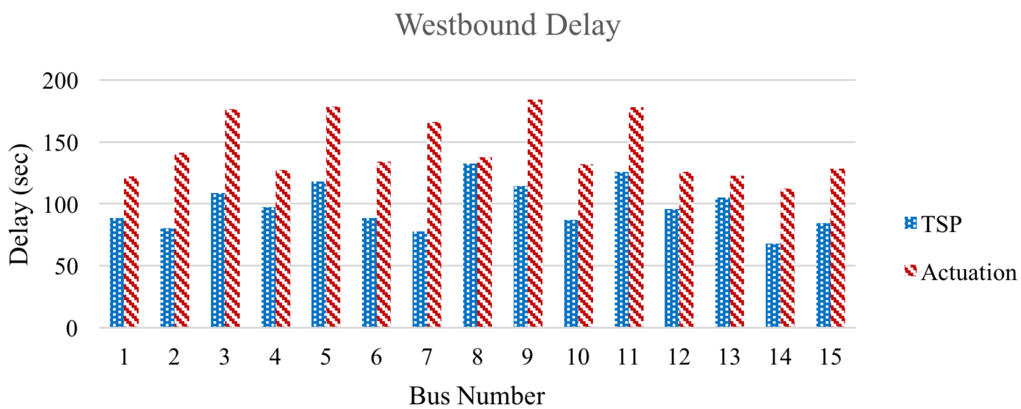
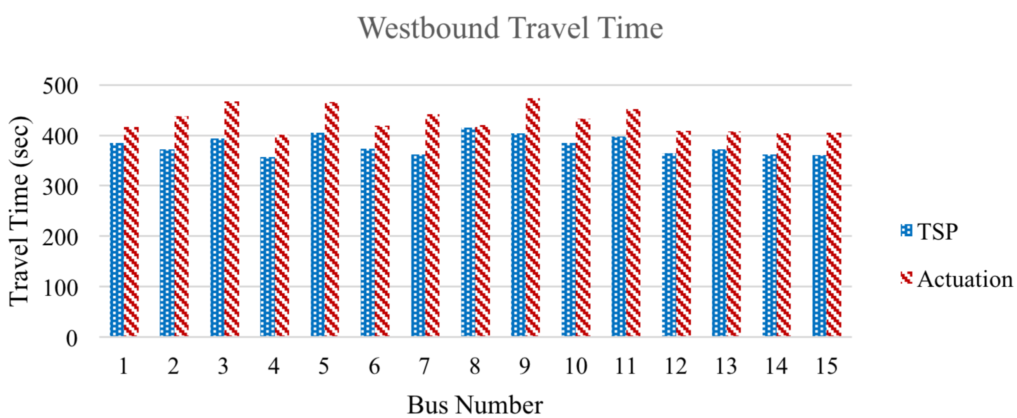


Figure 55. Eastbound Delay and Travel Time in Arizona Network



(a)



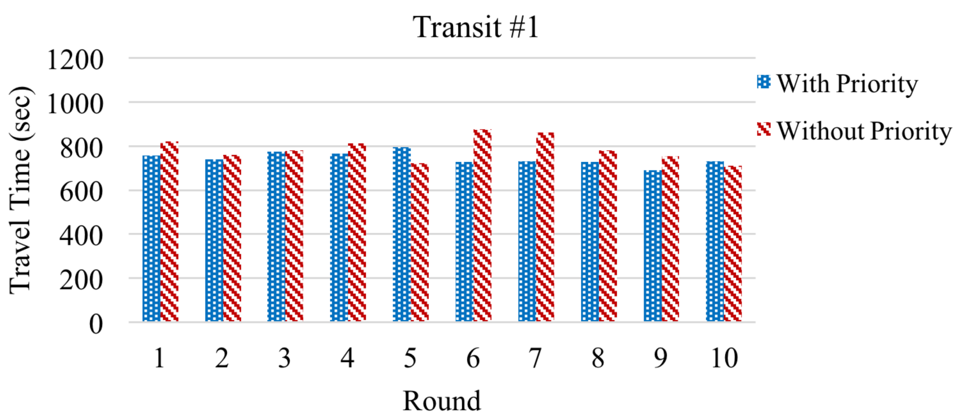
(b)

Figure 56. Westbound Delay and Travel Time in Arizona Network

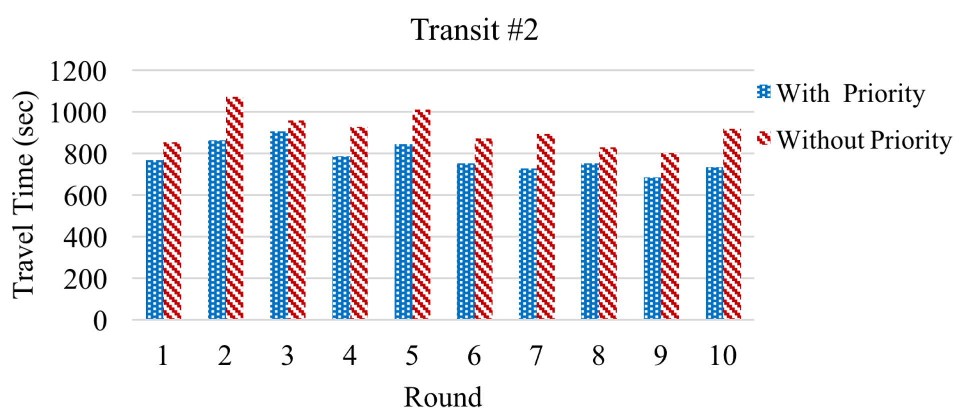
6.1.3.3 Field Testing Results

The MMITSS priority framework was implemented in the field network in the Arizona Connected Vehicle Test Bed in Anthem, AZ. It was tested as part of the MMITSS impact assessment (IA) analysis that was conducted on March 3, 2015 (Tuesday) and March 4, 2015 (Wednesday) in Anthem, AZ. Based on the detailed test plans, on Tuesday two trucks were enabled with signal priority and traveled for 10 round trips in northbound-southbound direction at Gavilan Peak and Daisy Mountain Dr. Meanwhile, two transit vehicles were enabled with signal priority and traveled eastbound-westbound through the network of six (6) intersections for 10 round trips. The transit vehicles headway was 20 minutes. There are 5 far-side bus stops in the network. The average dwell time at each bus stops was 20 seconds. Travel time sections were designed to capture travel time in each round for each vehicle. Wednesday was designated for the base-case data collection using GPS units in the vehicles. No priority was provided on Wednesday. Two trucks and two transit vehicles traveled 10 round trips while their departure times were exactly the same as on Tuesday. Figure 57 (a) and (b) show a one by one comparison of travel time for the two transit vehicles in each one of the 10 round trips on Tuesday (with priority) and on Wednesday (without priority). There was a 10.3% improvement in the average travel time of the two transit vehicles. Travel time standard deviation of the buses decreased by 41.3%. Figure 58 (a) and (b) show the results for the two trucks. GPS data of Truck #1 in the first round was unavailable. On Wednesday, Truck #1 stopped in

Round 5 and 7 for pedestrians (pedestrian clearance interval was 45 seconds). A similar situation occurred for Truck #2 in the fourth round. There was a 3.84% improvement in the average travel time of the two trucks. Travel time standard deviation of the trucks decreased by 21.78%. The results show the effectiveness of the priority control framework in real-world.

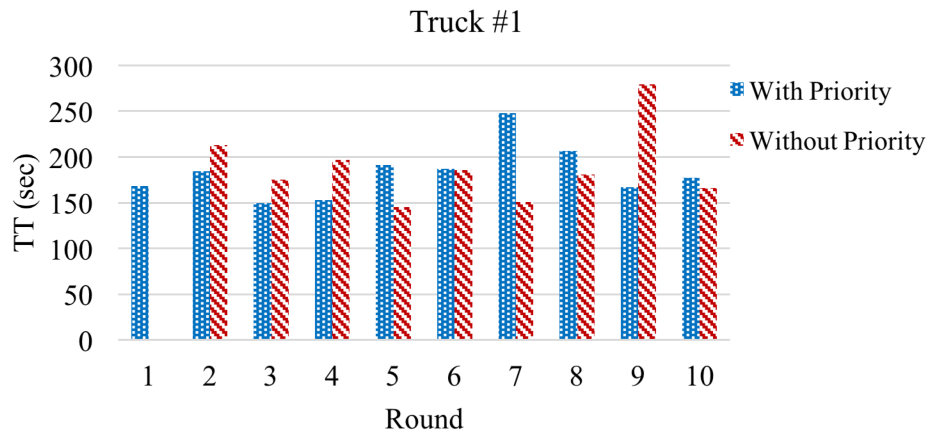


(a)

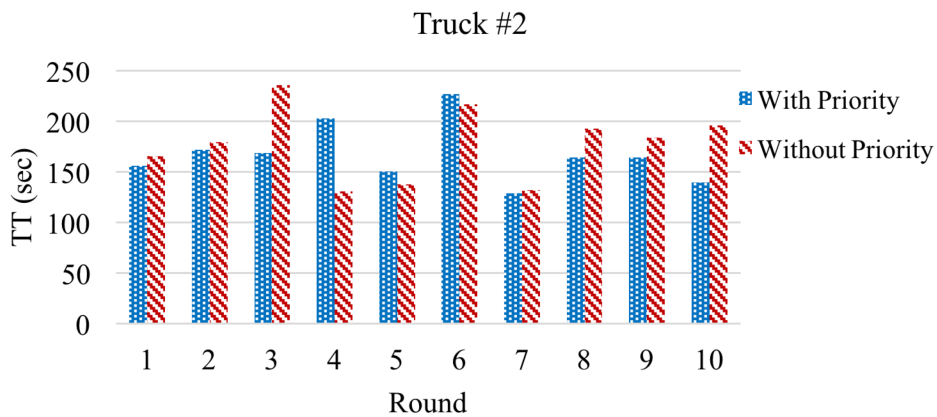


(b)

Figure 57. Transit Travel Time in the Arizona Network



(a)



(b)

Figure 58. Truck Travel Time in the Arizona Network

6.1.4 Performance Observation Experiments and Analysis

The MMITSS Performance Observer is responsible for acquiring data from other MMITSS components, including the Equipped Vehicle Trajectory Aware and the Traffic Controller Interface, to compute performance measures. The Performance Observer includes estimation, managing, and archiving the data. All of the acquired performance observations and metrics are collected and reported in terms of peak period, all day, or specially defined time period.

As stated in the MMITSS Detailed Design document the Basic Safety Message contains the vehicle temporary ID, latitude, longitude, elevation, speed, heading, and vehicle size. It doesn't have vehicle type or class information, although it can be determined using the Signal Request Message if the vehicle is priority eligible and making a priority request. The high resolution data (i.e. BSMs received every 0.1 second) are used to reconstruct the vehicle trajectories. These trajectories are used to compute many of the performance measures. Table 29 summarizes the performance measures being calculated and estimated in this component with their associated sources of data.

Table 29. Summary of Performance Measures being Calculated and Estimated

Performance Metric	Abbreviation	Unit	Data Source
Travel Time	TT	Second	MRP_EquippedVehicleTrajectoryAware
Delay	D	Second	MRP_EquippedVehicleTrajectoryAware
Travel Time Variability	TTV	Second	MRP_EquippedVehicleTrajectoryAware
Delay Variability	DV	Second	MRP_EquippedVehicleTrajectoryAware
Queue Length	QL	Meter/number of vehicles	MRP_EquippedVehicleTrajectoryAware
Number of Stops	NS		MRP_EquippedVehicleTrajectoryAware
Volume	V	Number of Vehicles	MRP_TrafficControllerInterface
Occupancy	O	%	MRP_TrafficControllerInterface
DSRC Range	RNG	Meter	MRP_EquippedVehicleTrajectoryAware
Packet Loss Rate	PLR	%	MRP_EquippedVehicleTrajectoryAware
Market Penetration Rate	MPR	%	MRP_EquippedVehicleTrajectoryAware & MRP_TrafficControllerInterface

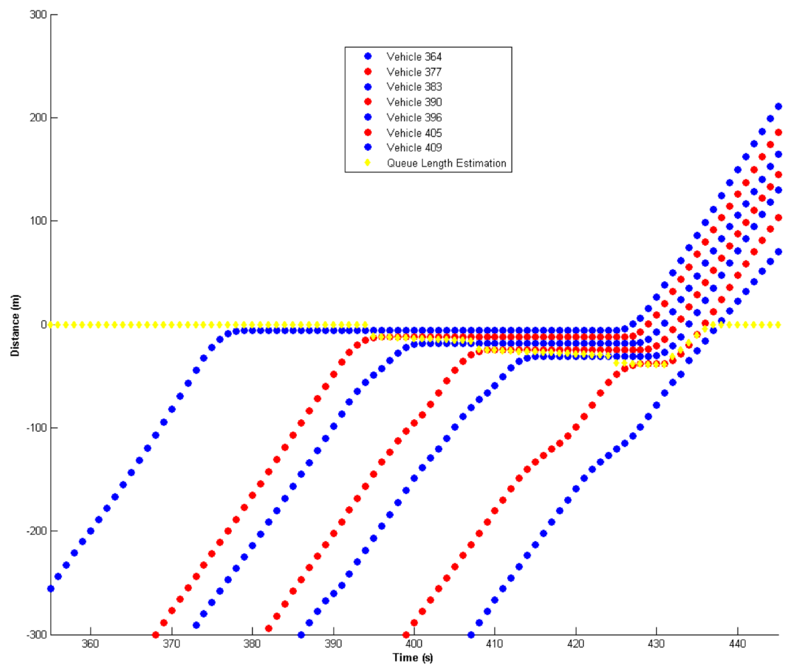
Both simulation and field data collected during different periods of time from Anthem network. Different performance measures are first introduced and then comparison of numerical results for different scenarios will be presented.

6.1.4.1 Queue Length

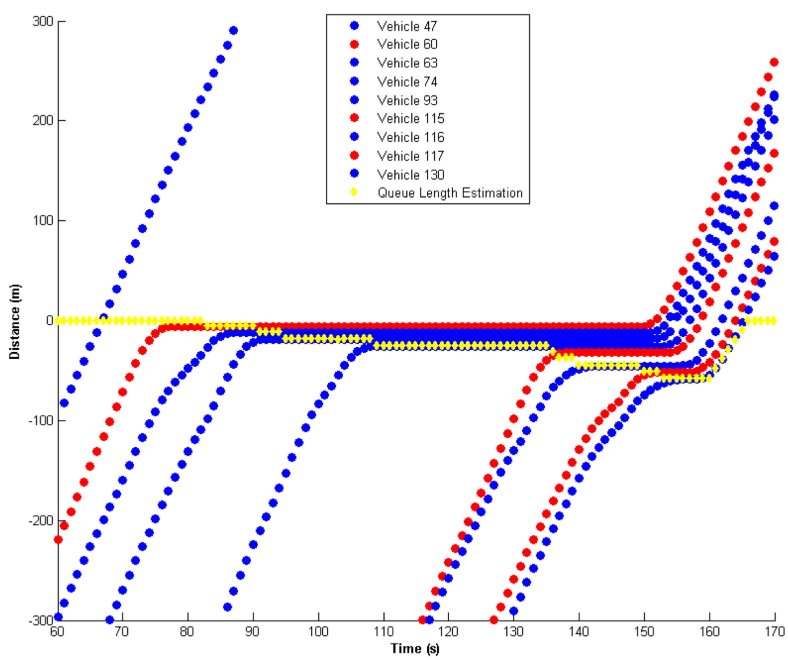
An important performance metric being analyzed at the intersection level is queue length. As explained in the design document the functional description of this measure is divided into two parts: Observing and estimating the queue length in real time in terms of the distance of the last stopped vehicle from the stop bar for each lane, and/or measuring the queue in terms of the number of stopped vehicles at each lane. These measures can be easily transformed into each other. For example, according to the data available from BSMs coming from the OBEs (Vehicle Trajectory Awareness), the length of each individual vehicle is known, and by knowing the number of vehicles in the queue, the overall queue length in terms of the distance from the stop bar can be calculated.

The main goal is to have an accurate and acceptable estimation of this measure when the market penetration rate is low. The queue length algorithm addresses the market penetration issue by adding a parameter called Queue Increase Rate. By having information about the last stopped vehicle a lower bound for the queue length is provided. Then by adding the Queue Increase Rate factor, which is calculated based on the combination of historical and real-time data coming from the loop detectors to the current known queue status, an accurate estimate of the queue length is achieved. Volume data is valuable for the lanes that are equipped with a healthy system detector.

Simulated results of southbound movement on Daisy Mountain and Gavilan Peak are shown in Figure 59. Blue dots represent the trajectory of unequipped vehicles, red dots show the trajectory of equipped vehicles, and the yellow line shows the estimated queue length. Figure 59(a) is for the southbound left lane with no system detector, and the estimation is purely based on the arrival rate of connected vehicles. Figure 59(b) is for the southbound right lane which has an active system detector that helps the estimated value to be more accurate.



(a)



(b)

Figure 59. Simulated Queue Length Estimation results: (a) with Detectors, (b) without Detectors

Field data collection for validating the results of this methodology was done on November 4th, 2014. The southbound right lane at the intersection of Daisy Mountain and Gavilan Peak was selected. A camera was placed 150ft away from the intersection to capture the queue. Table 30 shows the comparison between the observed queue length using the video recorded data and the estimated value using the system detectors data and BSMs received.

Table 30. Field Queue Length Estimation using System Detectors Data

Iteration	Video Time	Real Time	# of CVs in the Queue	Estimated Maximum Queue Length (meters ~ vehicles)	Observed Maximum Queue Length (meters ~ vehicles)	Penetration Rate
#1	9:07	15:57:48	1	16.5m ~ 3	18.5m ~ 3	33.3%
#2	10:48	15:59:33	0	27m ~ 4	13.5m ~ 2	0%
#3	12:43	16:01:10	1	35.84m ~ 6	32m ~ 5	20%
#4	14:10	16:03:10	0	13.5m ~ 2	15m ~ 2	0%
#5	16:01	16:04:39	1	43.5m ~ 7	70m ~ 8	12.5%
#6	17:50	16:06:16	0	54m ~ 8	50m ~ 8	0%
#7	20:09	16:08:36	1	6.09m ~ 1	6m ~ 1	100%
#8	21:28	16:10:00	0	13.5m ~ 2	6.5m ~ 1	0%
#9	22:37	16:11:17	2	49.58m ~ 8	40m ~ 6	33.3%

6.1.4.2 Travel Time and Delay

As explained in the design document, travel time and delay measures are characterized based on the data acquired from all the equipped vehicles traversing the intersection. Figure 60 shows the simulation results of travel time studies at the intersection of Daisy Mountain and Memorial Drive for northbound through movement under different market penetration rates. The estimated travel time values are accumulated every 5 minutes and as the penetration rate decreases the variation in estimating the travel time of all the vehicles increases. The purpose of this analysis is to show that the MMITSS Performance Observer can estimate the population travel time based only on observations of some percentage of connected vehicles (e.g. market penetration).

Field data collection as part of the impact assessment analysis occurred on March 3rd and 4th, 2015. Based on the details test plans presented in Impact Assessment report, on the first day (March 3rd), priority eligible vehicles were driving on different approaches of the intersections. The second day (March 4) was designated for the base-case data collection using GPS units. Figure 61 shows time-space diagrams for 10 rounds of testing on the northbound through movement at the intersection of Daisy Mountain and Gavilan Peak from 1:30 pm – 5:00 pm. Trajectories for the trucks on Tuesday, March 3rd and Wednesday, March 4th are displayed on Figure 61 (a) and (b) accordingly. The diagrams are drawn based on the BSMs received at the intersection RSE. Missing points in the trajectory of the vehicles on both days are an indication of packet loss in the wireless communications.

On Tuesday, while MMITSS was running only 1 full stop occurred to the priority requesting truck once it encountered the queue and slowed down. On Wednesday, when the MMITSS priority applications were not running except Performance Observer (to calculate the travel time, delay, and etc.) the vehicle stopped 5 times.

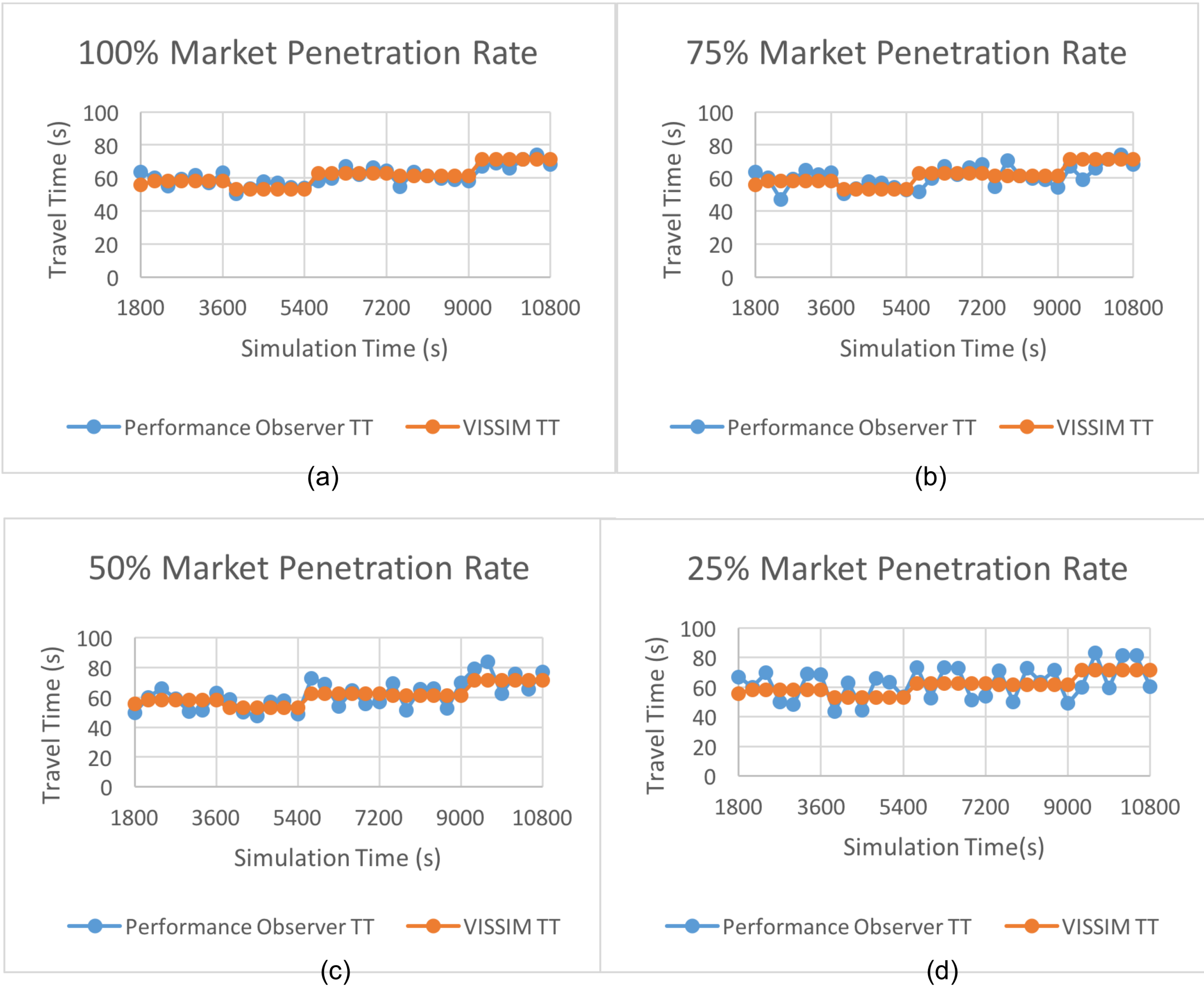
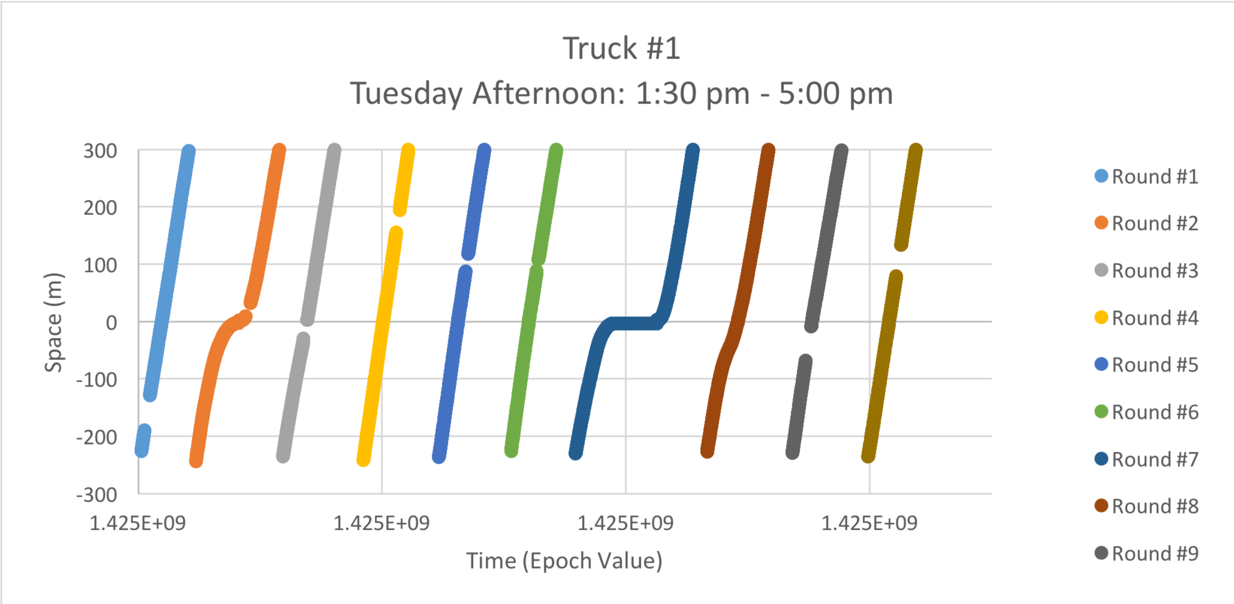
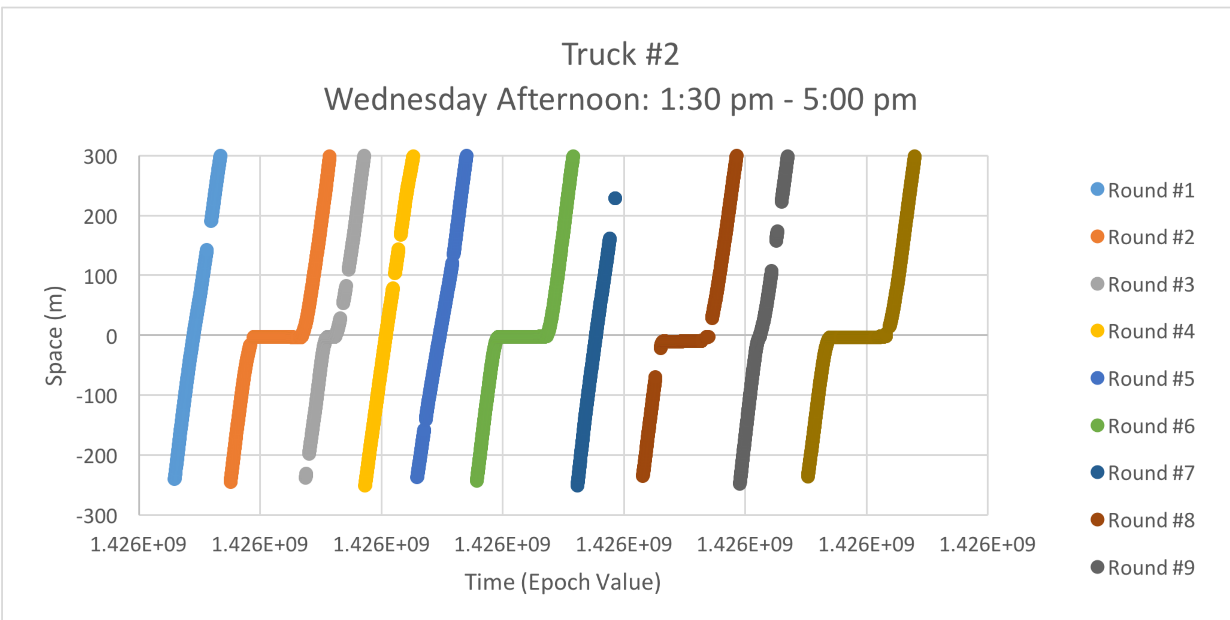


Figure 60. Travel Time results of Performance Observer vs. VISSIM for different Penetration Rates: (a) 100%, (b) 75%, (c) 50%, and (d) 25%



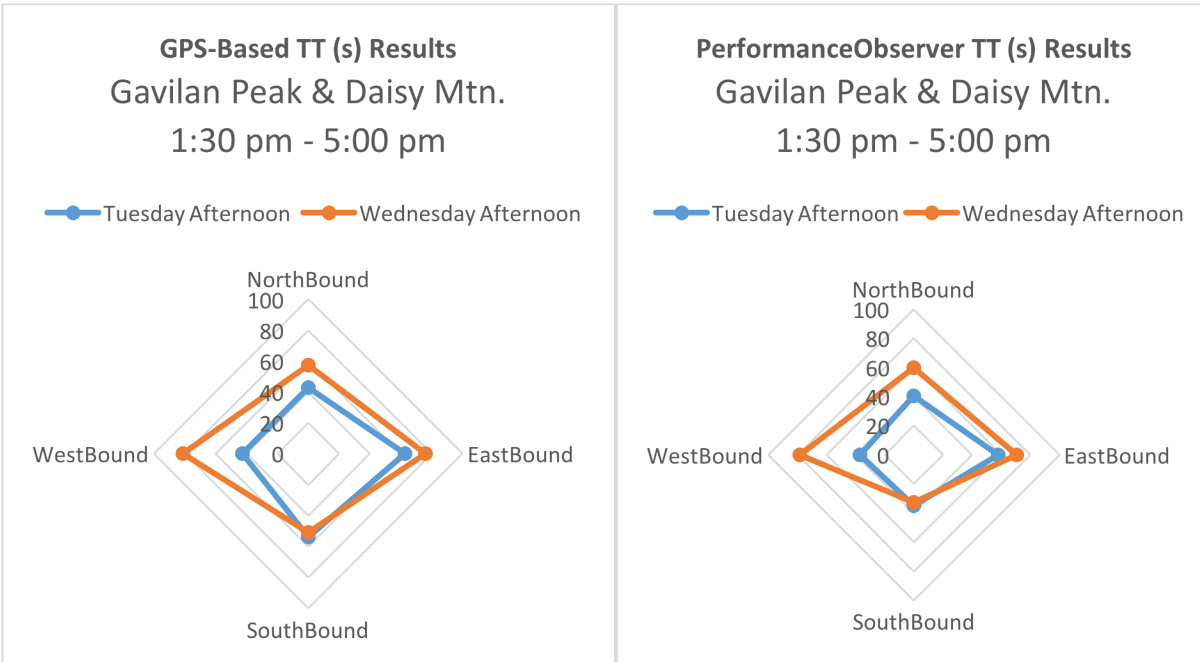
(a)



(b)

Figure 61. Time-Space Diagram of 10 rounds of data collection for trucks: (a) With MMITSS, and (b) Without MMITSS

Further analysis on travel time of the same set of field data categorized by movement is shown in Figure 62. The radar diagram on the left (Figure 62(a)) displays the average travel time calculated based on time-stamped GPS data logged on the OBE while the vehicle was transmitting BSMs. Figure 62(b), on the right, shows the average travel time estimated using the received BSMs in the RSE by the Performance Observer application. The purpose of this comparison is to show a correlation between the data calculated by an external entity (i.e. Impact Assessment group) and the data estimated by a MMITSS internal component (i.e. Performance Observer). Table 31 summarizes the statistics of the correlation analysis for the two days during the afternoon test runs and shows that the MMITSS Performance Observer is effective at estimating travel time.



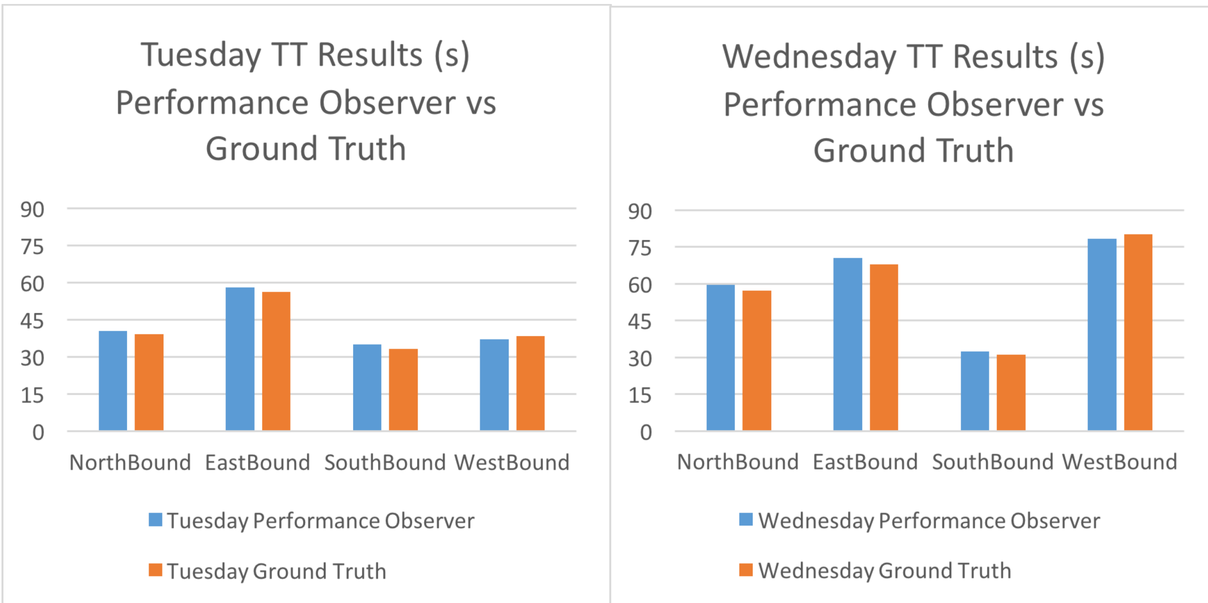
(a) (b)

Figure 62. Travel Time Analysis based on: (a) GPS data collected, and (b) Performance Observer data logs

Table 31. Correlation between MMITSS Observed and GPS-based Travel Time Data

	Tuesday Afternoon		Wednesday Afternoon	
	GPS-based TT (sec)	Performance Observer TT (sec)	GPS-based TT (sec)	Performance Observer TT (sec)
Northbound	43	40.44	57.60	59.66
Eastbound	62.56	57.90	75.89	70.62
Southbound	53.95	34.96	50.7	32.43
Westbound	42.66	37.14	81.15	78.26
	Correlation of Travel Times on Tuesday Afternoon	0.72	Correlation of Travel Times on Wednesday Afternoon	0.92
	Correlation of Travel Times between Performance Observer & GPS-based data on Tuesday and Wednesday			0.90

Accuracy of the travel time estimates is important factor in judging the ability of the Performance Observer to provide useful information and can be impacted by several factors. Accuracy assessment depends on the starting and ending travel time sections. These locations depend on where the vehicles are located when the BSMs are received by the RSE which can be variable based on packet loss and other factors. The vehicle GPS location extends beyond the RSE radio range, so only locations that align with the RSE points should be used. To address this source of variability, unified travel time segments are considered for the two data sets, and the results of estimated travel times obtained from Performance Observer application and ground truth values based on GPS data logged on OBEs are shown in Figure 63.



(a) (b)

Figure 63. Accuracy Analysis of Performance Observer compared to GPS-based Travel Time data for: (a) March 3rd, 2015, and (b) March 4th, 2015

Mean Absolute Percentage Error (MAPE) is calculated for the data sets shown in Figure 63, and the results of the accuracy investigation are presented in Table 32. Choosing the same travel time sections for the analysis has a direct impact on the results. Part of the error reported in the following table is caused by this factor.

Table 32. Mean Absolute Percentage Error between the Performance Observer Estimation and GPS-based Calculated Travel Time

	Tuesday Performance Observer TT (s)	Tuesday GPS-based TT (s)	*MAPE (%)*	Wednesday Performance Observer TT (s)	Wednesday GPS-based TT (s)	*MAPE (%)*
Northbound	40.45	39.2	***3.18***	59.66	57.39	***3.95***
Eastbound	57.90	56.47	***3.03***	70.62	68.01	***3.83***
Southbound	34.96	33.25	***5.15***	32.43	31.16	***4.08***
Westbound	37.14	38.26	***2.90***	78.26	80.21	***2.42***

6.1.4.3 DSRC Range and Geo Fence Area

The DSRC radio range is an important system performance measurement that impacts the distance (headway) of the vehicle information (BSM and SRM) available for the MMITSS control algorithms. This performance measure was analyzed at different points of time during the project, but the MMITSS Performance Observer is responsible for continuously capturing this data. In addition to being an important operational performance measures, this data is very helpful for calibration of the simulation models. Figure 64 shows two pictures of the RSE radio mounting locations. In Figure 64(a) the RSE is mounted on the side of the light pole (the radio shown in the upper left is part of a wireless network link to the I-17 and Daisy Mountain interchange). After capturing the radio range data, it was decided to remount the radios on the street lighting mast arm so that there would be no interference by having the antenna so close to the pole. Figure 64(b) shows the relocated mounting. Table 33 shows the improvements made in the range of the radio after remounting. The approaching range is the distance from which BSMs from vehicle approaching the intersection are received. The leaving range is the distance from which BSMs from vehicles leaving the intersection are no longer received, e.g. the last received location. Additional considerations need to me made to control the power based on the FCC CFR (Code of Federal Regulations) 47 Part 90 requirements.

(a)

(b)

Figure 64. Mounting Location of DSRC Radios. (a) Initial Mounting and (b) Remounted Position

The greater radio range can allow each MMITSS controlled intersection to have a better view of the traffic approaching the intersection. However, there is a significant issue which limits the capability of the MMITSS applications to utilize the data from the full range of the DSRC radios. There is limit on the size of a WAVE message that was experimentally determined to be 1 kb in size. This limitation requires that the quantity of data in the MAP message be limited. Since the MAP is composed of waypoints (in a local coordinate system) this requirement implies that the number of waypoints must be reduced. Waypoints are used to describe each lane on the approach (ingress) and leaving (egress) legs of an intersection. The number of waypoints required depends on the curvature of the roadway, location of critical geometric features such as turning pockets, and distance to be covered. In MMITSS, the extent of the MAP is used to create a geo-fence to limit the consideration of vehicles that are not in the roadway. Figure 65 illustrates the situation. The figure on the left shows the range from which BSM messages are received (yellow points in the lanes). The figure on the right shows the geo-fence area covered by the MAP. Clearly, the MAP is a significantly limiting factor in utilizing the full DSRC range. Experience in implementing the MAP showed that the number of lanes and roadway curvature and turning pockets required that the MAP range be severely limited. This impacts the collection of performance measures as well as the other MMITSS traffic control applications including I-SIG, TSP, EVP, and FSP. The MAP messages was developed using the 2009 version of the SAE J2735 standard. The 2016 version of the standard significantly redefined the MAP description and requires a data encoding method (UPER) that is much more efficient in message size compression. It is believed that they new standard will help address this limitation.

Table 33. DSRC Range Before and After Remounting the Radio

	Approaching Range (m)			**Leaving Range (m)**		
Movement	Before Remounting the Radio	After Remounting the Radio	***Improvement (%)***	Before Remounting the Radio	After Remounting the Radio	***Improvement (%)***
Northbound	318.19	623.35	***95.90***	175.55	380.79	***116.91***
Eastbound	536.58	861.97	***60.64***	496.07	753.69	***51.93***
Southbound	183.56	399.67	***117.73***	365.03	699.81	***91.71***
Westbound	527.21	799.86	***51.72***	536.23	874.22	***63.03***

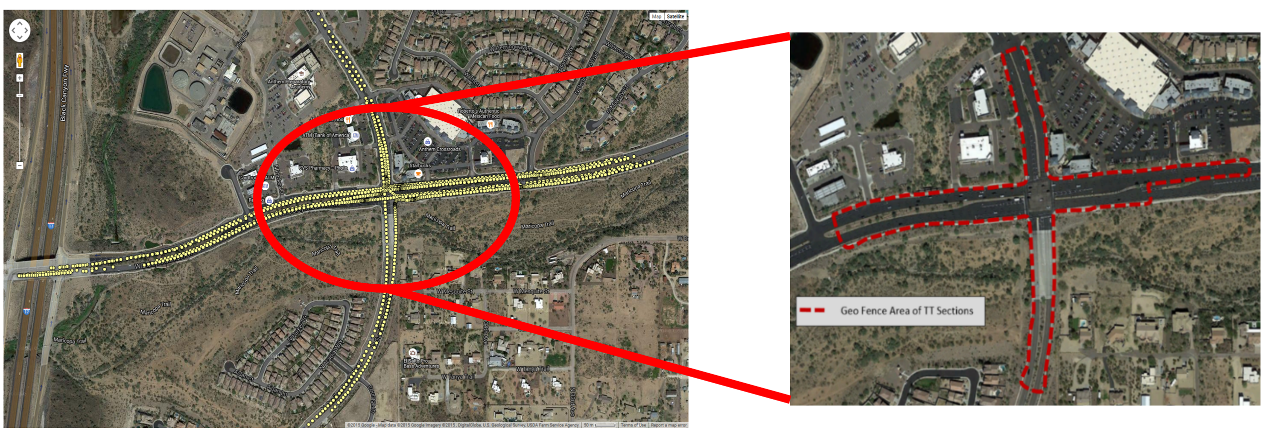


Figure 65. DSRC Range of the Radio based on Received BSMs vs. GeoFence Area based on MAP Lane Nodes

6.1.4.4 Number of Stops

This metric is characterized based on the speed of all the equipped vehicles at an intersection and is particularly interesting for Trucks. A threshold defined in the design document was set to be 2 miles per hour. A vehicle that has a speed less than this threshold value is considered to be fully stopped. The average number of truck stops per vehicle are estimated using the VISSIM simulation platform and a model of the intersection of Daisy Mountain and Gavilan Peak. In this simulation study, trucks composed 10% of the total vehicle fleet (demand) and some percentage (market penetration rate) of the trucks were considered to be connected vehicles.

The results are shown in Figure 66. The orange points represent the VISSIM measured number of stops for all (100%) of the trucks simulated. The blue points represent the MMITSS estimated number of stops using only BSM data that was communicated to the RSE. Different market penetration rates (of trucks) are considered. Data was collected over a simulation period of 3 hours with the first half an hour for warm-up. Each movement at the intersection is presented on one of the axis of the diagram. The results show the ability of MMITSS to accurately estimate the number of stops (truck) with the estimation error increasing as the market penetration decreases. This would be expected as the number of samples considered in the estimation is reduced.

Figure 66. Truck Average Number of Stops by movement for Market Penetration Rates of: (a) 100%, (b) 75%, (c) 50%, and (d) 25%

6.1.5 MMITSS Central System

The MMITSS Central System was developed using a series of web pages to allow operators to have access to both system configuration, including set up of the intersections and the N-Level Priority Policy Configuration, and visualization of the Performance Observations. The web server was designed to run on a Windows or Linux server in the same local network as the RSEs. Figure 67 shows the MMITSS Central System main web page that allows a user to access both configuration pages and performance reports. Figure 68 shows the MMITSS Central System performance report configuration page. This report allows the operator to select the performance measures, periods, measures, and chart styles to be selected. Figure 69 shows a sample performance report dashboard (with simulated data since insufficient field data is available to fully populate the database for the reports).

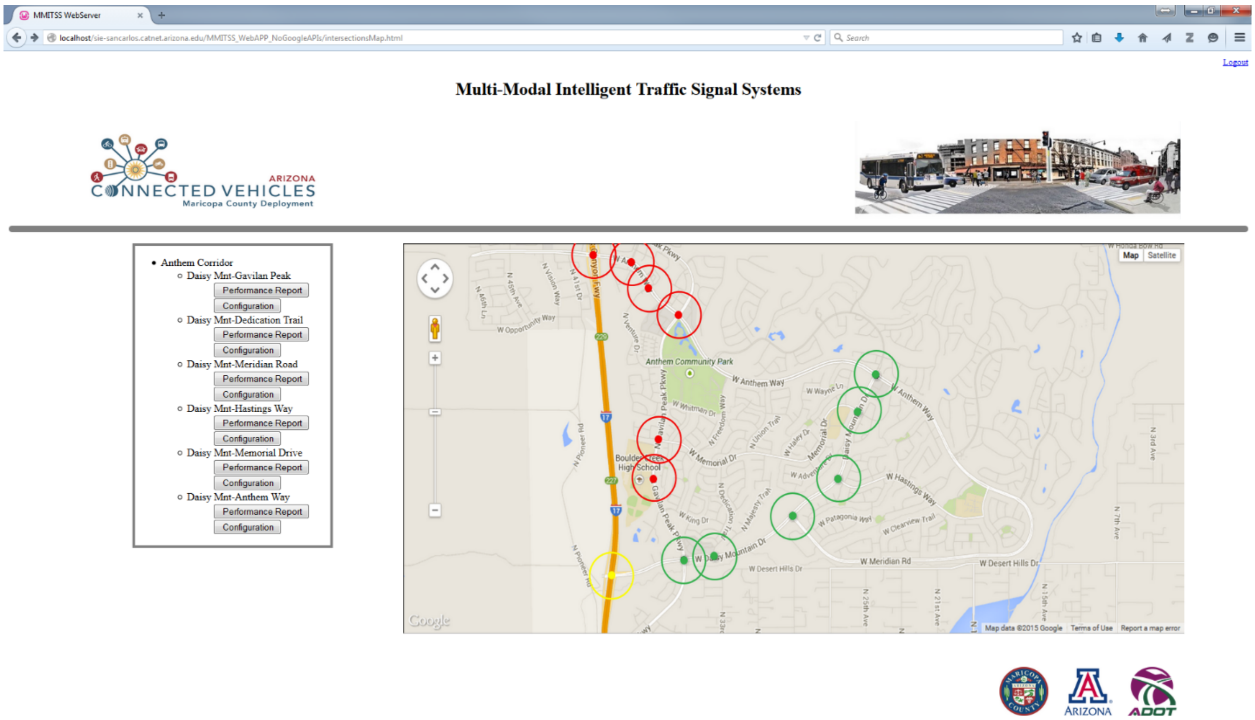


Figure 67. MMITSS Central System Main Web Page

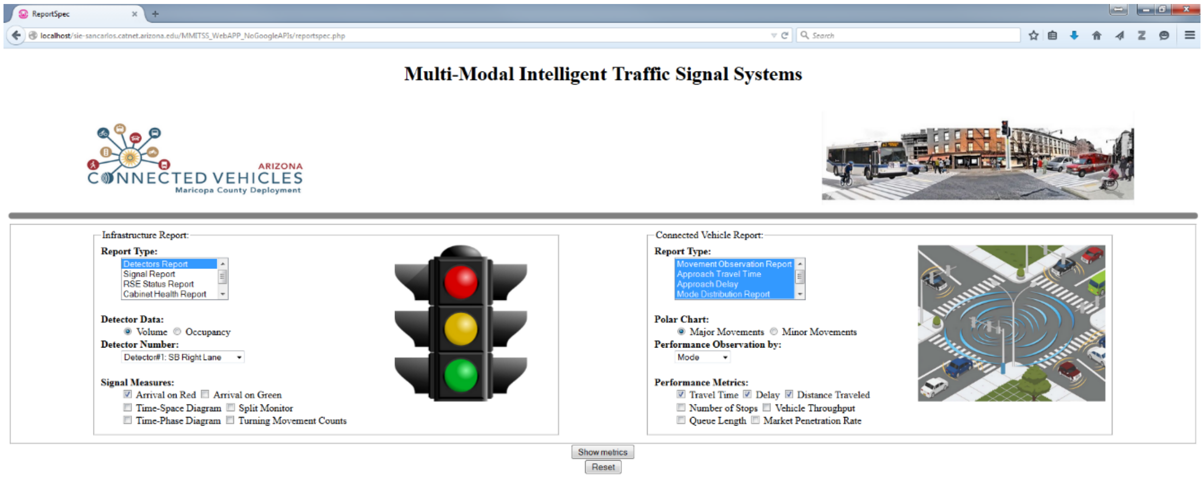


Figure 68. MMITSS Central System Performance Observer Report Setup

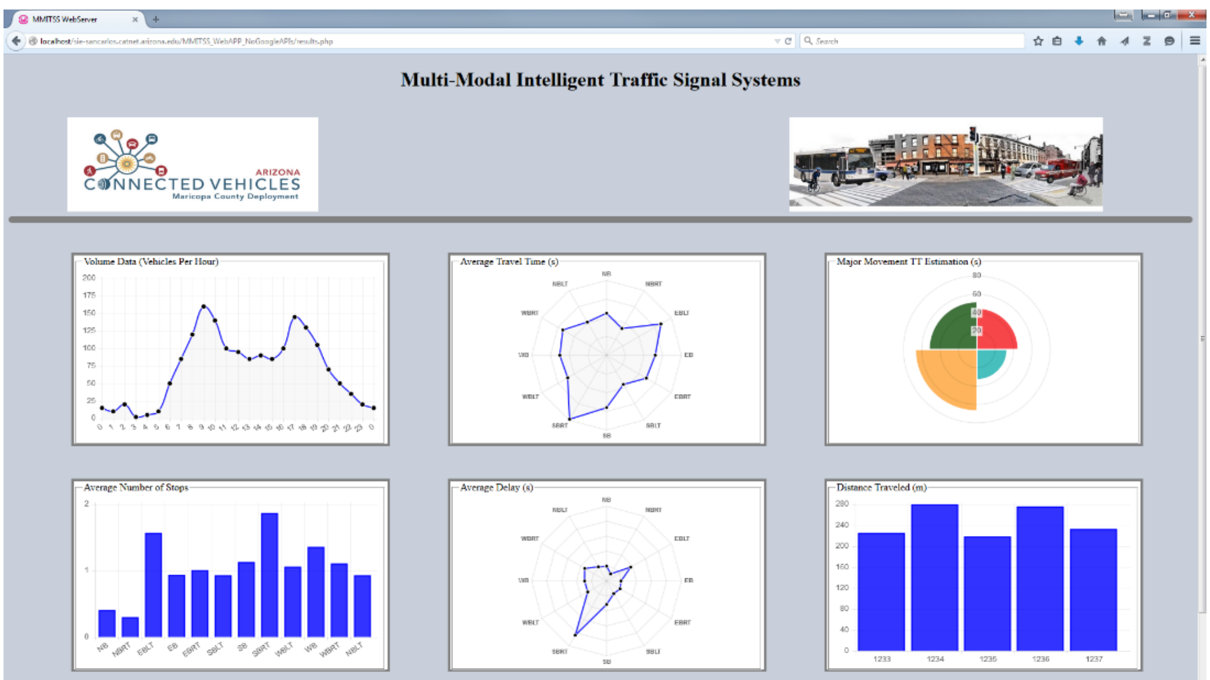


Figure 69. MMITSS Central System Performance Observation Dashboard

6.2 The California Simulation Platform and Experimental Results

This section presents the simulation result of MMITSS for the California testbed.

6.2.1 Description of Simulation Model

The simulation model was developed with VISSIM 7 and Econolite ASC/3 software controller module. Due to the lack of 2070 software controller module that is compatible with VISSIM 7, the Econolite ASC.3 software controller module was used for simulation. The Econolite software controller provides the standard NEMA dual-ring controllers with actuated-coordinated control logic which is the same as the 2070 controllers used in the California test bed. The El Camino Real test corridor runs approximately north-south and has three lanes in each direction, plus one or two protected left turn lanes at some signalized intersection. The VISSIM simulation network is shown in Figure 70. The simulation model is calibrated with field survey data collected by Caltrans during peak hours. Existing signal timing in the El Camino Real field intersections were incorporated into the ASC/3 controller module.

Figure 70. VISSIM Simulation Network

Econolite Transit Signal Priority (TSP) module for the ASC/3 Series controller was used for simulating signal priority control. TSP operation goes into effect automatically when a transit vehicle is detected. TSP operation takes place in several steps:

1) Check-in – Input to the controller from priority detection that senses the arrival of the transit vehicle and initiates the TSP functionality in the controller.
2) TSP Delay – Controller timing function that can be set to delay the beginning of TSP operation until a set interval after check-in.
3) Begin TSP Active Sequence – Controller begins using the TSP timing data for improved transit operation.
4) Check-out – Input to the controller from priority detection that determines the departure of the transit vehicle.
5) End TSP Sequence - The end of the TSP routine is based on check-out, timeout or lock released.

When the transit vehicle is detected, a single-pulse signal is applied to the TSP Check-In input on the ASC/3 controller. This input causes a TSP Check-In, followed by the delay timer, followed by the start of the TSP Sequence. When the transit vehicle clears the intersection a single-pulse is applied to the TSP Check-Out pin on the controller. This causes a TSP Check-Out, followed by the end of TSP Sequence. TSP operation that begins with a single-pulse check-in typically ends due to a single pulse check-out, as shown in Figure 71.

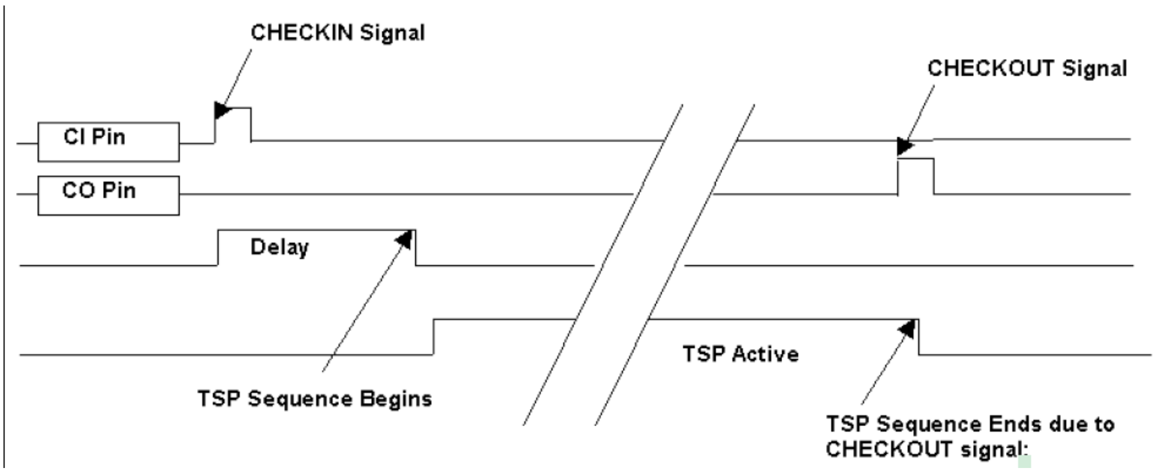


Figure 71. TSP Ending with a Single Pulse Check-Out

A Visual Basic program was developed to collect second-by-second CV trajectory data through VISSIM COM interface. For the purpose of extracting travel time between consecutive signalized intersections, a link associated with an intersection constitutes a unique segment, which includes the through, right-turning and left-turning intersection movements as shown in Figure 72. The section of the simulation corridor is divided into 132 segments in total.

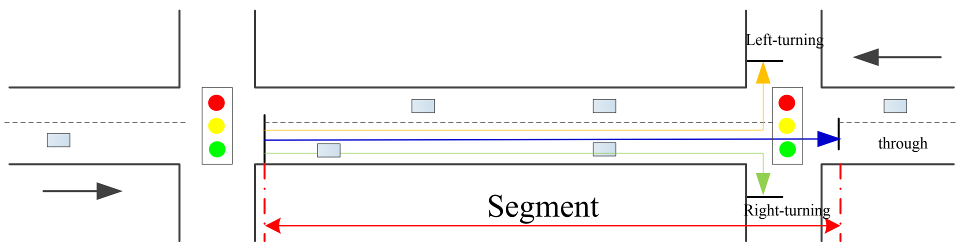


Figure 72. Definition of Link and Segment

An algorithm coded in Java was developed to locate the segment where the specified probe vehicle traverses as well as its entry time, departure time and the associated signal cycle clock times in this segment. Thus, the corresponding travel time of a probe vehicle is extracted. The resultant data sets contain a suite of information including vehicle ID, travel time, segment ID, the entry time and exit time in and out of the specific segment. The segment travel time is the mean travel time of probe vehicles over a 15-second interval. A time interval of 15 seconds is selected for analysis because it is a time span that is long enough for movements along a short link and the movements at intersections. Yet, it is short enough to capture the dynamic variations in travel time as a result of the signal timing.

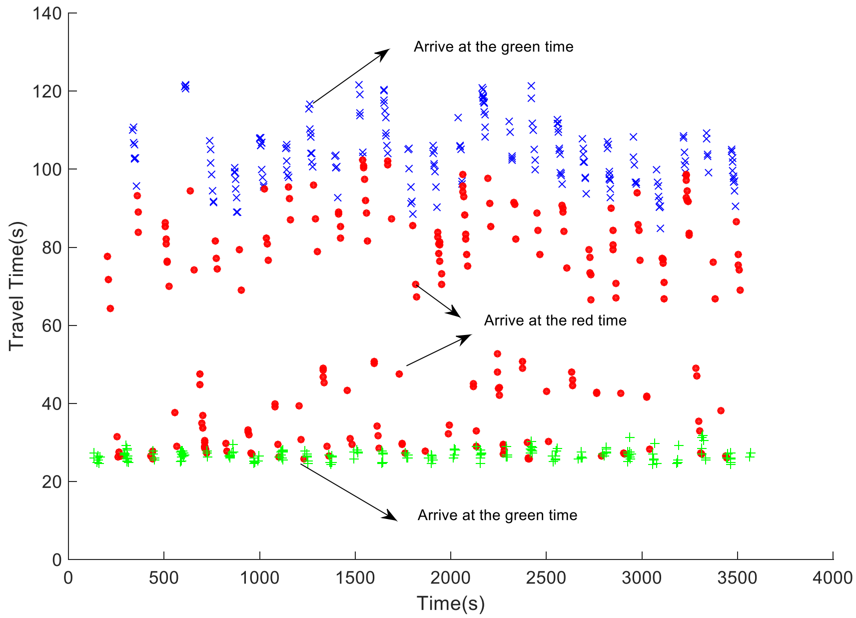


Figure 73. Segment Travel Time vs. Entry Time in Signal Cycle

Figure 73 shows a sample data set collected from the VISSIM simulation. Each point in Figure 73 represents an individual travel time of a vehicle corresponding to its entry time in the signal cycle. The travel time data for vehicles entering during the red phase and those during the green phase show very distinct patterns. It is straightforward to see that the travel time for a vehicle is short if the remaining green time is sufficient for the vehicle to pass through the segment, as indicated by the green dots located in the lower portion of Figure 73. For those entering in the green phase with insufficient time for them to pass through, the travel time becomes much longer as they are required to stop and wait for the next green phase as indicated by the blue dots in the upper portion of Figure 73. The travel times of vehicles entering during the red phase stop for the red signal and line up in a queue as shown by the red dots located between the green and blue ones in Figure 73.

6.2.2 Predicting of Segment Travel Time

Figure 73 clearly shows a strong correlations between the urban segment travel time and the vehicle's entry time the signal cycle. For the purpose of future expanding the MMITSS traffic and priority control to the section level, a model has been developed to predict the segment travel time based on vehicle's entry time in the signal cycle.

6.2.3 Critical Time Identification

For every vehicle, its travel time consists of a free flow travel time and an intersection delay. The free flow travel time is a function of the travel distance and the travel speed, while the intersection delay includes a signal delay and a queuing delay. Further, the signal delay can be decomposed into three components: D_d is the time needed for deceleration, D_W is the time spent waiting for the signal to turn green, and D_a is the additional time for a vehicle to accelerate up to the free flow speed. The queue delay D_q is the time that a vehicle waits in the discharging queue.

Generally speaking, all vehicles approaching an intersection will encounter one of the three following scenarios: (1) come a complete stop due to the red signal phase or the standing queue; (2) decelerate to a certain speed to join the moving queue; (3) pass through the intersection without any deceleration. The key question is to determine which scenario that the vehicle will encounter when it enters the specified segment at time t_{in}.

As illustrated in Figure 74, if a vehicle enters the segment with the remaining green time less than the required minimum travel time, it should stop to wait for the next green time. If the starting time of the next green phase is set as zero in the horizontal axis as show in Figure 74, the first critical time t_0 can be calculated with the following equation:

$$t_0 = -r - \frac{L}{u_f} \tag{1}$$

where r is the duration of the red time; L is the link length (m) ; u_f is the free flow speed.

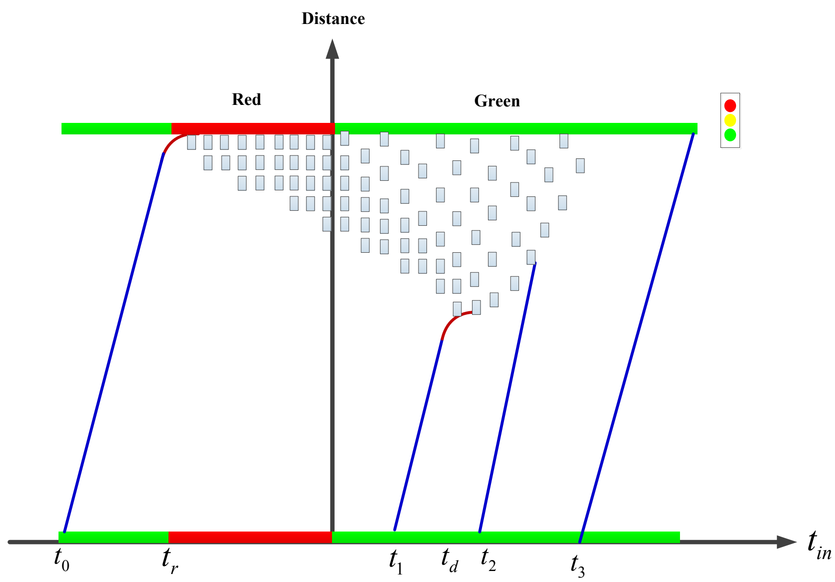


Figure 74. Critical Time Points Identification

With this definition, a subject vehicle entering the link within a time period later than t_0 should decelerate to stop and wait at the intersection. However, there exists another critical time point, after which the subject vehicle entering the segment will not need to come to a complete stop but only need to decelerate to a certain lower speed u ($u \neq 0$). If a subject vehicle reaches the end of queue at a moment when the queue is already moving or in other words the discharge shockwave of the queue has already propagated backward to the rear end of the queue. In this case, the subject vehicle will decelerate to join the moving queue but does not need to come to a stop. Figure 75 further depicts the queue formulation and discharge processes. During the red phase the vehicles are accumulated from the stop bar. Then, when the signal turns green, the first vehicle in the queue will start to move and the discharge shockwave begin to propagate backward. Note that, the queue length may still continue to increase if more vehicles arrive in the discharging process, until the vehicle at the rear end of the queue begins to move. t_d is the time when the queue reaches the maximum length, which can be calculated based on the geometry of triangles in Figure 75:

$$w \cdot t_d = u_w \cdot (t_d + r) \quad \Rightarrow t_d = \frac{u_w \cdot r}{w - u_w} \tag{2}$$

where u_w is the queueing growth shock wave speed and w the discharge shock wave speed.

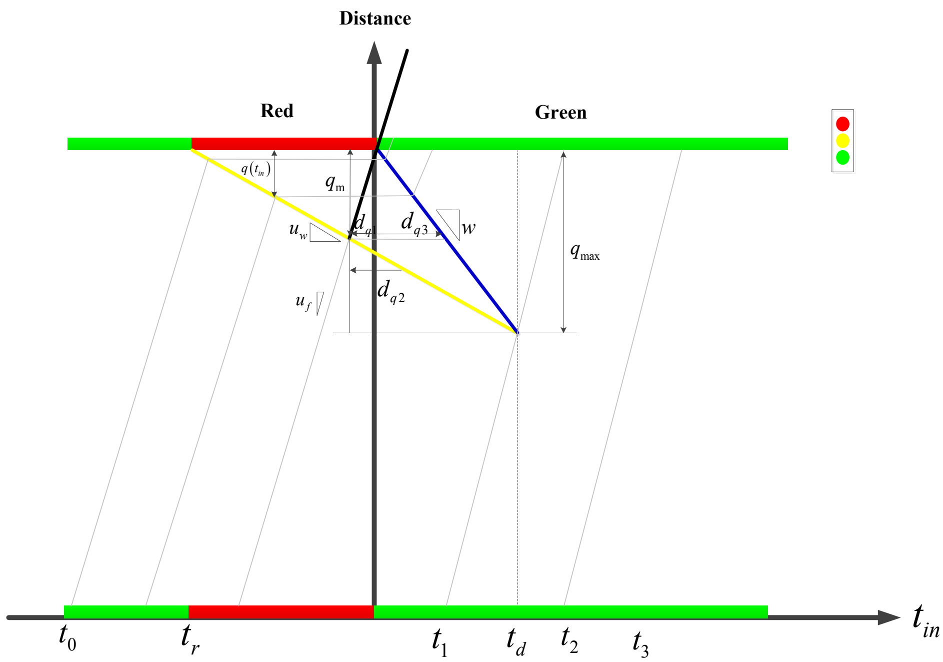


Figure 75. Queue Delay at Intersection

Therefore, if the vehicle arrives at the end of the queue at time t_d, it will be the last vehicle that will come to a stop. T_1 represents the required time before this vehicle stops to join the queue, and it can be obtained with the follow equation:

$$(T_1 - \frac{u_f}{2\gamma_d})u_f = L - \left(q_{\max}\text{-}l_s\right) \Rightarrow T_1 = \frac{L - q_{\max} + l_s}{u_f} + \frac{u_f}{2\gamma_d} \tag{3}$$

Where γ_d the vehicle deceleration rate, $q_{\max}$ is the maximum queue length in the current cycle; l_s is the effective length of the vehicle. With the calculation of T_1, the entry time of the last stopping vehicle can be then estimated with the following equation, which yields the second critical time point. Specifically, the vehicles with their entry time between t_0 and t_1 will need to completely stop as they approach the intersection.

$$t_1 = t_d - T_1 = \frac{u_w \cdot r}{w - u_w} - \frac{L - q_{\max} + l_s}{u_f} - \frac{u_f}{2\gamma_d} \tag{4}$$

The next step is to identify the third critical time point, after which the vehicle will not need to decelerate. Continuing with the analysis above, if the last vehicle in the queue has accelerated from 0 to the free flow speed, the newly arrived vehicles approaching to the queue needn't decelerate. So the first vehicle that passes the intersection without deceleration should arrive at the end of the queue when the last vehicle has accelerated to the free flow speed. Similar to the calculation of t_1 above, the time period before the newly arrived vehicle joining the moving queen is assumed to be T_2 and can be obtained with the following equation:

$$q_{\max} - \frac{u^2{}_f}{2*\gamma_a} = L - u_f * T_2 \Rightarrow T_2 = \frac{L - q_{\max}}{u_f} + \frac{u_f}{2*\gamma_a} \tag{5}$$

where γ_a is the acceleration rate; $\frac{u^2{}_f}{2*\gamma_a}$ is the distance that the queue moves forward. Accordingly, this vehicle arrives at the end of the queue at the time $t_d + \frac{u_f}{\gamma_a}$, which is the time for the queue to reach the maximum length then accelerate to the free flow speed, and t_2, which is the time point when the vehicle entering the segment will not need to decelerate , can be expressed as follows:

$$t_2 = t_d + \frac{u_f}{\gamma_a} - T = \frac{u_w \cdot r}{w - u_w} + \frac{u_f}{2*\gamma_a} - \frac{L - q_{\max}}{u_f} \tag{6}$$

Lastly, for the vehicle entering the segment with the remaining green time insufficient to pass through within this green phase, it will need to stop at the intersection. Thus, the third critical time point can be obtained with the following equation:

$$t_3 = g - \frac{L}{u_f} \tag{7}$$

In the above derivations, several critical time points have been identified to help construct the prediction model. In summary,

1) The vehicles entering the segment during the first time interval t_0 and t_1, the vehicle should come to a complete stop before the intersection.
2) If vehicles enter the segment between time t_1 and t_2, they will need to decelerate to a certain speed between the free flow speed and zero
3) The time interval between t_2 and t_3 allows the vehicle pass through the intersection without any delay.

6.2.4 Delay Prediction

For the vehicles entering the segment between time t_0 and t_1, they will experience the deceleration delay D_d and acceleration delay D_a. These delays are the additional times to account for the vehicle decelerating from the free-flow speed to a stop and accelerating from speed zero to free flow speed, respectively. These two delays can be estimated with the following equations:

$$D_d = \frac{u_f^2}{2\gamma_d} - \frac{u_f}{\gamma_d} = \frac{u_f}{2\gamma_d} \tag{8}$$

$$D_a = \frac{u_f^2}{2\gamma_a} - \frac{u_f}{\gamma_a} = \frac{u_f}{2\gamma_a} \tag{9}$$

As for the waiting time of a vehicle before the stop bar at the intersection, it depends on the time when the vehicle reaches the queue. The vehicles arriving at the queue when the phase is red will experience a waiting time. Actually, the waiting delay is the remaining red time at the instant when the vehicle reaches the stop line or the end of the queue. For instance, the first vehicle that stops at the beginning of the red phase has a waiting delay equal to the length of the red phase, r. Therefore, for the calculation of the waiting delay, it is important to predict the time instant when the vehicle reaches the queue, t_q. Assuming that the queue increases from the starting time of the red phase with the shockwave speed of u_w as shown in Figure 75, the queue length is $\left(t_{in} + r + T_3\right) \cdot u_w$, when the subject vehicle arrives at the rear end of the queue, where T_3 is the time before the subject vehicle arriving at the queue.. The traveling time before the subject vehicle arrives at the queue and the queue length encountered by the subject vehicle with its entry time t_{in} can be calculated with the following equations:

$$\begin{aligned} &\left(t_{in} + r + T_3\right) \cdot u_w = L - u_f T_3 + \frac{u_f^2}{2\gamma_d} \\ &\Rightarrow T_3 = \frac{L + \frac{u_f^2}{2\gamma_d} - \left(t_{in} + r\right) u_w}{u_w + u_f} \\ &\Rightarrow q\left(t_{in}\right) = \left(\frac{L + \frac{u_f^2}{2\gamma_d} + \left(t_{in} + r\right) u_f}{u_w + u_f} \right) u_w \end{aligned} \tag{10}$$

where $q\left(t_{in}\right)$ is the queue length that the vehicle entering the segment at time t_{in} will encounter.

Then, t_q can be calculated correspondingly:

$$t_q = t_{in} + T_3 = \frac{L + \frac{u_f^2}{2\gamma_d} - u_w r + u_f t_{in}}{u_w + u_f} \tag{11}$$

where t_q is the time when the vehicle arrives at the end of the queue.

Based on the above, if $t_q < 0$ D_w=-t_q ; else D_w=0 .

As shown in Figure 75, $q_{\max}$ represents the maximum queue length in the current signal cycle and q_m is the distance at which the queueing delay is at a maximum. According to the geometric relations in Figure 75, $q_{\max}$ and q_m can be estimated with the following equations:

$$q_{\max} = w \cdot t_d = \frac{w \cdot u_w \cdot r}{w - u_w} \tag{12}$$

$$q_m = r \cdot \frac{u_f \cdot w}{u_f + u_w} \tag{13}$$

The queueing delay d_q of a vehicle entering the link at time t_{in} can be estimated by considering the following three types of delays. The total queueing delay of a vehicle arriving at a queue is the sum of the delays given by the following formulae:

$$d_{q1} = \frac{\min\left(q(t_{in}), q_{\mathrm{m}}\right)}{u_f} \tag{14}$$

$$d_{q2} = \frac{-\left(\min\ \left(\max\left(q\left(t_{in}\right), q_m\right), q_{\max}\right) - q_m\right)}{u_w} \tag{15}$$

$$d_{q3} = \frac{\min(q(t_{in}), q_{\max})}{w} \tag{16}$$

$$D_q = d_{q1} + d_{q2} + d_{q3} \tag{17}$$

Vehicles entering the link between time t_1 and t_2 will decelerate to a certain speed u when approaching the intersection. If the entry time of a vehicle is close to t_1, the vehicle will join the queue with a speed close to 0; on the other hand, vehicles entering the link at a time near t_2 will join the queue with a speed close to the free flow speed. As noted, the speed of a vehicle joining the queue is dependent on its entry time. For simplicity, it is assumed that u varies from 0 to u_f as a linear function of the arriving time between time t_1 and t_2 with a deceleration rate of γ_d , as shown with the following equation:

$$u = \frac{t_{in} - t_1}{t_2 - t_1} \cdot u_f \tag{18}$$

The corresponding deceleration delay equals to

$$D_{s1} = \frac{u_f - u}{\gamma_d} - \frac{u_f^2 - u^2}{2\gamma_d \cdot u_f} \tag{19}$$

Substituting the equation (18) into (19), then D_{s1} is expressed as

$$D_{s1} = \frac{u_f \left(t_2 - t_{in}\right)^2}{2\gamma_d (t_2 - t_1)^2} \tag{20}$$

Similarly, a vehicle will accelerate from u to u_f with an acceleration rate of γ_a and its acceleration delay is

$$D_{s1} = \frac{u_f \left(t_2 - t_{in}\right)^2}{2\gamma_a (t_2 - t_1)^2} \tag{21}$$

A vehicle entering the segment during the time interval t_1 and t_2 will have no waiting delay but have a queueing delay since the signal phase is changing to green and the queue is discharging when they arrives at the intersection. For vehicles entering the link after t_2 but before t_3, they will not experience any delay.

Based on the analysis above, the delay for the vehicle with entry time t_{in} can be summarized with Eqs. (22) - (25). Thus, the predicted travel time $TT\left(t_{in}\right)$ for vehicle entering the link at time t_{in} is the sum of the free traveling time and the predicted delay, as presented in Equation (26).

$$D_d = \begin{cases} \dfrac{u_f}{2\gamma_d} & t_0 < t_{in} \le t_1 \\ \dfrac{u_f \left(t_2 - t_{in}\right)^2}{2\gamma_d (t_2 - t_1)^2} & t_1 < t_{in} < t_2 \\ 0 & t_2 \le t_{in} \le t_3 \end{cases} \tag{22}$$

$$D_w = \begin{cases} \dfrac{L + \dfrac{u_f^2}{2\gamma_d} - u_w r + u_f t_{in}}{u_w + u_f} \; or \; 0 & t_0 < t_{in} \le t_1 \\ 0 & t_1 < t_{in} < t_2 \\ 0 & t_2 \le t_{in} \le t_3 \end{cases} \tag{23}$$

$$D_a = \begin{cases} \dfrac{u_f}{2\gamma_d} & t_0 < t_{in} \le t_1 \\ \dfrac{u_f \left(t_2 - t_{in}\right)^2}{2\gamma_a (t_2 - t_1)^2} & t_1 < t_{in} < t_2 \\ 0 & t_2 \le t_{in} \le t_3 \end{cases} \tag{24}$$

$$D_q = \begin{cases} \dfrac{\min\left(q(t), q_{\mathrm{m}}\right)}{u_f} + \dfrac{-\left(\min\ \left(\max\left(q\left(t\right), q_m\right), q_{\max}\right) - q_m\right)}{u_w} + \dfrac{\min(q(t), q_{\max})}{w} & t_0 < t_{in} \le t_1 \\ 0 & t_1 < t_{in} < t_2 \\ 0 & t_2 \le t_{in} \le t_3 \end{cases} \tag{25}$$

$$TT\left(t_{in}\right) = \frac{L_s}{u_f} + D_d + D_w + D_a + D_q \tag{26}$$

The following describes the event-based approach to estimate the real-time queue length. The input to the method is event data that consist of signal timing and probe trajectory data. Corresponding procedures are developed to handle each type of event and update the system state. When the estimation method receives a request to estimate the real-time queue length, the queue estimation procedure conducts the estimation on the basis of the current system state.

The signal timing data consist of (a) the time when the traffic light turns green and (b) the time when the traffic light turns red. The probe trajectory data comprise the vehicle identification, time, location, and speed. The authors designed a data structure, called the system state, which represents each probe vehicle. The system state includes the vehicle identification and a sequence of time, location, and speed data. In addition to the data directly received from probe vehicles, the system state also includes estimated data, such as (a) the time when the vehicle stops, (b) the distance to the stop bar if the vehicle stops, (c) the time when the vehicle starts to depart after the traffic light turns green, and (d) the time when the vehicle passes the stop bar after the traffic light is green. The system state is updated when an event is received. The real-time queue length is estimated on the basis of the current system state.

A data fusion algorithm is proposed to improve the estimation performance by fusing probe trajectory data and loop detector data. The outline of the data fusion approach is as follows. First, queue length is estimated by using the Kalman filtering technique on the probe vehicle data and loop detector data separately. Second, the Kalman filtering formulations are solved, and the weights are set as the reciprocal of the covariance in the Kalman filtering. Finally, the weighted combination of the queue length from two data sources is determined as the estimated queue length.

6.2.5 Shockwave speed estimation

Based on the descriptions in the previous section, a model for real-time prediction of travel time on a signalized arterial can now be developed by utilizing the parameters embedded in those equations above, such as signal phase and timing, shockwave speeds, etc. One critical element in this model is the ability to predict the queue length at an intersection encountered by the subject vehicle when it enters the link at time t_{in}. As shown in Figure 75, the temporal and spatial accumulation and discharge of the queue are determined by the shockwave speeds. The traditional kinematic wave theory will be adopted to derive the shock wave speeds. As indicated in Figure 76, the accumulating and discharging shock wave speeds are determined by Eqs. (27) and (28). By applying Eq.(27) and Eq.(28), the shock wave speeds can be obtained with the assumed arrival flow rate (Q_a) saturation flow rate (Q_m), saturation density (K_m), and jam density (K_j). These variables can be estimated by using the flow and space-mean speed based on a single advance loop detector.

$$u_w = \frac{Q_a - 0}{K_a - K_j} \tag{27}$$

$$w = \frac{Q_m - 0}{K_m - K_j} \tag{28}$$

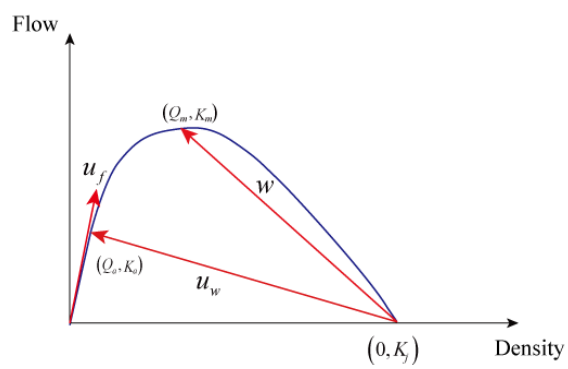


Figure 76. Shockwave Speed in Fundamental Diagram

A probe vehicle reports its trajectory second by second. As indicated in Figure 77(a) and (b), the reported time and distance to the stop bar when two separate probe vehicles stop in a queue can be used to calculate the accumulation shock wave speed. Similarly, as shown in Figure 77(c) and (d) the reported time and position when two probe vehicles start to move in the queue can be used for the calculation of discharge shock speed. By applying the following equations (Eq.(29) and Eq.(30)), the two shockwave speeds can be estimated:

$$u_w = \frac{d_2 - d_1}{t_2 - t_1} \tag{29}$$

$$w = \frac{d_2 - d_1}{t_2^{'} - t_1^{'}} \tag{30}$$

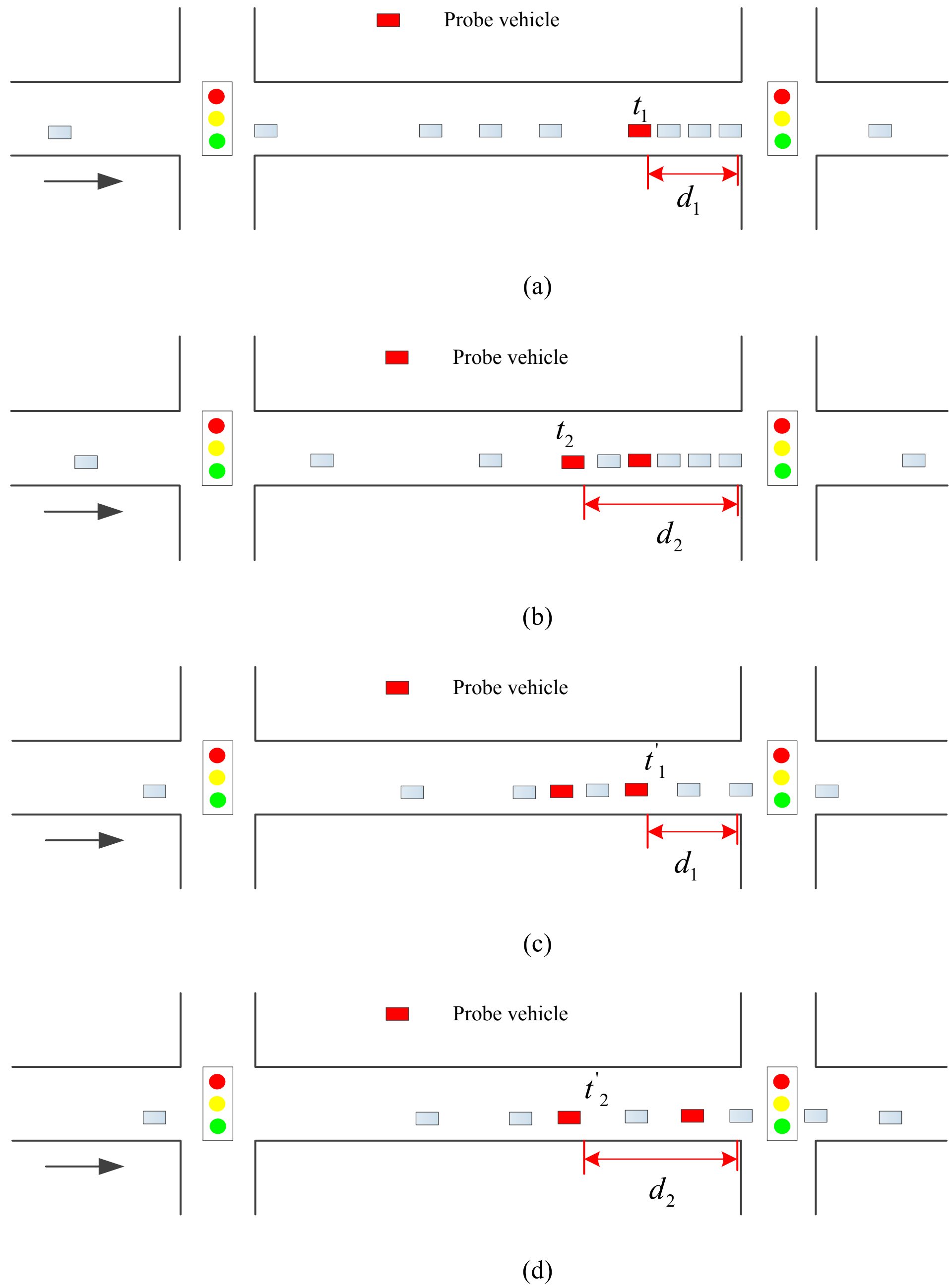


Figure 77. Probe Vehicle Data for Shockwave Speed Estimation

For real-time applications, we will use the instantaneous values of $u_w(t_{in})$ and $w(t_{in})$ to represent the shockwave speeds at time t_{in} and calculate the critical time points and the associated delay instead of using previously calculated values of u_w and w. $u_w(t_{in})$ and $w(t_{in})$ also need to be estimated since the shockwave speed data are not available at every moment. More specifically, $u_w(t_{in})$ is always unavailable for vehicles entering the link at a time near t_0. Meanwhile, $w(t_{in})$ can be obtained when the queue begins to discharge. To amend this point, we use the following rules to estimate $u_w(t_{in})$ and $w(t_{in})$:

1) If $u_w(t_{in})$ and $w(t_{in})$ at the recent time step is unavailable, and there are no available u_w and w data within the current cycle, $u_w(t_{in})$ and $w(t_{in})$ are assumed to be equal to the average shockwave speeds in the previous cycle.
2) If $u_w(t_{in})$ and $w(t_{in})$ at the recent time step is unavailable, but there are accessible u_w and w data within the current cycle, $u_w(t_{in})$ and $w(t_{in})$ are assumed to be equal to the shock wave speeds at a time step nearest t_{in} .

Following these rules, it becomes feasible to estimate $u_w(t_{in})$ and $w(t_{in})$ with the aforementioned rules. $u_w(t_{in})$ and $w(t_{in})$ are usually not accessible at the beginning of the red phase, but according to the equations for delay estimation, the delay of the vehicles arriving at this time period are not sensitive to the shockwave speeds since their queueing delay is small while the waiting delay represents a large portion of the total delay. Therefore, the estimate error is relatively small and acceptable as will be shown in a later section.

6.2.6 Implementation procedure

Figure 78 presents a flow chart to explain the implementation procedure for the travel time prediction algorithm. To implement this algorithm, the first step is to check whether the shockwave speed data is available at a recent time step. If the shockwave speed is unavailable, the historical shockwave speed is used to estimate the critical time points. Otherwise, use the recent shockwave speed data to estimate the critical time points. Then, the vehicle entry time t_{in} is compared to the critical time points to identify which time interval it is located in. With the determination of the specified time interval, the vehicle travel time can be predicted with the corresponding formulae described above.

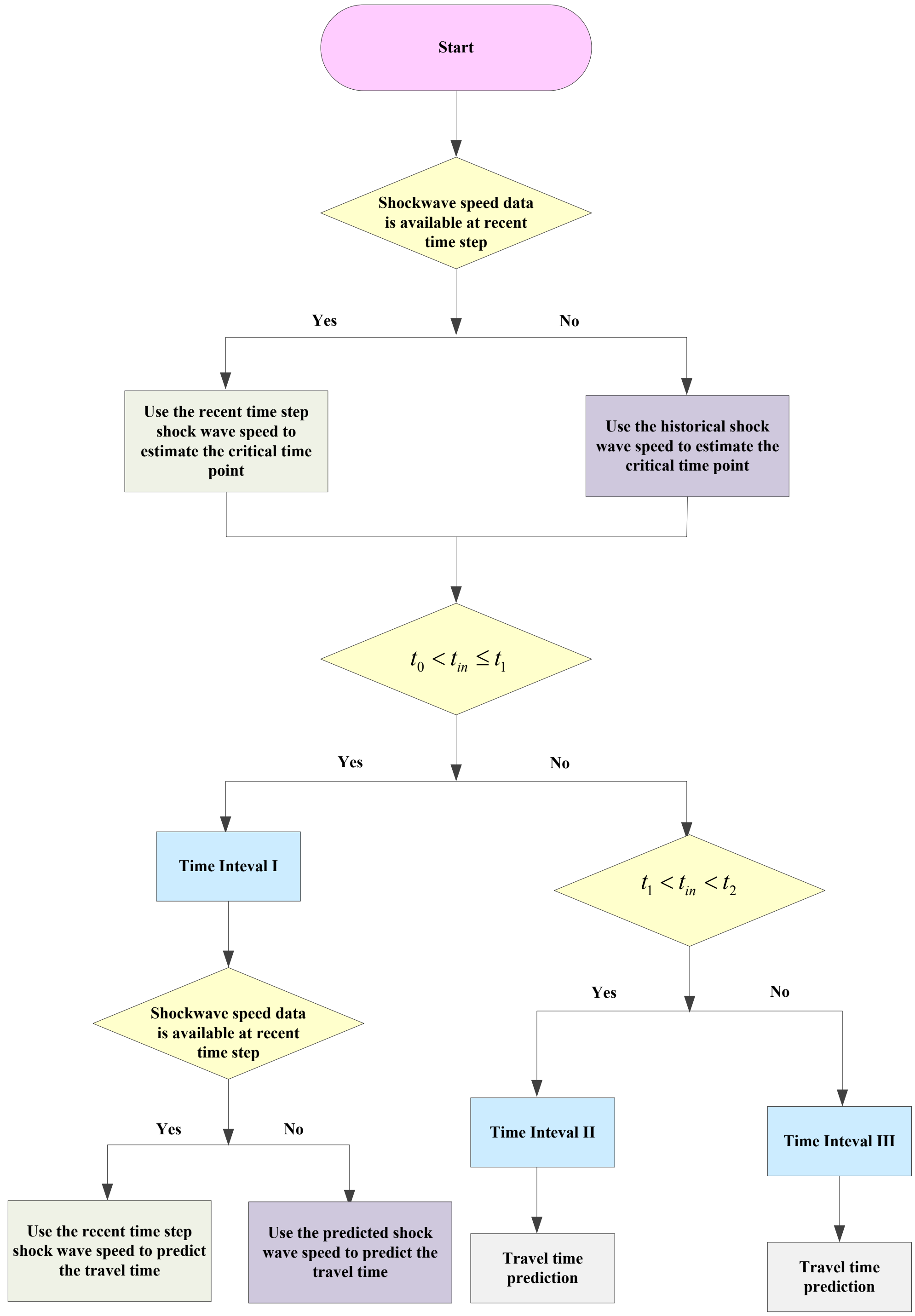


Figure 78. Prediction Model Implementation Procedure

6.2.7 Simulation Results

Figure 79 and Figure 80 show the comparison of estimated queue length (distance in meters to the stop-bar), difference in number of stops and difference in travel time (in seconds) of 'with' and 'without' signal priority when the penetration rate is 100%. The simulation results clearly show that in terms of queue length, number of stops and average travel time, the network with TSP clearly outperforms the network without TSP.

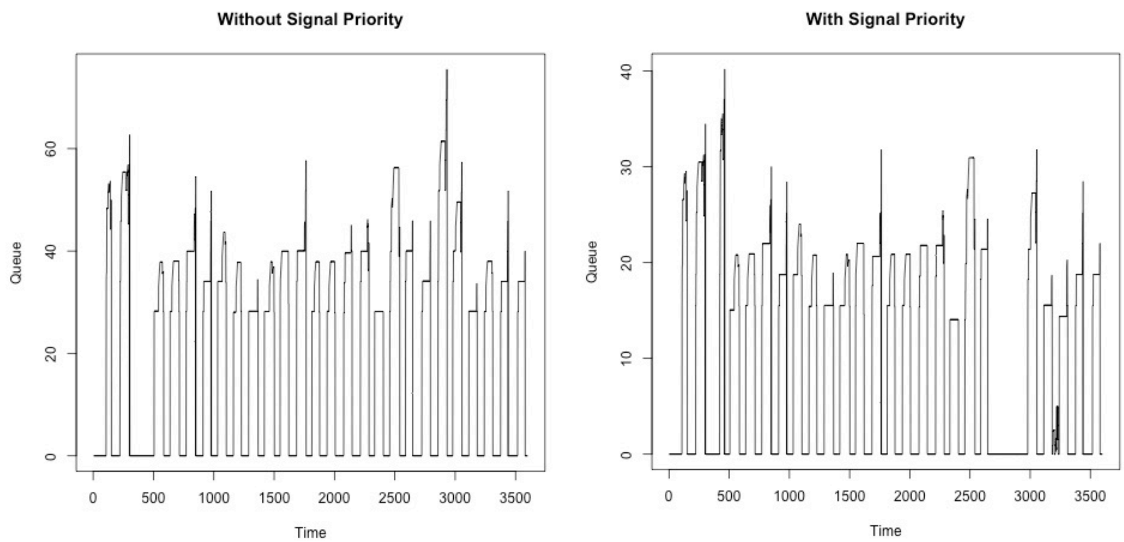


Figure 79. Queue Length Comparison of with and without Signal Priority

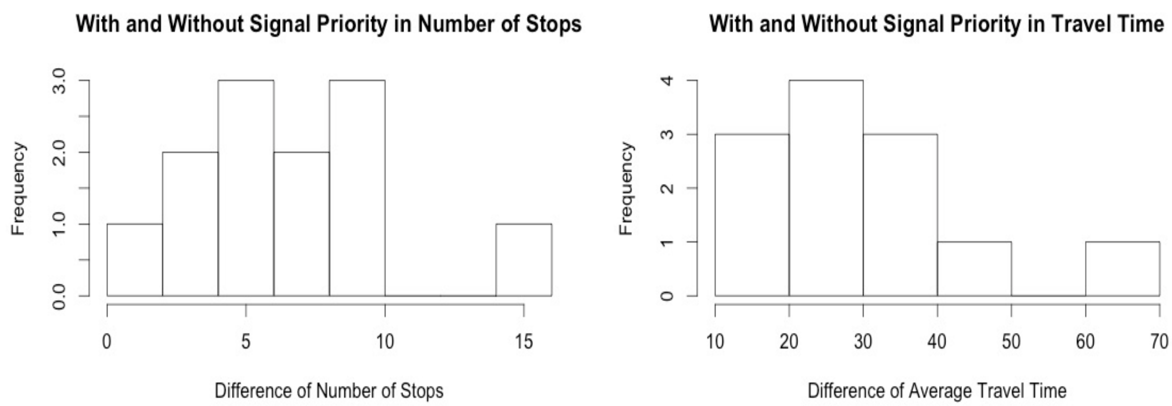


Figure 80. Comparisons of Travel Time and Number of Stops

Figure 81 presents the vehicle link travel times (in seconds) predicted by the proposed model and those obtained from the VISSIM simulation for six consecutive signal cycles. In the figure, each point represents the individual segment travel time of a vehicle that enters the link at its corresponding time of entry, t_{in}. These figures show that there is a close relationship between the link travel time and the entry time. The proposed model predictions are able to correctly track the travel times of various vehicles entering in different periods.

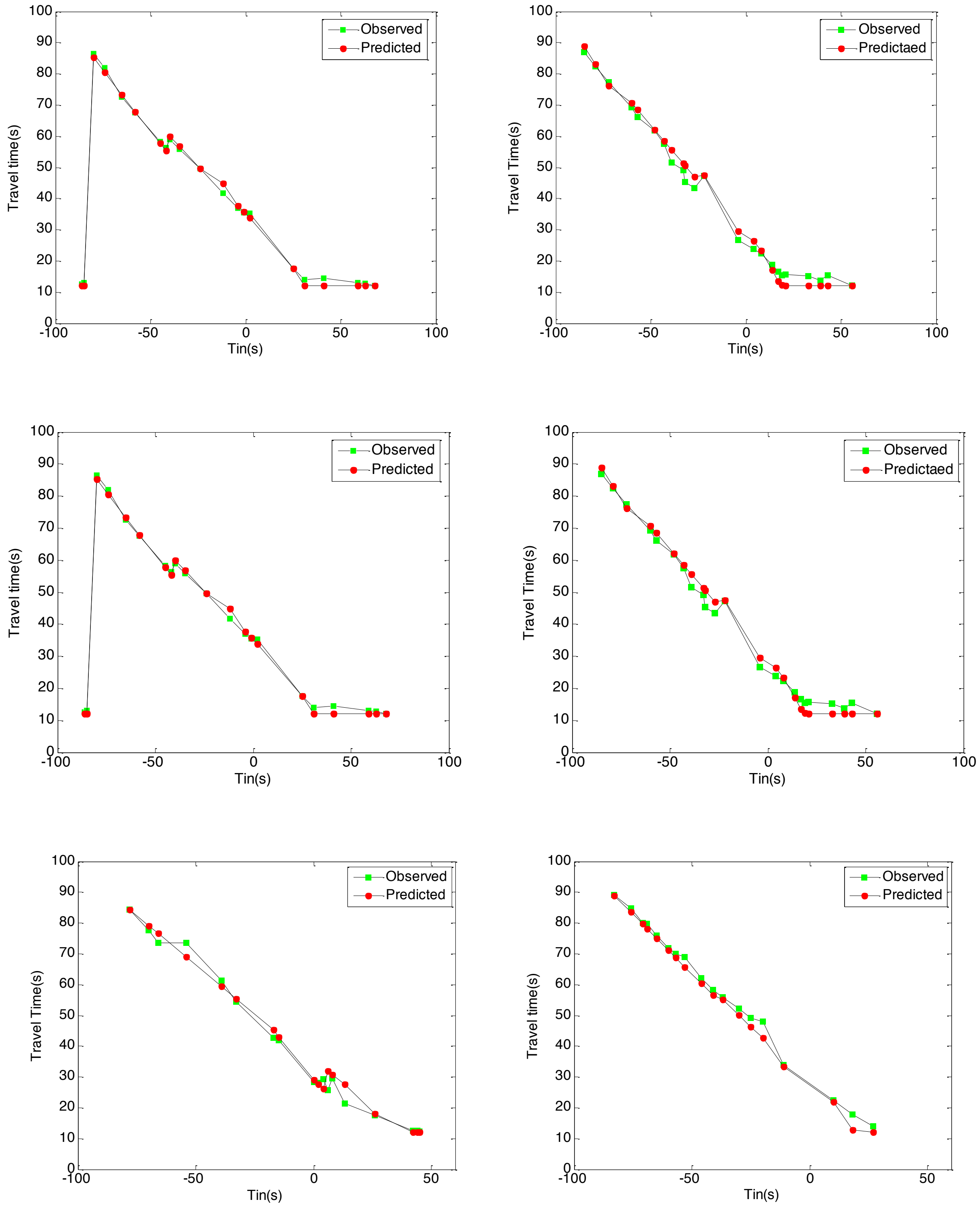


Figure 81. Observed vs. Predicted Travel Time corresponding to Entry Time in Signal Cycle

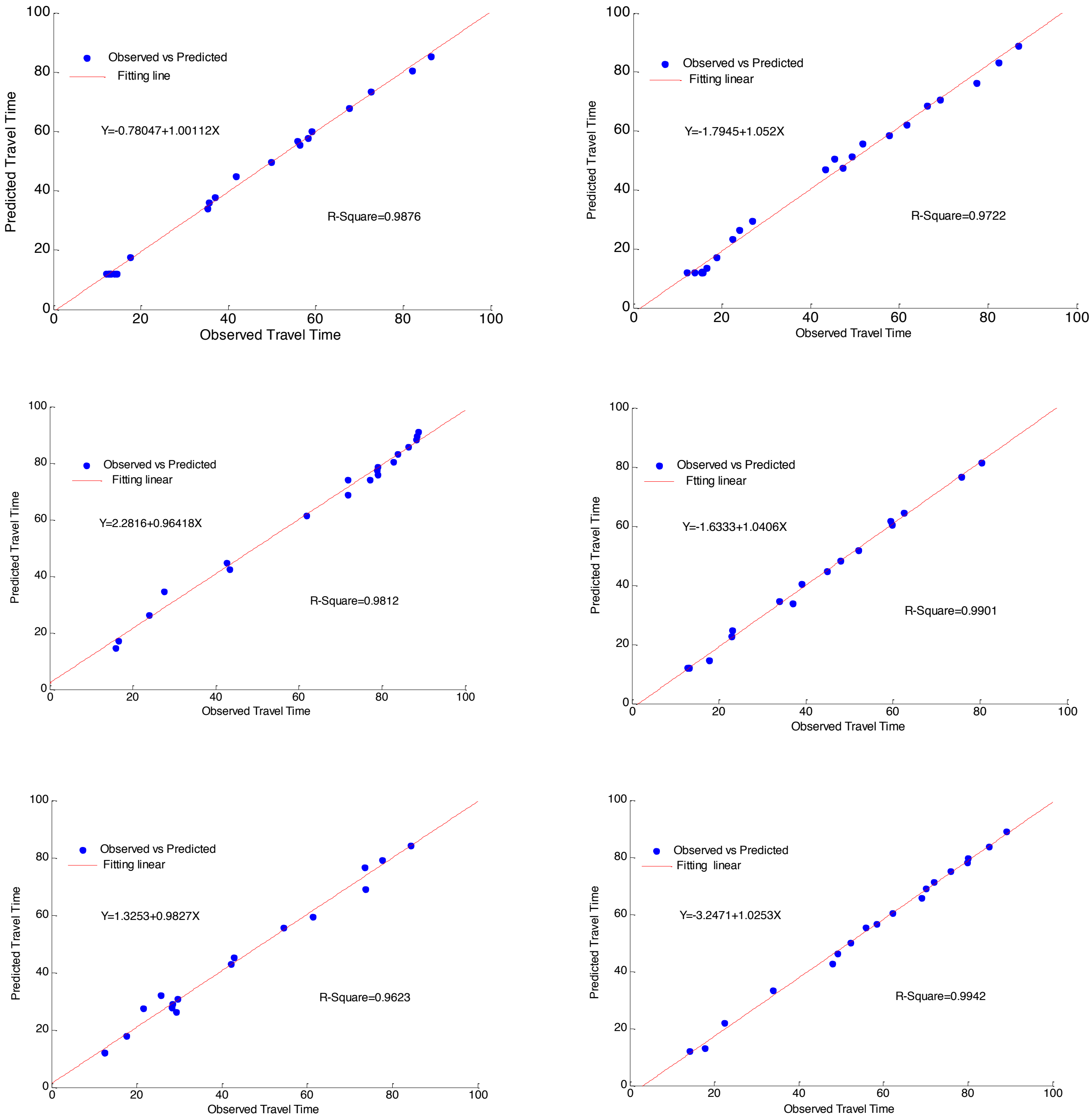


Figure 82. Correlation between Predicted and Observed Travel Time

Figure 82 shows the predicted travel time and the observed travel time (in seconds) are tightly correlated over the investigated periods. A linear regression is applied to compare the predicted with the observed link travel times. It can be seen that the predicted link travel times based on the proposed model exhibit no apparent bias and have very high correlations with the observed values (R^2>0.96).

The difference between the observed and predicted data is further evaluated with two indicators: the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), which are defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(TT_i^p - TT_i^o\right)^2} \quad (31)$$

$$MAPE = 100 * \frac{1}{N}\sum_{i=1}^{N}\left|\frac{TT_i^p - TT_i^o}{TT_i^o}\right| \quad (32)$$

where TT_i^p is the predicted travel time of the i th vehicle and TT_i^o is the observed travel time of the i th vehicle. N is the total number of vehicles compared in the data set. The performance metrics, based on the two indicators, for the investigated periods are presented in Table 34. The RMSE varies from 1.25s to 2.71s and the MAPE is in the range of 3.37% to 8.67%. Both the RMSE and MAPE are satisfactorily low and acceptable for the purpose of this work.

Table 34. Performance Measurement Results

	RMSE(S)	MAPE(%)
Cycle 1	1.25	4.10
Cycle 2	2.57	8.67
Cycle 3	2.42	3.37
Cycle 4	1.54	4.51
Cycle 5	2.71	6.10
Cycle 6	2.30	4.85

The main differences between the predicted and observed travel times can be reasoned to be due to the unpredictable mid-link delay that is caused primarily by several factors, including movements of buses at the bus stop, random driving behaviors of vehicle following, lane change by vehicles, etc. Moreover, it should be pointed out a large error may occur at the boundary conditions when the entry time of the vehicles is near two critical time points t_0, t_3. For instance, the travel time of a vehicle that enters the link with 8 seconds remaining in the green phase is estimated to be 89.31s based on the prediction model. However, in the simulation result, a vehicle entering at the same time is shown to pass the intersection within the current green phase and its travel time is 11.02s. These special cases may happen in real-world situations as vehicles may or may not pass through the intersection when the remaining time is close to the required minimum travel time. Although only one single case happens during the investigated periods in the simulation, this type of special cases should be taken into account in future evaluation to improve the reliability and robustness of the proposed model.

7 Open Source Application Development Portal (OSADP) Submission (AZ)

The Arizona MMITSS software has been submitted to the USDOT Open Source Application Development Portal at http://www.itsforge.net/ Figure 83 is a screen capture of the MMITSS GitHub repository when submitted to OSADP in December 2015. The repository was updated in August 2016.

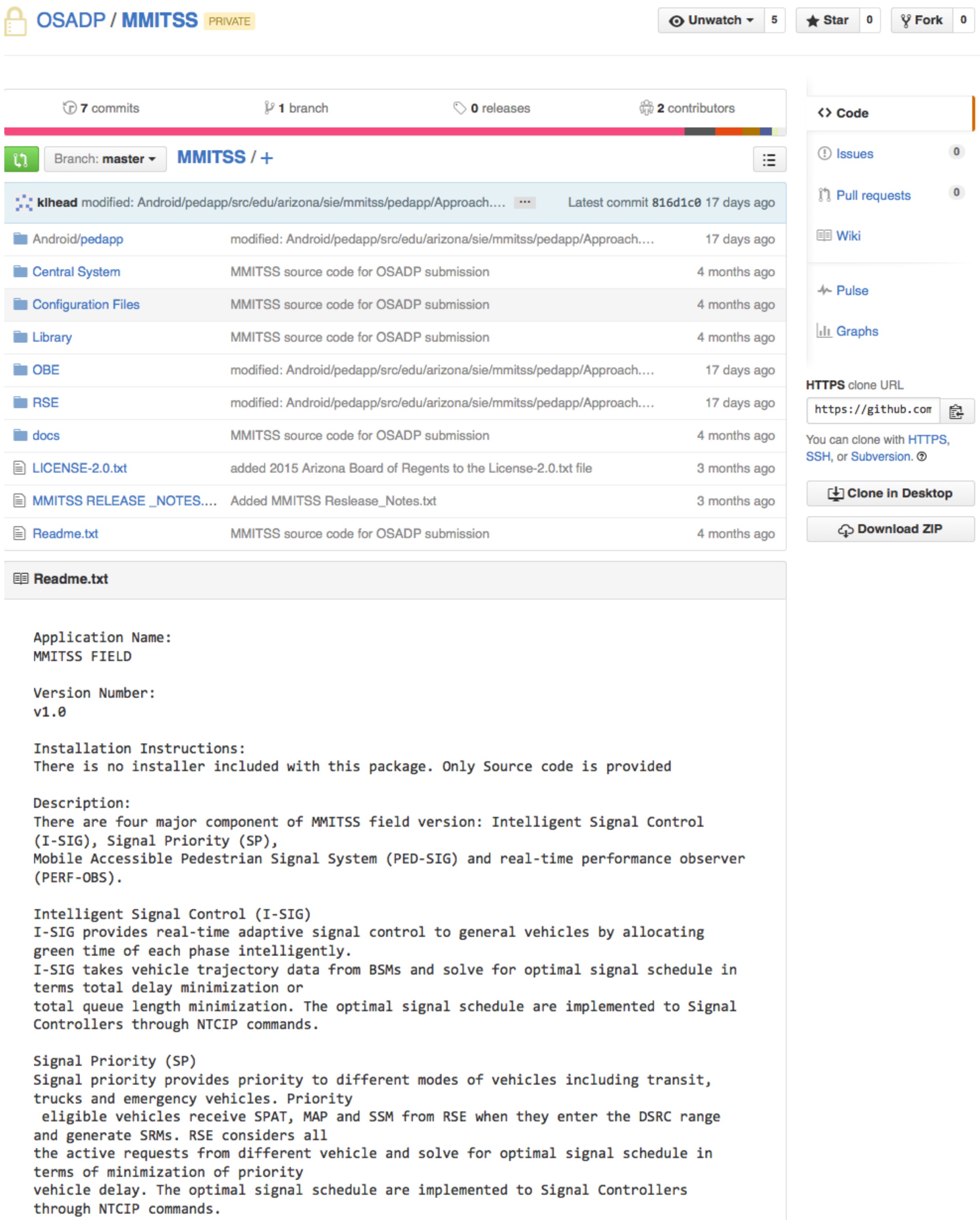


Figure 83. MMITSS submission under OSADP

The California MMITSS software has also been submitted to OSADP. Figure 84 is a screen capture of the MMITSS_CA GitHub repository when submitted to OSADP in February 2016. The California MMITSS submission will be available for public access once the public software license is included in the submission.

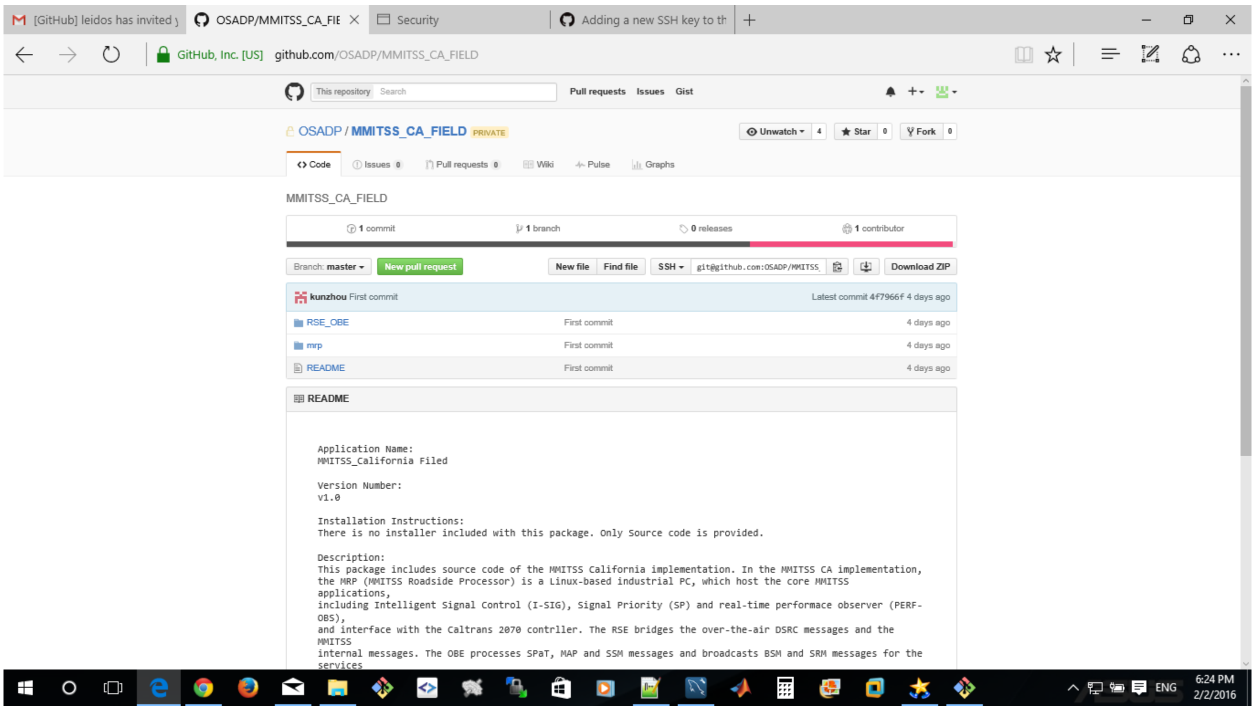


Figure 84 California MMITSS submission under OSADP

8 Recommendations for Future MMITSS Development and Deployment Opportunities

The Phase II MMITSS development, field testing, and demonstrations resulted in two (2) MMITSS prototypes (AZ and CA) that realized advanced traffic control in a connected vehicle environment. Both prototypes implemented innovative traffic control algorithms that utilized the rich data available from connected vehicles. The AZ MMITSS utilized concepts from adaptive and priority based traffic signal control together with the NTCIP standard for communications to the traffic signal controller. The CA MMITSS utilized enhancements to more traditional traffic signal control where the control logic resided within the Caltrans 2070 controller firmware, but adjustments were made based on connected vehicle data. These accomplishments, including the implementations in the two test beds, have created a valuable platform for additional development and field evaluation, that can provide understanding and knowledge that is necessary for improvement of traffic signal operations in a connected vehicle environment, and that will support future development to address more complex traffic management needs and opportunities.

The project team, and Pooled Fund Members, have identified several future research and development activities that will continue the development of MMITSS. This section provides some initial discussion of these activities.

8.1 Field Testing with Higher Market Penetration

One of the greatest needs is to increase the market penetration of connected vehicles in the test beds to allow real performance observation under more realistic conditions that will exist in the future when new vehicles are equipped with connected vehicle technologies. The current field testing didn't support this level of testing and evaluation. Note that when the penetration rate is low, movement of individual equipped vehicles cannot adequately represent the traffic conditions. Traditional vehicle detection systems (using point detectors) provide less data with significantly less information for supporting the Connected Vehicle data needs. These detector systems cannot meet the full requirements for MMITSS testing. Therefore, MMITSS applications involving the interaction of a population of vehicles, such as queue estimation, priority with full consideration of the traffic conditions, and the implementation of the I-SIG, cannot be adequately tested or validated.

The research team has identified two approaches for increasing the market penetration in the test beds. The first approach is to equip a significant share of the vehicles that are served by the traffic signals in the test beds. This option is more realistic in the Arizona Connected Vehicle Test Bed since the population of Anthem, AZ (Maricopa County) is in a closed network where all resident vehicles are likely to travel the test bed intersections. It is estimated that there are approximately 10,000 vehicles registered to the residents in Anthem, so equipping 2000 vehicles, or 20%, would create an interesting level of market penetration. At times, the actual number of equipped vehicles at the intersections would be greater than 20%. At other times it would be lower, but it would allow performance observation and realistic market conditions. Maricopa County has presented the MMITSS concept to the Anthem community board, in a public meeting, and received support for recruiting residents to participate in the study. In addition, Maricopa County has an interagency agreement with the Daisy Mountain Fire Department, so that all emergency vehicles that respond in the test bed could be equipped and provided emergency vehicle priority. Maricopa County is working with the Deer Valley Unified School District to equip their fleet of school buses that serve the schools in Anthem. These partnerships and this approach provide an excellent approach to increasing market penetration.

The second approach is based on using infrastructure based sensors to detect vehicles (both equipped and unequipped) and to create surrogate BSMs that can be used by MMITSS. This approach is more appropriate for the California test bed where the traffic volume is substantial higher than Arizona testbed and it would be difficult to equip enough of the vehicles that traverse the heavily utilized transportation corridor. Researchers at PATH have explored the use of alternative sensing technologies including binocular video processors that can track vehicles in real time and could be used to generate surrogate BSMs. This opportunity would be very useful in other markets across the country where adoption of connected vehicle technology, or turnover of the vehicle fleet, is slow. Transit vehicles, and some private (passenger) vehicles could still be equipped with connected vehicle technologies so that both could be used simultaneously.

These two approaches will enable the project team to evaluate all aspects of MMITSS functions in a real-world condition, for a wide range of penetrations, from 20% in Arizona and up to 100% in California.

8.2 Additional Performance Measures/Observations

The MMITSS Performance Observation capabilities has demonstrated the ability to monitor and assess performance of individual vehicles in real time using connected vehicle data. Connected Vehicle data has been demonstrated very effective in simulation to provide ways that haven't been possible for measuring traffic performance and system performance. To test full set of MMITSS performance observation capabilities developed under this project will provide critical contribution to MMITSS as well as other connected vehicle applications.

Queue length is one of the most important performance measures of an intersection as well as a critical input for signal coordination and coordinated signal priority. In MMITSS phase 2, an algorithm that uses vehicle trajectory data to determine the critical points of shockwave generation and merging, and estimate the shockwave speed and queue length was implemented in VISSIM simulation. However, this queue length estimation algorithm has not being tested with the field data due to the low penetration rate of the connected vehicles. The comprehensive detection system planned for California testbed makes it easy to implement the queue length estimation using real-world data, and to provide input to the signal coordination and section level coordinated signal priority control. By randomly assign detected vehicles as connected vehicles, the impact of penetration rate of connected vehicles on queue length estimation can be also quantified. The results of queue estimation at around 20% penetration rate can then be tested in Arizona testbed.

8.3 Additional Use Cases

Several of the use cases identified in Phase I were not implemented in this Phase II effort (see Table 1 and Section 4.8). Several of these use cases should be included in the next version of MMITSS including:

- Congestion Control
- Nearside Bus Stops
- Transit Priority for Left Turn Signals (note that the AZ MMITSS can provide priority for any phase at an intersection after the vehicle enters the turn lane. Caltrans is pursuing further changes in the control software to allow priority for non-coordinated phases).
- Operational Scenarios for Rail Crossings in Urban Areas
- Equipped Bicyclist(s)
- Inclement Weather Accommodations for Non-Motorized Travelers
- Coordinated Freight Signal Priority along a Truck Arterial
- Route Based Intersection Priority/Preemption

These use cases capture advanced Stakeholder Needs that were identified in the Phase I Stakeholder Needs Assessment activities and should be included in the enhancements of MMITSS.

8.4 Additional Research on Intelligent and Priority Control Algorithms

The AZ MMITSS was based on new traffic control algorithms that were specifically designed to utilize connected vehicle data. There are needs to enhance some of the new algorithms and develop other new algorithms to address the use cases (Section 8.3). These enhancements include:

- Section Level Priority Control

- Integration with Coordination
- N-Level Priority Policy: Impacts and Effects
- Vehicle Dynamics in Priority Requests (Freight, Transit, EV)
- Queue Dynamics

Sectional level priority control is necessary for emergency, transit, and freight vehicles but presents challenges due to the need to know vehicle route information and to accommodate the stochastic effects of link travel time and queueing along the route. In the AZ MMITSS, coordination is considered as a special type of priority and the same priority control algorithm is used to achieve coordination. Integration of priority request with priority based coordination, along with intelligent (adaptive) phase control, is important to achieve the cumulative desired benefits. The N-Level Priority Policy mechanism was introduced in the AZ MMITSS and allows an agency or operator to determine the priority hierarchy (e.g. trucks are more important than local transit, but express transit is more important than trucks, etc.) and how the selection of priority weights, or ranking, impacts both intersection and section level performance. Initial simulation studies have shown the ability of the modal weights in the optimization to impact the performance of the different modes.

Signals in the California test network are operated under the actuated-coordinated control mode with multiple time-of-day timing plans. Signal offset is programed based on historical field survey data and/or field observations, and does not reflect the dynamic and changing of traffic flow and speed. During peak hours, queue spillback often occurs at closely spaced intersections. With the enhanced queue length estimation and detection of vehicle trajectories, the coordination timing plan parameters can be dynamically adjusted to adapt to the prevailing traffic conditions. Improved traffic coordination or progression also benefits the priority eligible vehicles when they are traveling with the major platoons. In MMITSS phase 2, signal priority control at CA testbed is managed at the intersection level, without the consideration of coordinating the signal priority control along a section of signals. A vehicle obtained priority at an upstream intersection may have to stop at a downstream intersection, resulting in possible interruption of signal progression and no net travel time saving for the particular vehicle. The section-level priority control coordinates the priority strategies among a section of signals, based on the projected movements of requested vehicles and signal status of each intersection, thereby gaining net travel time savings for priority eligible vehicles and reducing side impact on general traffic.

Vehicle dynamics in priority requests are currently not considered in the generation of priority requests. At both CA and AZ testbeds, each request is based on the vehicles speed and distance to the intersection stop bar. Different types of vehicles may have different dynamics, such as freight trucks, that can impact how they can be most effectively served. As compared with automobiles, heavy duty vehicles have different dynamics and have longer braking and acceleration distances. By incorporating dynamics of heavy vehicles and traffic conditions, priority can be better served, which in turn will reduce the number of unnecessary stops for transit buses and trucks to offer reduced delays, improved safety and reduced energy consumption. The vehicle dynamics are important considerations in providing effective priority – especially in section, or route level, priority.

Vehicle policies, especially transit policies, can impact if and how priority requests are generated. For example, if a transit vehicle is running on schedule, then it may not be necessary to issue a request. Some agencies and operators believe that a full transit vehicle deserves more priority than an empty vehicle. Others believe that a dead heading bus that is late getting to the start of a route is more important than a full, late bus. The ability to establish control policies for different classes of vehicles is an important enhancement. Finally, queues impact the performance of the intelligent signal control and priority algorithms. This impact has been considered in the MMITSS prototypes, but there is a need to investigate better approaches to managing queues (in congestion) and clearing queues for priority vehicles. The current MMITSS prototypes provide an idea platform for research and development to these important factors and the performance can be improved.

8.5 Freeway Interchange/Ramp Meter Control and Priority

Traffic control extends beyond signalized intersections and includes freeway interchanges and ramp meters as well. In 2007, the ADOT and MCDOT sponsored a research project (SPR-653) to explore the use of vehicle-infrastructure integration (VII) for improving the response time and safety of first responders. The E-VII project explored priority at traffic signals as well as priority at ramp meters for emergency vehicles. The project also explored the use of a road side alert (RSA) and integration with the Arizona 511 system. In 2014, ADOT sponsored a small project to implement MMITSS at the interchange of I-17 and Daisy Mountain in the Arizona Connected Vehicle Test Bed. The project explored priority control concepts at diamond interchanges where the logic is more complex due to the use of overlaps and special phasing. These early efforts were precursors to the current AZ MMITSS system and demonstrate the capability to control freeway interchanges and ramp meters.

Coordination of freeway ramp metering and arterial traffic signal coordination is one of the critical strategy for Integrated Corridor Management (ICM). Previous studies have demonstrated that this ICM strategy can be very effective for achieving efficient operation of both systems. The California Connected Vehicle Testbed along El Camino Real in Palo Alto is parallel to Highway 101. Although El Camino Real does not intersect with Highway 101, significant traffic volume, particularly during the peak traffic hours, moves through a number of arterial streets to travel along El Camino.   We plan to instrument DSRC along one of the arterial streets that connects Highway 101 and El Camino Real. We will develop and implement MMITSS arterial and freeway coordination strategies to evaluate the effectiveness of such strategies.

There is a need to consider both ramp metering control and ramp meter priority. Concepts for advanced ramp metering that use data from connected vehicles should be explored through the same systems engineering process as the current MMITSS. Stakeholders should be engaged to determine their needs. A Concept of Operations should be developed that captures the use cases and different views of the system. The impact of market penetration and strategies to use vehicle control capabilities, such as variable speed control, could change how ramp metering works. These, and other concepts, need to be explored and tested using the MMITSS simulation and field testing capabilities.

8.6 Extending the Communication Range beyond DSRC

DSRC was the primary means of communication for the Phase II MMITSS development and testing, but other communications media could provide significant enhancements for many of the MMITSS use cases. For example, DSRC is limited to 300 meters by regulation, but that only provides approximately 20 seconds of headway for information for traffic control and priority algorithms. Additional time would allow the algorithms to plan ahead and better accommodate traffic conditions. DSRC signals can be blocked by obstacles such as trees in the road medians. Other communication media could be used to extend the range. For example, peer-to-peer communications between adjacent intersections (RSE) could be used to relay information such as priority requests. Or, media such as 3G/4G/5G or other wireless technologies could be used to send messages over the backhaul network, including using services in traffic operations centers or in cloud data processing centers. The benefits of using alternative communications media need to be explored for MMITSS applications.

8.7 Integration with other Connected Vehicle Applications

MMITSS is only one of many connected vehicle applications that have been developed and tested. There are potential benefits from integration of some of these other applications with MMITSS. For example, Eco-Approach and Departure integration could create additional benefits for both passenger vehicles, trucks, and transit as well as create interesting opportunities for electric and other alternative fuel vehicles. The pedestrian application could be enhanced to support bicycles and pedestrian functions such as automated calls (instead of the ped having to press a button). The Savari Smartcross application has some of these features, but there is a need to explore the use of smartphones further to provide a more useful application for pedestrians, especially disabled pedestrians. The pedestrian applications need to be able to send the pedestrian safety message (PSM) that allows vehicles to be aware of vulnerable road users to support safety applications. MMITSS could use this information in making traffic control signal indications dynamic in cases where a turning vehicle's path might conflict with a pedestrian in a cross walk. The integration with other connected vehicle applications provides opportunities for additional benefits that could be tested using simulation and the test beds.

9 Summary of Scholarly Publications, Presentations, and Field Demonstrations

Several Scholarly papers have been submitted, published and presented at national, international, and local transportation confernces. The following list includes some of these papers:

1. Khoshmagham, S., Feng, Y., Zamanipour, M., & Head, K. L. (2016). "Travel Time Observation in Privacy Ensured Connected Vehicle Environment Using Partial Vehicle Trajectories and Extended Tardity". Working paper to be submitted to the Journal of Intelligent Transportation System.
2. Zamanipour, M., Feng, Y., Head, K. L. and Khoshmagham, S. (2016). "An Integrated Multi-Modal Priority and Intelligent Adaptive Signal Control in Connected Vehicle Environment". Working Paper.
3. Feng, Y., Head, K. L., Khoshmagham, S., & Zamanipour, M. (2015). "A real-time adaptive signal control in a connected vehicle environment". Transportation Research Part C: Emerging Technologies, 55, 460-473.
4. Feng, Y., Zamanipour, M., Head, K.L., Khoshmagham, S. (2015). "Connected Vehicle Based Adaptive Signal Control and Applications". Accepted and submitted for publication in the Journal of Transportation Research Record.
5. Zamanipour, M., Head, K.L., Feng, Y., Khoshmagham, S. (2015). "An Efficient Priority Control Model for Multi-Modal Traffic Signal". Accepted and submitted for publication in the Journal of Transportation Research Record.
6. Khoshmagham, S., Feng, Y., Zamanipour, M., & Head, K. L. (2015). "Multi-Modal Data Analytics Comparative Visualization Tool: A Case Study of Pedestrian Crossing Design". Accepted and submitted for publication in the Journal of Transportation Research Record.
7. Feng, Y., Khoshmagham, S., Zamanipour, M., & Head, K. L. (2015). "A Real-time Adaptive Signal Phase Allocation Algorithm in a Connected Vehicle Environment". Transportation Research Board 94th Annual Meeting (No. 15-3547).
8. Zamanipour, M., Khoshmagham, S., Feng, Y., and Head, K.L. (2015). "A Simulation Platform for Test and Evaluation of Signal Control Applications in a Connected Vehicle Environment". Submitted for presentation and publication at the 2015 TRB Annual meeting. [The paper was not accepted for presentation or publication, but will be revised and submitted to another journal.]

9. Head, K.L., Feng, Y., Zamanipour, M., Khoshmagham, S., Khosravi, S., and Mucheli, S. (2015). "A Multi-Modal Intelligent Traffic Signal System: Architecture, Components, and Implementations". Working Paper.
10. Khoshmagham, S., Head, L., & Saleem, F. (2014). "Performance Assessment of Multi-Modal Traffic System Using Micro-Simulation Methods". Transportation Research Board 93rd Annual Meeting.
11. Zamanipour, M., Ding, J., and Head, K.L. (2014). "Priority System for Multimodal Traffic Signal Control". In Transportation Research Board 93rd Annual Meeting (No. 14-3579).
12. Head, K. L., Jun Ding, and M. Zamanipour. (2103). "A priority system for multi-modal traffic signal control". Proceedings of mobilTUM Conference. Munich, Germany.
13. Kun Zhou, Huadong Meng, John Spring and Wei-Bin Zhang (2016), Development and Field Testing of MMITSS in California, California PATH Research Report, University of California at Berkeley, UCB-ITS-PRR-2016-2.

10 References

Bai, F., Stancil, D. D., & Krishnan, H. (2010). Toward Understanding Characteristics of Dedicated Short Range Communications (DSRC) from a Perspective of Vehicular Network Engineers. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking* (pp. 329–340). New York, NY, USA: ACM. http://doi.org/10.1145/1859995.1860033

Battelle, " SIGNAL PHASE AND TIMING AND RELATED MESSAGE Binary Format (BLOB) Details", FHWA Office of Operations Research and Development, February 17, 2012, Rev. C.

Bullock, D., Johnson, B., Wells, R. B., Kyte, M., & Li, Z. (2004). Hardware-in-the-loop simulation. *Transportation Research Part C: Emerging Technologies*, *12*(1), 73–89. http://doi.org/10.1016/j.trc.2002.10.002

Byrne, N., Koonce, P., Bertini, R., Pangilinan, C., & Lasky, M. (2005). Using Hardware-in-the-Loop Simulation to Evaluate Signal Control Strategies for Transit Signal Priority. *Transportation Research Record: Journal of the Transportation Research Board*, *1925*, 227–234. http://doi.org/10.3141/1925-23

Ding, J., He, Q., Head, L., Saleem, F., & Wu, W. (2013). Development and Testing of Priority Control System in Connected Vehicle Environment. Presented at the Transportation Research Board 92nd Annual Meeting. Retrieved from https://trid.trb.org/view.aspx?id=1241792

DLR - Institute of Transportation Systems - SUMO – Simulation of Urban MObility. (n.d.). Retrieved May 25, 2016, from http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/

Feng, Y., Head, K.L., and Ma, W., Estimating Vehicles in the Dilemma Zone and Application to Fixed-time Coordinated Signal Optimization. Transportation Research Record, Vol. 2439, pp. 62-70, 2014.

Gartner, N.H., "OPAC: a demand-responsive strategy for traffic signal control", Transportation Research Record: Journal of the Transportation Research Board, Vol. 906, 1983, pp. 75–81

Gettman, D., Shelby, S., Head, L., and Ghaman, R. "Overview of the ACS-Lite Adaptive Control System", 13th World Congress on Intelligent Transport Systems and Services, 2006.

Goodall, N., Smith, B., Park, B., 2013. Traffic Signal Control with Connected Vehicles. Transportation Research Record: Journal of the Transportation Research Board, Vol. 2381, pp. 65–72.

He, Q., K. L. Head, and J. Ding. Heuristic Algorithm for Priority Traffic Signal Control. Transportation Research Record: Journal of the Transportation Research Board, Vol. 2259, No. -1, Dec. 2011, pp. 1–7.

Head, L., D. Gettman, and Z. Wei. Decision Model for Priority Control of Traffic Signals. Transportation Research Record: Journal of the Transportation Research Board, Vol. 1978, Jan. 2006, pp. 169–177.

Mirchandani, P., & Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, *9*(6), 415–432. http://doi.org/10.1016/S0968-090X(00)00047-4

NTCIP 1202, National Transportation Communications for ITS Protocol Object Definitions for Actuated Traffic Signal Controller (ASC) Units, v02.19. November 2005.

Pigne, Y., Danoy, G., & Bouvry, P. (2010). A platform for realistic online vehicular network management (pp. 595–599). IEEE. http://doi.org/10.1109/GLOCOMW.2010.5700389

Piórkowski, M., Raya, M., Lugo, A. L., Papadimitratos, P., Grossglauser, M., & Hubaux, J.-P. (2008). TraNS: realistic joint traffic and network simulator for VANETs. *ACM SIGMOBILE Mobile Computing and Communications Review*, *12*(1), 31. http://doi.org/10.1145/1374512.1374522

Rondinone, M., Maneros, J., Krajzewicz, D., Bauza, R., Cataldi, P., Hrizi, F., … Cartolano, F. (2013). iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications. *Simulation Modelling Practice and Theory*, *34*, 99–125. http://doi.org/10.1016/j.simpat.2013.01.007

Sen S. and Head K.L., Controlled Optimization of Phase at an Intersection, Transportation Science. Vol. 31, No. 1, pp. 5 – 17, 1997.

Urbanik, T., Beard, S., Gettman, D., Head, L., Bullock, D., Smaglik, E., Campbell, R., and Ablett, M., "Traffic signal state transition logic using enhanced sensor information", NCHRP Project 3-66, Transportation Research Board, 2003.

Wu, H., Lee, J., Hunter, M., Fujimoto, R., Guensler, R., & Ko, J. (2005). Efficiency of Simulated Vehicle-to-Vehicle Message Propagation in Atlanta, Georgia, I-75 Corridor. *Transportation Research Record: Journal of the Transportation Research Board*, *1910*, 82–89. http://doi.org/10.3141/1910-10

11 Appendices

11.1 Acronyms

ABSM	Alternate Basic Safety Message
AC	Alternating Current
ADA	Americans with Disabilities Act (1990)
AQ	Air Quality
API	Application Programming Interface
APS	Accessible Pedestrian Signals
ASC	Actuated Signal Controller
ATIS	Advanced Traveler Information Systems
ATDM	Active Traffic and Demand Management
ATV	All-Terrain Vehicle
BRT	Bus Rapid Transit
BSM	Basic Safety Messages
CA	California
CDRL	Contract Deliverables Requirements List
CMMI	Capability Maturity Model Integration
CONOPS	Concept of Operations
CTS	Cooperative Transportation System
CV	Connected Vehicle
DC	Direct Current
DMA	Dynamic Mobility Applications
DOORS	Dynamic Object Oriented Requirement System
DOT	Department of Transportation
DSRC	Dedicated Short Range Communication
EMS	Emergency Medical/Management Services
ESD	Electro-static Discharge
ETA	Estimated Time of Arrival
EV	Emergency Vehicle
EVP	Emergency Vehicle Preemption
FHWA	Federal Highway Administration
FOM	Figure of Merit
FPS	Feet Per Second
FTA	Federal Transit Administration
FYA	Flashing Yellow Arrow
GID	Geometric Intersection Description
GPS	Global Positioning Systems
IC	Information Center
ID	Identification
IM	Incident Management
INCOSE	International Council on Systems Engineering
ISIG	Intelligent Traffic Signal System
ITS	Intelligent Transportation System
LOS	Level of Service
MD	Maryland
MHz	Megahertz (10^6 Hertz)
MMITSS	Multi-Modal Intelligent Traffic Signal System
MOE	Measures of Effectiveness
MPH	Miles Per Hour
MRP	MMITSS Roadside Processor
MTBF	Mean Time Between Failure

MTTF	Mean Time to Failure
NHTSA	National Highway Traffic Safety Administration
NTCIP	National Transportation Communications for ITS Protocol
OBE	On-Board Equipment
OD	Origin-Destination
OEM	Original Equipment Manufacturer
PATH	Partners for Advanced Transportation Technology
PFP	Pooled Fund Project
PFS	Pooled Fund Study
PI	Principal Investigator
PII	Personally Identifiable Information
PMPP	Point to Multi-Point Protocol
POV	Privately Owned Vehicle
R&D	Research and Development
RSE	Roadside Equipment
RV	Recreational Vehicle
SE	Systems Engineering
SPaT	Signal Phase and Timing
SRM	Signal Request Message
SSM	Signal Status Message
STMP	Simple Transportation Management Protocol
SVN	Subversion (PFP Repository with Version Control)
SyRS	System Requirements
TMDD	Traffic Management Data Dictionary
TSC	Traffic Signal Controller
TSP	Transit Signal Priority
UA	University of Arizona
UC	University of California
UML	Unified Modeling Language
USDOT	United States Department of Transportation
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VDOT	Virginia Department of Transportation
VMT	Vehicle Miles Traveled

11.2 Savari Roadside Equipment and On-Board Vehicle Equipment

StreetWAVE Road-Side-Unit (RSU)

Savari is the leading vendor of DSRC-based RSEs in ITS test beds and commercial applications. Savari StreetWAVE™ has proven its reliability from the freezing cold of a Michigan winter to the extreme heat of an Arizona summer. It offers multiple configuration options for DSRC radios, Wi-Fi and GPS, and is compliant with the United States Department of Transprotation (USDOT) RSE 4.0 specification. StreetWAVE comes with the most thoroughly tested and highly regarded RSE software. In fact, the Ann-Arbor Safety Pilot Model Deployment uses Savari RSUs exclusively.

Key Features

Mobility
- Ease of implementation and deployment
- USDOT specification conformance
- Ease of configuration
- Sturdy mechanical design

Features
- On-board GPS for location services and synchronization
- PoE with full surge protection
- FCC certified
- USDOT rQPL vendor
- Available SDK for application development
- Available remote management support
- PCAP logging and error reporting and support for message

DSRC Message support
- WSA Broadcasts
- Traveler Information Messages
- Roadside Alert
- Signal Phase and Timing (SPaT) Messages
- GID/MAP Messages
- RTCM Messages

Standard support
- IEEE 802.11p
- IEEE 1609.2
- IEEE 1609.3
- IEEE 1609.4
- SAE J2735
- NTCIP

Networking
- IPv6 and IPv4 support
- SIT Tunnel support
- SSL
- SSH
- TLS

Applications

Selected V2I Safety (sample)
- I2V - Traffic signal violation warning
- I2V - Curve speed warning
- I2V - Left turn assistant
- I2V - Stop sign movement assistance
- I2V - Red light violation warning
- I2V - Reduced speed zone warning

Components

Processor	533 Mhz
Memory	256 MB DDR DRAM
Storage	8 GB Flash
DSRC Interfaces	Dual Radio Support
GPS	U-Blox Traking Sensitivity -160dBM
Power Rating	48 VDC
Temperature	-40C to +85C
Dimensions	8" (L) x 8 ½" (H) x 2 ¾" (D)
Antenna Connectors	N-Type Male
LED's	Power and Status
Standards Compliance	SAE J1211 sections 3.2.7 and 3.2.8
FCC Compliance	FCC Part 15B/IC ICES-003 Class A, FCC Rule Part 90, 95L
Traffic Controller Compatibility	Compatible with NTCIP compliant traffic controllers
Accessory Kit (Optional)	Pole mounting kit PoE (Power over Ethernet) Controller Kit
SDK (sold separately)	Available as VM image for rapid deployment and testing of connected vehicle applications. Sample applications are available through Comprehensive Programmers Guide

SAVARI™

www.savari.net // +1 408 833 6369

www.savari.net

MobiWAVE On Board Equipment

Overview

The MobiWAVE™ On Board Equipment (OBE) family of products includes the Vehicle Awareness Device (VAD), the Automotive Safety Device (ASD), the Modular Communications Platform (MCP), and the Software Development Kit (SDK). The MobiWAVE™ OBE is designed to address the needs of the connected vehicle market. The MobiWAVE™ OBE supports a variety of automotive safety and commercial applications.

MobiWAVE™ VAD and ASD are compact, ruggedized safety devices. They are capable of transmitting signed "Here I Am" basic safety messages (BSM) to other vehicles and devices over a dedicated short range communications (DSRC) 5.9 Gigahertz (GHz) wireless network using the protocol stack and other standards associated with DSRC for vehicular communications. These include: IEEE 802.11p, IEEE 1609.1 through 1609.4, and J2735 and a performance standard under development by the automobile industry. Both feature a highly accurate GPS receiver and a 5.9GHz DSRC radio. The DSRC radio is used to communicate to other vehicles. The GPS receiver is used to determine accurate location of the vehicle. These devices have a provisioning/test interface that can receive and load new versions of software, new configurations and credentials, and instructions to perform logging functions and download log messages to external storage. The VAD and ASD can be mounted on different classes of vehicles like light passenger cars, trucks, and public transit buses.

MobiWAVE™ MCP is a module, which is easily integrated into a variety of embedded vehicle devices and platforms. The MCP features a high power DSRC/Wi-Fi radio transceiver, a highly accurate GPS receiver, and a powerful application processor. It comes pre-loaded with Savari firmware. It is ready to be deployed to support a variety of connected vehicle applications.

MobiWAVE™ SDK is integrated with VAD, ASD, MCP and can be used to develop new features and applications.

Technical Specifications

Device	Power	Wireless	GPS	Port	Antenna	Storage
VAD	12VDC USCAR connector	1 25dbmDSRC/ Wi-Fi 5.15-5.9GHz, 10, 20 MHz channels, 802.11a	+/- 2M Position Accuracy, 50% CEP	1 Ethernet 1 RS-232 2 USB 2 FAKRA	Multiband Wi-Fi/DSRC/GPS	Up to 512MB internal, USB external
ASD	12VDC USCAR connector	2 concurrent 25dbmDSRC/ Wi-Fi 5.15-5.9GHz, 10, 20 MHz channels, 802.11a	+/- 2M Position Accuracy, 50% CEP	1 Ethernet 1 RS-323 2 USB 3 FAKRA	Multiband Wi-Fi/DSRC/GPS	Up to 4GB internal, USB external
MCP	3.3VDC	1 25dbmDSRC/ Wi-Fi 2.4-2.4835GHz 5.15-5.9GHz, 10, 20 MHz channels, 802.11a/b/g/n	+/- 2M Position Accuracy, 50% CEP	1 SDIO 3 MMCX	N/A	N/A

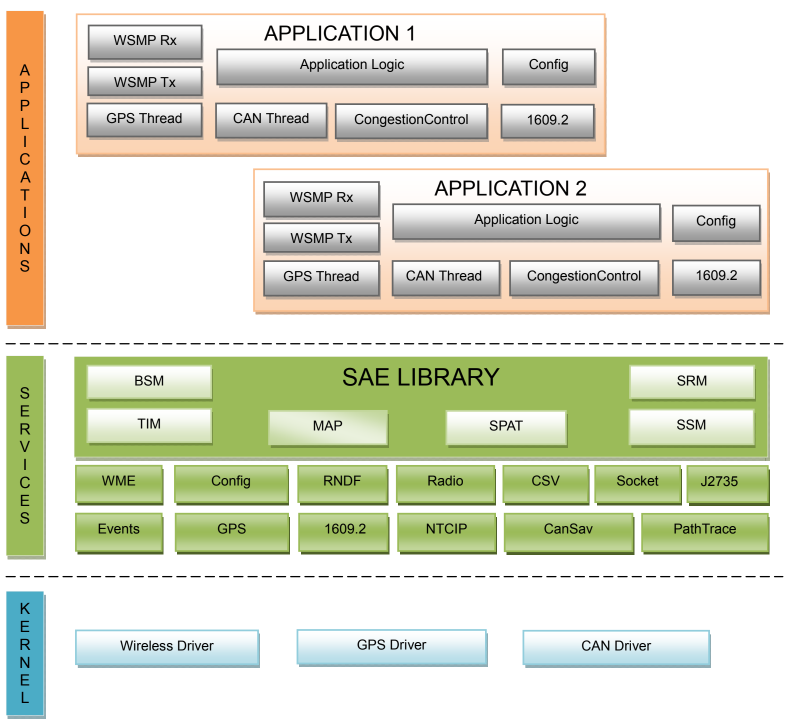
SDK

MobiWAVE™ Software Development Kit (SDK) provides the tools for implementing a variety of safety applications for the ASD and MCP platforms. The SDK includes the Savari On-Board Operating System (SOBOS), which is based on Linux, the APIs, the example safety applications as well as testing and integration tools. All of the MobiWAVE™ devices use the SOBOS as the software framework.

The APIs support GPS, CAN, SAE 2735, and WME. Additional APIs can be used to configure radio interfaces for channel specific parameters along with retrieving debug information. All application messages are signed using the IEEE 1609.2 standard.

All program development is done using the Linux environment. SDK can support the following applications:

- Emergency Brake Light Warning
- Forward Collision Warning
- Intersection Movement Assist
- Blind Spot and Lane Change Warning
- Do not pass Warning
- Control Loss Warning



Savari On Board Operating System

1